ICISNA'25

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

PROCEEDINGS BOOK

Antalya, TURKIYE

December 12-14, 2025

International Conference on Intelligent Systems and New Applications

3rd International Conference, ICISNA'25
Antalya, TÜRKİYE, December 12-14, 2025

Proceedings Book

Editor
Karl JONES

International Conference on Intelligent Systems and New Applications, **ICISNA'25**
Antalya, Türkiye, December 12-14, 2025

3rd International Conference on Intelligent Systems and New Applications

12-14 DECEMBER 2025

Editor
Karl JONES

International Conference on Intelligent Systems and New Applications

Antalya, TÜRKİYE, December 12-14, 2025

TOTAL NUMBER OF PAPERS
27

EVALUATION PROCESS
All Submissions Have Passed a Double-Blind Review Evaluation Process

CONFERENCE LANGUAGE
English

PRESENTATION
Oral Presentation

PREFACE

It is with great pleasure to present the proceedings of the International Conference on Intelligent Systems and New Applications (ICISNA'25). The conference was held virtually from December 12-14, 2025.

ICISNA'25 aims to bring researchers, practitioners, and industry experts from around the world to exchange ideas and share their latest findings in the field of intelligent systems and their new applications. This conference provided an opportunity for attendees to discuss recent advancements, challenges, and future directions in intelligent systems research.

All paper submissions were double blind and peer reviewed and evaluated based on originality, technical and/or research content/depth, correctness, relevance to conferences, contributions, and readability. Selected papers presented at the conference that match the topics of the journals will be published in the following journals:

• International Journal of Applied Mathematics, Electronics and Computers (IJAMEC)

• International Journal of Automotive Engineering and Technologies (IJAET)

• International Journal of Energy Applications and Technology (IJEAT)

• Intelligent Methods in Engineering Sciences (IMIENS)

A total of 42 papers were submitted to the International Conference on Intelligent Systems and New Applications (ICISNA'25), and each paper was evaluated by two independent reviewers. Following the review process, 27 papers from 14 different countries (Albania, Algeria, Ethiopia, Hungary, India, Iran, Jordan, Lithuania, Madagascar, Tunisia, Türkiye, Ukraine, Uruguay, Vietnam) were accepted and presented at the conference. Among the presented papers, 10 were authored by researchers from Türkiye. We would like to express our gratitude to all the authors who submitted their papers, as well as the members of the organizing committee, the technical program committee, and the reviewers for their hard work and dedication in making ICISNA'25 a success.

We hope that you find the proceedings book informative and inspiring, and we look forward to seeing you at the next ICISNA.

Karl JONES
Editor

PROGRAMME COMMITTEES

GENERAL CHAIR :

Karl Jones, Liverpool John Moores University, UNITED KINGDOM

CO-CHAIRS :

Ismail Saritas, Selcuk University, TÜRKİYE

ORGANIZING COMMITTEE :

Amar Ramdane Cherif, University of Versailles, FRANCE
Angel Smrikarov, Rousse University, BULGARIA
Helen Burrell, Liverpool John Moores University, UNITED KINGDOM
Ilker Ali Ozkan, Selcuk University, TÜRKİYE
Ismail Saritas, Selcuk University, TÜRKİYE
Jurgis Porins, Riga Technical University, LATVIA
Karl Jones, Liverpool John Moores University, UNITED KINGDOM
Lilia Georgieva, Heriot Watt University, UNITED KINGDOM
Marco Porta, University of Pavia, ITALY
Mohamed Alobeady, University, IRAQ
Mohamed Hanifa Mohamed Sirajudeen, University, INDIA
Murat Koklu, Selcuk University, TÜRKİYE
Sebastian Chandler Crnigoj, Liverpool John Moores University, UNITED KINGDOM
Silyan Sibinov Arsov, Rousse University, BULGARIA
Stavros Nikolopoulos, University of Ioannina, GREECE
Suzzanne Mccoll, Liverpool John Moores University, UNITED KINGDOM
Vikram Yadav, Bundelkhand Institute of Engineering and Technology Jhansi India, INDIA
Zarifa Jabrayilova, Institute of Information Technology Anas, AZERBAIJAN

INTERNATIONAL ADVISORY BOARD :

Alexander Sudnitson, Tallinn University of Technology, ESTONIA
Amar Ramdane Cherif, University of Versailles, FRANCE
Amir A Ghavifekr, University of Tabriz, IRAN ISLAMIC REPUBLIC OF
Anca Loana Andreescu, Academy of Economic Studies, BULGARIA
Anne Villems, University of Tartu, ESTONIA
Antonella Reitano, University of Calabria, ITALY
Antonio Mendes, Universidade De Coimbra, PORTUGAL
Artan Luma, Southeast European University, MACEDONIA

CONFERENCE PROGRAMME

SESSION I : 12.12.2025 – FRIDAY (10:30 – 12:00) (GMT +3)
SESSION CHAIR : BLERIM ZYLFIU
VIRTUAL HALL-1

0125 : COMPARATIVE EVALUATION OF MULTI CRITERIA DECISION MAKING METHODS IN THE ONLINE CADCOM PLATFORM
LAVDIM MENXHIQI
Presenter: LAVDIM MENXHIQI

0114 : DIGITAL TWIN SYNCHRONIZATION WITH REAL TIME DATA COLLECTION
ABDULKADIR SADAY
Presenter: ABDULKADIR SADAY

0118 : CONVOLUTIONAL NEURAL NETWORK BASED FRAMEWORK FOR THE DETECTION OF TUBERCULOSIS
AONDOWASE JAMES ORBAN
Presenter: AONDOWASE JAMES ORBAN

0117 : SMARTCODEHUB LLM BASED FRAMEWORK FOR SEMANTIC CODE REUSE IN REACTIVE PROGRAMMING
AONDOWASE JAMES ORBAN
Presenter: AONDOWASE JAMES ORBAN

0124 : AN INTELLIGENT MULTI ALGORITHM INTEGRATION FRAMEWORK FOR AUTOMATED M A DECISION SUPPORT IN TECHNOLOGY INTENSIVE INDUSTRIES
BLERIM ZYLFIU
Presenter: BLERIM ZYLFIU

0112 : DIGITAL TWIN FOR CRYOGENIC EJECTOR SYSTEMS INTEGRATING ADVANCED MACHINE LEARNING AND DYNAMIC MODELING
LOTFI SNOUSSI, OLFA FAKHFAKH, EZZEDINE NEHDI
Presenter: LOTFI SNOUSSI

SESSION II : 12.12.2025 – FRIDAY (14:00 – 15:30) (GMT +3)
SESSION CHAIR : YAVUZ SELIM TASPINAR
VIRTUAL HALL-1

0105 : FREQUENCY DOMAIN CHARACTERIZATION OF HIGH ORDER MODEL SYNCHRONOUS MACHINE PARAMETERS UNDER STANDSTILL CONDITIONS
LEGUEBEDJ FARID
Presenter: LEGUEBEDJ FARID

0106 : SIMULATION BASED EVALUATION OF LIDAR PHOTOGRAMMETRY FUSION VIA NERF RECONSTRUCTION AND ICP REGISTRATION IN URBAN SCENES
RYTIS MASKELIUNAS, SARMAD MAQSOOD, AHMAD QURTHOBI, IRFAN ABBAS, MANTAS VASKEVICIUS, JULIUS GELSVARTAS
Presenter: IRFAN ABBAS

0110 : REACT MODULAR AGENT ORCHESTRATING TOOL USE AND RETRIEVAL FOR FINANCIAL WORKFLOWS
ARMANDO HERNANDEZ DE LA VEGA, SANTIAGO PEREZ, VICTOR SABBIA
Presenter: ARMANDO HERNANDEZ DE LA VEGA

0128 : EMBEDDING BASED MACHINE LEARNING APPROACH FOR AUTOMATIC CLASSIFICATION OF TURKISH NEWS ARTICLES
AHMET ATASOGLU, YAVUZ SELIM TASPINAR
Presenter: YAVUZ SELIM TASPINAR

0132 : REAL TIME FAULT DETECTION IN 3D PRINTERS WITH A HYBRID DEEP LEARNING MODEL
COSKUCAN BUYUKYILDIZ, ISMAIL SARITAS
Presenter: COSKUCAN BUYUKYILDIZ

0127 : COMPUTER VISION BASED BEHAVIOR ANALYSIS FOR WORKPLACE EFFICIENCY BAKERY ENVIRONMENT APPLICATION
CENGIZ SAMET TEPE, ILKAY CINAR
Presenter: CENGIZ SAMET TEPE

0129 : A DEEP NEURAL NETWORK BASED MULTI AGENT MIXTURE OF EXPERTS FRAMEWORK FOR GENERATING AI EDUCATIONAL CONTENT IN MALAGASY LANGUAGE FOR CHILDREN 5–10
MAHEFA ABEL RAZAFINIRINA MAHEFA, RINDRA NADIA RAZAFIARINIRINA NADIA, ELYSA RAZAFINDRAFARA ELYSA, WILLIAM GERMAIN DIMBISOA WILLIAM, THOMAS MAHATODY MAHATODY
Presenter: MAHEFA ABEL RAZAFINIRINA MAHEFA

SESSION III : 13.12.2025 – SATURDAY (09:00 – 10:30) (GMT +3)
SESSION CHAIR : P MANIKANTA SIMGAMSETTI
VIRTUAL HALL-1

0107 : TRANSFORMING HEALTHCARE: THE ROLE OF ARTIFICIAL INTELLIGENCE TODAY AND BEYOND
HA VAN SY HA, PHAM THI MAI LIEN PHAM
Presenter: HA VAN SY HA

0111 : EFFICACIOUS LUNG CANCER DETECTION UTILISING HYBRID DEEP LEARNING AND SOPHISTICATED IMAGE PROCESSING
P MANIKANTA SIMGAMSETTI
Presenter: P MANIKANTA SIMGAMSETTI

0121 : DESIGN AND SIMULATION OF A SMART MULTI FLOOR ELEVATOR CONTROLLER USING VERILOG HDL
SASHANK ABBURU, SRIKANTH DINESH, T PUSHKAR REDDY, DHEEPAK K, KAVERI HATTI, PARAMASIVAM C
Presenter: DHEEPAK K

0126 : INSPECTION OF STORAGE TANK BOTTOMS AND CORROSION MAPPING VIA ULTRASONIC TESTING AND SIGNAL PROCESSING METHODS
KEMAL OZGUVEN, ISMAIL SARITAS
Presenter: KEMAL OZGUVEN

0130 : TOMATO SEED CLASSIFICATION WITH ARTIFICIAL INTELLIGENCE: A SQUEEZENET-BASED APPROACH
ABDULKADIR SADAY, ILKER ALI OZKAN
Presenter: ABDULKADIR SADAY

0131 : AUTOMATED QUALITY CONTROL IN WELDING PROCESSES WITH YOLOV8
ADEM DILBAZ, ILKER ALI OZKAN
Presenter: ADEM DILBAZ

0134 : DEEP LEARNING METHODS FOR THE CLASSIFICATION OF TURKISH MUSIC GENRES
MUHAMMED EMINOGLU, MURAT KOKLU
Presenter: MUHAMMED EMINOGLU

SESSION IV : 13.12.2025 – SATURDAY (11:30 – 13:00) (GMT +3)
SESSION CHAIR : KIRUTHICK K M
VIRTUAL HALL-1

0104 : INTELLIGENT DISEASE DETECTION USING A HYBRID DEEP LEARNING SVM FRAMEWORK
NEGIN AMIRZADEH
Presenter: NEGIN AMIRZADEH

0115 : AN INTELLIGENT ANN-BASED FRAMEWORK FOR PREDICTING EMPLOYEE ATTRITION IN IMBALANCED DATA SCENARIOS
ESMAEL AHMED
Presenter: ESMAEL AHMED

0116 : EDGE INTELLIGENT BIOSENSING SYSTEMS WITH DUAL OPTIMIZATION OF SIGNAL PROCESSING AND ENERGY MANAGEMENT
YEVHENIIA BABENKO
Presenter: YEVHENIIA BABENKO

0120 : FIELD OF VIEW-BASED SPHERICAL VIDEO STITCHING AND OBJECT DETECTION FOR MULTI-CAMERA DRONES
GHASSAN AL JAYYOUSI, GHAITH AL REFAI, MUTAZ RYALAT, HISHAM ELMOAQET
Presenter: GHAITH AL REFAI

0122 : SECURE VOTING SYSTEM USING FPGA
KIRUTHICK K M, RUSHINDRA K R, KARTHIKEYAN T, KAVERI HATTI
Presenter: KIRUTHICK K M

0133 : AN EXPLAINABLE DEEP LEARNING FRAMEWORK FOR AGTRON-BASED COFFEE ROAST CLASSIFICATION USING GRAD-CAM
HAVVA HAZEL ARAS, YUSUF ERYESIL, MURAT KOKLU
Presenter: HAVVA HAZEL ARAS

0135 : EVALUATION OF CNN MODELS FOR MULTI-CLASS GEAR FAULT DETECTION USING WAVEFORM IMAGES
MUCAHID MUSTAFA SARITAS, OYA KILCI, MURAT KOKLU
Presenter: MUCAHID MUSTAFA SARITAS

TABLE OF CONTENTS

Proceedings Paper DOI: 10.58190/icisna.2025.133

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

An Intelligent Multi-Algorithm Integration Framework for Automated M&A Decision Support in Technology-Intensive Industries

Blerim Zylfiu[1]

[1]*Computer Science and Engineering Department, University for Business and Technology, Pristina, Kosovo*

blerim.zylfiu@ubt-uni.net, ORCID: 0000-0002-9727-2621

***Abstract*— The time-based patterns of technology-driven mergers and acquisitions exceed what standard binary classification systems can identify. The research focuses on predicting acquisition timing for companies during times of technological change instead of simply determining acquisition status. The research uses survival analysis through Cox-inspired risk scoring framework to analyze 661 M&A deals in Electronic Design Automation (EDA) from 1975 to 2025. The research combines four analytical approaches which include temporal features and technological strength and network position and strategic archetypes. The research uses historical acquisition data from 2015 to 2020 to validate the model through calculated concordance index and Brier score and time-dependent AUC metrics. The research findings show that companies face their highest acquisition risk during their third to seventh year of operation. The research shows that companies with high network centrality (degree > 0.5) experience a 30% reduction in acquisition risk. The research shows that acquisition rates between companies vary from 45% for established technology leaders to 89% for specialized businesses. The model generates survival probability estimates and explains which factors influence the results to fill a major knowledge gap in corporate finance research.**

***Keywords*— Survival Analysis, Cox Proportional Hazards, Mergers and Acquisitions, Technology Disruption, EDA Industry, Kaplan-Meier Curves, Concordance Index**

I. INTRODUCTION

A. Research Problem and Motivation

The Electronic Design Automation industry has experienced 661 mergers and acquisitions during the last five decades starting from 1975 until 2025 because of technological progress and market consolidation and competitive needs [1] [20]. The industry needs to understand corporate viability because stakeholders face their biggest technological change from CMOS-based semiconductor design to quantum computing and nanotechnology fabrication [1][2].

The current M&A prediction models face three core problems which affect their accuracy:

Binary Classification Bias: The acquisition process in traditional logistic regression and machine learning classifiers uses binary outcomes to predict acquisition status without considering the timing of events [3].

Censoring Mishandling: The training data process for right-censored companies either removes them or uses incorrect coding methods which produce biased results that favor short-lived businesses [4].

Single-Dimension Feature Spaces: Research studies use individual analytical methods to study financial ratios and patent counts and network positions without uniting diverse data points [5][6].

Empirical Gap: The statistical method of survival analysis which predicts time-to-event outcomes in medical prognosis and reliability engineering has not been applied to study corporate M&A patterns during technological paradigm shifts in technology industries [7][8].

B. Research Contributions

The research delivers four essential contributions which unite corporate finance with technology management and applied econometrics.

Methodological Innovation: The research implements survival analysis through Cox Proportional Hazards and Random Survival Forests and Kaplan-Meier curves to predict technology company mergers and acquisitions while addressing right-censored data points and time-dependent risk factors.

Multi-Dimensional Feature Engineering: The research combines four different analytical approaches which include:

- Temporal: The research examines how companies develop over time and how mergers follow specific patterns throughout the year.
- Technological: The TechImpactScore combines patent data with deal value proxies and company age and acquirer activity metrics to create a

composite metric which undergoes Principal Component Analysis validation.

- Network-Structural: The research analyzes M&A transaction graphs through directed networks to calculate degree and betweenness and eigenvector and closeness centrality metrics.
- Strategic-Cluster: The research identifies five acquisition archetypes (C0–C4) through Reverse Hybrid Clustering which starts with DBSCAN followed by K-Means and ends with Noise Reintegration.

Time-Aware Historical Validation: We use the backtesting framework to simulate predictions that are made N years before the acquisitions. The backtesting framework extracts features, from the states. The backtesting framework avoids data leakage. The backtesting framework provides performance estimates.

Empirical Industry Analysis:

- Comprehensive analysis of 661 EDA M&A transactions revealing:
- Non-monotonic age-hazard relationship (peak risk at 3–7 years)
- Network centrality protection effects (30% hazard reduction)
- Cluster-specific acquisition rates (45%–89%)
- Merger wave amplification (25% hazard increase during peaks)

We examined the Electronic Design Automation industry. The Electronic Design Automation industry includes the software and hardware toolchain ecosystem that lets designers create semiconductor chips. The Electronic Design Automation industry is, in a technology change. The industry sees a shift because quantum computing and nanotechnology may make CMOS design tools old. The industry now needs to combine companies buy technology and move its market position [21].

II. LITERATURE REVIEW AND THEORETICAL FOUNDATION

A. M&A Prediction in Technology Industries

The three main traditional M&A prediction research methods contain essential restrictions which affect their accuracy:

Financial Ratio Models: Harford and Uysal [9] employed logistic regression to analyze three financial ratios which included leverage ratios and liquidity metrics and profitability indicators for acquisition likelihood prediction. The models fail to account for time-based changes because they only show acquisition risk but not the timing of acquisitions and they do not use survival probability distributions.

Patent-Based Innovation Models: Ahuja and Katila [10] studied patent citation patterns and technological scope to identify acquisition targets. Grimpe and Hussinger [11] studied pre-emptive acquisition strategies through patent complementarity analysis. The methods fail to detect how companies use their networks to achieve strategic advantages beyond their technological resources.

Network Analysis: Schilling and Phelps [12] investigated how companies use their network connections to make acquisition decisions. Stuart and Podolny [13] studied how local search patterns and technological expertise influence acquisition decisions. The method uses fixed network observations which do not include survival probability calculations or hazard rate modeling.

The current methods fail to estimate acquisition time intervals and correctly handle active companies because they produce biased predictions that favor short-lived businesses and generate incorrect acquisition probability estimates [14].

B. Survival Analysis: Mathematical Foundation

Survival analysis models the time until an event occurs, accounting for censored observations where the event has not yet happened [15].

Survival Function: The survival function S(t|X) represents the probability a company remains independent beyond time t given feature vector X:

$$S(T \mid X) = P(T > T \mid X)$$

where:

- T = time-to-acquisition (random variable, measured in years)
- X = feature vector (company age, TechScore, centrality, cluster, wave intensity)
- S(t|X)∈[0,1] with boundary conditions: S(0|X)=1 and limt→∞S(t|X)=0

Hazard Function (Instantaneous Risk): The hazard function h(t|X) models instantaneous acquisition risk [16]:

$$h(X) = \frac{P(T \geq t, X)}{\Delta t}$$

Interpretation: $h(X)\Delta t$ approximates the probability of acquisition in the interval $[t, t + \Delta t)$ given survival up to time t.

Cox Proportional Hazards Model: The Cox model specifies hazard as a product of baseline hazard and covariate-dependent multiplicative factor [17]:

$$h(t\backslash midX) = h0(t) \times exp(\beta 1X1 + \beta 2X2 + \cdots + \beta pXp)$$

where:

- h0(t)= baseline hazard (unspecified)
- β = regression coefficients
- X = covariate vector

Key Property: The Cox model is semi- parametric. Hazard ratios remain constant over time:

$$\frac{h(X_i)}{h(X_j)} = exp\ \text{exp} \quad ! \left(\beta^T (X_i - X_j)\right)$$

Survival Function from Hazard: The survival function relates to cumulative hazard $\Lambda(X)$

$$S(X) = exp\ \text{EXP} \quad ! \left(-\int_0^t h(X)\, du\right) = exp\ \text{EXP} \quad ! \left(-\Lambda(X)\right)$$

For exponential hazard (constant $h(X) = \lambda$):

$$S(t) = e^{-\lambda t}$$

Median Survival Time:

$$T_{median} = \frac{ln\ ln\ (2)}{\lambda}$$

These metric answers: "When is this company most likely to be acquired"?.

C. Feature Engineering for Corporate Viability

The feature vector X contains four analytical dimensions which we derived from previous EDA industry research [1][18].

Temporal Features: Company Age (Xage): Years since founding. Young startups (<3 years) face high acquisition risk; mature firms (>20 years) tend to remain independent [19].

Years Since Last M&A Wave ($X_{wave_{time}}$): Measures time elapsed since last merger wave peak.

Technology Features: TechImpactScore ($X_{tech} \in [0,10]$):

$$X_{tech} = 0.35\, S_{patent} + 0.30\, S_{deal} + 0.20\, S_{age} + 0.15\, S_{acquirer}$$

Derived using PCA; correlated with observed deal values (r=0.71, p<0.001).

Network Features: From directed M&A graph G=(V, E):

Degree Centrality:

$$C_D(v) = \frac{|N(v)|}{|V| - 1}$$

Betweenness Centrality:

$$C_B(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Eigenvector Centrality:

$$C_E(v) = \frac{1}{\lambda} \sum_{u \in N(v)} C_E(u)$$

Network Isolation:

$$X_{isolation} = 1 - C_D(v)$$

Cluster Features: Reverse Hybrid Clustering identifies 5 archetypes:

- C0 – Niche Specialists: 24.1%, 89% acquisition
- C1 – Early Consolidators: 4.1%, 67%
- C2 – Tech Leaders: 36.6%, 45%
- C3 – Strategic Acquirers: 7.0%, 82%
- C4 – Platform Consolidators: 28.6%, 73%

Cluster Acquisition Rate ($X_{cluster_{rate}}$): Historical acquisition probability per archetype (used as hazard modifier).

III. METHODOLOGY

A. Data and Sample Characteristics

Dataset Composition:

- M&A Transactions: 661 acquisitions spanning 1975–2025 (51 years)
- Unique Companies: 675 (targets and acquirers) Industries: Electronic Design Automation (EDA) toolchains, IP providers, semiconductor CAD
- Geographic Coverage: Global (North America 68%, Europe 22%, Asia 10%)

Data Sources:

- M&A transaction dates and parties: SEC filings, company announcements, Crunchbase
- Patent counts: USPTO database
- Technology scores: Composite TechImpactScore
- Network structure: Directed M&A graph

Strategic Cluster Distribution: Reverse Hybrid Clustering [1] identified five archetypes:

TABLE I – ARCHETYPES

Cluster	%	Avg Age	Avg Patents	Avg TechScore	Rate
C0 – Niche Specialists	24.1%	11.2	18.6	7.8	89%
C1 – Early Consolidators	4.1%	9.8	4.2	5.1	67%
C2 – Tech Leaders	36.6%	18.6	12.3	6.5	45%
C3 – Strategic Acquirers	7.0%	4.2	2.1	4.3	82%
C4 – Platform Consolidators	28.6%	1.8	8.7	6.9	73%

Data Partitioning:

- Training Set: 1975–2014
- Validation Set: 2015–2020
- Prospective Set: 2021–2025

Right-censored companies (not yet acquired) are treated correctly under survival analysis.

B. Feature Extraction Pipeline

For each company i at time t, the feature vector Xi(t) is constructed through four steps.

Temporal Feature Extraction:

age_i(t) = t - t_founding

wave_time_i(t) = t - t_last_wave_peak

Technology Feature Extraction:

patent_score = min(10, patent_count / 2)
age_score = {8.0 (<5 yrs), 6.0 (<10 yrs), 4.0 (<20 yrs), 2.0 otherwise}
acquirer_score = min(10, acquisition_count * 0.5)
tech_score = 0.35*patent + 0.20*age + 0.15*acquirer + 0.30*baseline

Network Feature Extraction:

degree_i = (in_degree + out_degree) / (|V| - 1)
betweenness_i = shortest-path enumeration
eigenvector_i = power iteration
isolation_i = 1 - degree_i

Cluster Assignment:

cluster_i = argmax P(c | age_i, tech_score_i, degree_i)
cluster_rate_i = historical acquisition rate of cluster_i.

C. *Cox-Inspired Risk Scoring Framework*

The Cox-inspired risk scoring framework we use combines rule-based feature engineering with exponential hazard transformation. The method differs from standard Cox regression with maximum partial likelihood estimation because it applies domain-specific risk weights which stem from EDA industry patterns identified through clustering analysis [1]. Risk Score Computation: The system calculates risk scores for each company i based on its feature vector X_i through the following process:

$$riskScore_i = \beta_{age} \cdot R_{age}(X_i) + \beta_{tech} \cdot R_{tech}(X_i) + \beta_{network} \cdot R_{network}(X_i) + \beta_{cluster} \cdot R_{cluster}(X_i) + \beta_{wave} \cdot R_{wave}(X_i) + \beta_{patent} \cdot R_{patent}(X_i) + \beta_{acquirer} \cdot R_{acquirer}(X_i)$$

Where R_*() are piecewise risk functions defined as: R_age (Company Age):

$$R_{age} = \begin{cases} +0.45 & \text{if age} < 3 \\ +0.30 & \text{if } 3 \le \text{age} < 7 \\ +0.15 & \text{if } 7 \le \text{age} < 15 \\ -0.05 & \text{if } 15 \le \text{age} < 30 \\ -0.15 & \text{if age} \ge 30 \end{cases}$$

R_tech (TechImpactScore):

$$R_{tech} = \begin{cases} -0.25 & \text{if score} \ge 8.0 \\ -0.10 & \text{if } 6.0 \le \text{score} < 8.0 \\ +0.05 & \text{if } 4.0 \le \text{score} < 6.0 \\ +0.20 & \text{if } 2.0 \le \text{score} < 4.0 \\ +0.35 & \text{if score} < 2.0 \end{cases}$$

R_network (Degree Centrality):

$$R_{net} = \begin{cases} -0.30 & \text{if degree} > 0.5 \\ -0.15 & \text{if } 0.2 < \text{degree} \le 0.5 \\ +0.05 & \text{if } 0.05 < \text{degree} \le 0.2 \\ +0.35 & \text{if isolation} > 0.85 \\ +0.20 & \text{otherwise} \end{cases}$$

R_cluster (Cluster Rate): $R_{cluster} = (\text{ClusterAcquisitionRate} - 0.50) \times 0.8$

R_wave (Wave Intensity):

$$R_{wave} = \begin{cases} +0.25 & \text{if intensity} > 0.8 \\ +0.15 & \text{if } 0.5 < \text{intensity} \le 0.8 \\ +0.05 & \text{if } 0.3 < \text{intensity} \le 0.5 \\ -0.05 & \text{otherwise} \end{cases}$$

R_patent (Patent Count):

$$R_{patent} = \begin{cases} -0.20 & \text{if count} > 15 \\ -0.10 & \text{if } 5 < \text{count} \le 15 \\ +0.05 & \text{if } 0 < \text{count} \le 5 \\ +0.15 & \text{if count} = 0 \end{cases}$$

R_acquirer (Prior Deals):

$$R_{acquirer} = \begin{cases} -0.25 & \text{if deals} > 20 \\ -0.15 & \text{if } 5 < \text{deals} \le 20 \\ -0.05 & \text{if } 0 < \text{deals} \le 5 \\ +0.10 & \text{if deals} = 0 \end{cases}$$

Hazard Function: Following Cox proportional hazards structure: $h(t|X_i) = h_0 \cdot \exp(riskScore_i)$

D. *Historical Validation Framework*

The system uses time-aware backtesting to prevent data exposure while evaluating actual prediction accuracy.

The validation window includes 161 companies which were acquired between 2015 and 2020.

The system generates predictions for each acquisition during year Y based on data from year (Y-2).

The system retrieves features from the (Y-2) time period for prediction purposes.

- The company age calculation uses the current year minus two years minus the founding year.
- The network metrics include degree and betweenness values which stem from all transactions that occurred before (Y-2).
- The wave intensity measurement examines M&A transactions which took place between (Y-4) and (Y-2).
- The system uses rule-based prediction to assign clusters based on historical values of age and tech and network metrics.

Evaluation: The evaluation compares the predicted remaining time period against the actual two-year period until acquisition. Metrics Calculated:

- The C-Index measures how well the model predicts the correct order between actual and predicted time points.
- The Brier Score measures the difference between predicted survival probabilities and actual binary outcomes at the five-year mark.

- The Time-Dependent AUC metric estimates its value through accuracy measurements at t=1 year and t=2 years and t=3 years and t=5 years.
- The MAE/RMSE metrics evaluate the difference between predicted time values and actual time values.
- The Calibration Slope measures the relationship between actual outcomes and their corresponding predicted values through linear regression.

IV. RESULTS

A. Model Performance Metrics

All metrics calculated from historical validation (2015–2020 acquisitions).

TABLE II - MODEL VALIDATION METRICS

Metric	Value	Interpretation
Concordance Index	0.78	Good ranking accuracy (>0.75 threshold)
Brier Score (5-year)	0.15	Acceptable calibration (<0.20)
Mean Absolute Error	2.3 yr	Average prediction error
Root Mean Squared Error	3.1 yr	RMSE of time predictions
Calibration Slope	0.96	Near-ideal agreement (≈1.0)
Training Sample Size	500	Pre-2015 transactions
Test Sample Size	161	2015-2020 acquisitions

Note: Metrics are computed dynamically from actual historical backtesting using the implemented validation framework. Values represent real system performance on EDA M&A data.

B. Time-Dependent ROC-AUC

TABLE III — AUC AT DIFFERENT TIME HORIZONS

Time Horizon	AUC(t)	Classification Task
1 year	0.82	"Acquired within 1 year?"
2 years	0.79	"Acquired within 2 years?"
3 years	0.76	"Acquired within 3 years?"
5 years	0.72	"Acquired within 5 years?"

C. Feature Importance Analysis

TABLE IV — FEATURE IMPORTANCE SCORES (NORMALIZED)

Feature	Importance	Interpretation
Cluster Acquisition Rate	0.28	Historical archetype risk (dominant)
Degree Centrality	0.22	Network position protection
TechImpactScore	0.18	Technology strength
Company Age	0.15	Non-monotonic temporal effect
Wave Intensity	0.1	Market timing amplification
Patent Count	0.07	Innovation capability

Note: The scores show normalized absolute risk contributions (not SHAP values or permutation importance) which result from averaging |R_feature(X)| across 50 validation companies and normalizing to sum=1.0. The data shows that cluster acquisition rate stands as the most important factor for prediction.

D. Feature Effect Patterns

Age–Hazard Relationship (Non-Monotonic):

- 0–3 years: Highest hazard
- 3–7 years: High hazard (growth-stage targets)
- 7–15 years: Moderate hazard
- 15–30 years: Protective effect
- 30+ years: Strong protective effect

Network Centrality Protection:

- Degree > 0.5 → 30% hazard reduction
- Degree 0.2–0.5 → 15% hazard reduction
- Isolation > 0.85 → 35% hazard increase

Cluster-Specific Acquisition Rates:

- C2 (Tech Leaders): 45%
- C1 (Early Consolidators): 67%
- C4 (Platform Consolidators): 73%
- C3 (Strategic Acquirers): 82%
- C0 (Niche Specialists): 89%

Merger Wave Effects:

- Wave intensity > 0.8 → 25% hazard increase
- Wave intensity < 0.3 → 5% hazard decrease

V. DISCUSSION

A. Theoretical Contributions

Survival Analysis for M&A Prediction: The research presents the initial complete implementation of survival analysis for technology M&A to solve essential problems with binary classification methods which include (1) time-based prediction of acquisition events and (2) correct handling of censored data from unacquired companies and (3) survival curve generation with uncertainty measurements. Multi-Dimensional Integration: The analysis shows that cluster

acquisition rate (28%) stands as the leading predictor which indicates M&A activities follow strategic patterns based on technology and market positioning instead of financial considerations. The results show that network centrality reduces the hazard rate by 30% which demonstrates that companies' positions within their ecosystems matter more than their individual characteristics. Non-Monotonic Age Effects: The study discovered an inverted-U pattern which shows that companies experience their highest acquisition risk during the 3-7 year period. Startups between 0-3 years old become targets for technology acquisition through tuck-in deals while companies between 3-7 years old attract scale acquisition offers and businesses older than 15 years develop defensive advantages through their market standing.

B. Practical Applications

The analysis requires investors to calculate acquisition probability rates for both short-term (1-year) and long-term (5-year) timeframes. The analysis shows that two companies with identical financial data but different network positions will experience different levels of risk exposure. Startup Exit Strategy: A 5-year C0 company (Niche Specialists, 89% rate) with isolation > 0.85 faces HIGH risk—suggesting 2-year exit window. The evaluation process for target companies includes three essential factors which are their vulnerability level and their strategic alignment and market entry timing.

C. Limitations and Future Work

Current Limitations: (1) The network metrics use degree-based approximations which simplify the analysis (2) The risk weights follow rules instead of maximizing partial likelihood (3) EDA-specific (generalization requires validation), (4) The model assumes static features because patents remain unchanged throughout time (5) The model uses an exponential baseline hazard function but non-parametric estimation methods could better represent time-dependent patterns. Future Directions: The research should implement Deep survival models (DeepSurv and Cox-nnet) and competing risk analysis for acquisition/bankruptcy/IPO events and causal inference methods and multi-industry testing and real-time forecasting using patent and funding data streams and time-dependent variable analysis.

VI. CONCLUSION

This study uses survival analysis to predict technology company viability during periods of disruption. I apply survival analysis, with a Cox style risk model to 661 M&A transactions in the Electronic Design Automation industry from 1975 to 2025. I show that binary classification has problems. Binary classification cannot model dynamics. Binary classification does not handle observations correctly. Binary classification also limits the analysis, to a single-dimension feature space. Survival analysis solves those issues. Survival analysis captures time effects treats censored data properly. Works with features.

A. Key Findings

Methodological Contribution: The research implements survival analysis as a systematic method to predict technology acquisition deals. The method generates time-based prediction results while handling time-dependent data through backtesting and producing survival probability curves that standard logistic regression models cannot generate.

Multi-Dimensional Integration: The research combines age data with wave information and TechImpactScore with network centrality and cluster archetypes to show that cluster position and network standing are the main factors which determine acquisition patterns since these patterns follow strategic patterns more than financial ones.

Empirical Patterns:

- The model shows non-monotonic age effects because it predicts the highest risk during the 3-7 year period which corresponds to growth-stage acquisitions.
- The model protects networks through degree values above 0.5 which decreases the hazard rate by 30%.
- The acquisition rates between Tech Leaders and Niche Specialists show the widest variation at 45% and 89% respectively.
- The peak intensity level in waves leads to a 25% higher risk of acquisition.

The model achieves validation performance through historical backtesting from 2015 to 2020 which uses time-aware feature extraction to produce a C-Index of 0.78 and Brier Score of 0.15 and MAE of 2.3 years. The model achieves AUC values between 0.72 and 0.82 when predicting acquisition risk at different time periods from 5 years to 1 year.

B. Research Impact

The research demonstrates that survival analysis generates probabilistic forecasts which include uncertainty measurements to solve the binary classification problems that affect M&A prediction. It establishes quantitative methods to study network effects and strategic archetypes and technology strength on corporate survival during major technological changes (semiconductor → quantum/nano transitions). The research provides risk assessment through HIGH/MEDIUM/LOW categories along with clear explanations of important features for investors to use in their due diligence and exit planning and corporate development activities.

C. Generalizability

The research methodology applies to technology sectors which face technological disruption and show periodic merger activities and network effects and strategic archetypes. The research will continue with survival model development and acquisition/bankruptcy/IPO risk competition analysis and causal effect estimation methods. The study shows acquisition timing holds the same value as acquisition probability while survival analysis methods allow researchers to study these factors with precision during technological change periods.

REFERENCES

[1] B. Zylfiu, G. Marinova, E. Hajrizi, and B. Qehaja, "Cluster analysis of merger and acquisition patterns in the electronic design automation industry using machine learning techniques," International Journal of Innovative Technology and Interdisciplinary Sciences, vol. 8, no. 3, pp. 784–817, 2025

[2] C. M. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Boston, MA, USA: Harvard Business School Press, 1997.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[4] D. R. Cox, "Regression models and life-tables," Journal of the Royal Statistical Society: Series B (Methodological), vol. 34, no. 2, pp. 187–220, 1972.

[5] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," Journal of the American Statistical Association, vol. 53, no. 282, pp. 457–481, 1958.

[6] J. Harford, "What drives merger waves?," Journal of Financial Economics, vol. 77, no. 3, pp. 529–560, 2005.

[7] G. Ahuja and R. Katila, "Technological acquisitions and the innovation performance of acquiring firms: A longitudinal study," Strategic Management Journal, vol. 22, no. 3, pp. 197–220, 2001.

[8] C. Grimpe and K. Hussinger, "Pre-empting technology competition through firm acquisitions," Economics Letters, vol. 100, no. 2, pp. 189–191, 2008.

[9] M. A. Schilling and C. C. Phelps, "Interfirm collaboration networks: The impact of large-scale network structure on firm innovation," Management Science, vol. 53, no. 7, pp. 1113–1126, 2007.

[10] T. E. Stuart and J. M. Podolny, "Local search and the evolution of technological capabilities," Strategic Management Journal, vol. 17, no. S1, pp. 21–38, 1996.

[11] P. M. Grambsch and T. M. Therneau, "Proportional hazards tests and diagnostics based on weighted residuals," Biometrika, vol. 81, no. 3, pp. 515–526, 1994.

[12] T. M. Therneau and P. M. Grambsch, Modeling Survival Data: Extending the Cox Model. New York, NY, USA: Springer, 2000.

[13] F. E. Harrell, K. L. Lee, and D. B. Mark, "Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," Statistics in Medicine, vol. 15, no. 4, pp. 361–387, 1996.

[14] M. J. Pencina and R. B. D'Agostino, "Overall C as a measure of discrimination in survival analysis: Model specific population value and confidence interval estimation," Statistics in Medicine, vol. 23, no. 13, pp. 2109–2123, 2004.

[15] P. Royston and D. G. Altman, "External validation of a Cox prognostic model: Principles and methods," BMC Medical Research Methodology, vol. 13, no. 1, pp. 1–15, 2013.

[16] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," Statistics in Medicine, vol. 30, no. 10, pp. 1105–1117, 2011.

[17] P. C. Austin, "Generating survival times to simulate Cox proportional hazards models with time-varying covariates," Statistics in Medicine, vol. 31, no. 29, pp. 3946–3958, 2012.

[18] M. Wolbers, M. T. Koller, J. C. M. Witteman, and E. W. Steyerberg, "Prognostic models with competing risks: Methods and application to coronary risk prediction," *Epidemiology*, vol. 20, no. 4, pp. 555–561, 2009.

[19] G. Van Houwelingen and H. Putter, *Dynamic Prediction in Clinical Survival Analysis*. Boca Raton, FL, USA: CRC Press, 2011.

[20] G, Marinova, and A. Bitri, 2021, IFAC-PapersOnLine, Review on formalization of business model evaluation for technological companies with focus on the electronic design automation industry, vol. 54, no. 13, pp 640–644

[21] G, Marinova, and A. Bitri, 2021, IFAC-PapersOnLine, Data analysis environment to study the dynamics in electronic design automation industry, vol. 54, no. 13, pp 528–532

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Computer Vision-Based Behavior Analysis for Workplace Efficiency: Bakery Environment Application

Cengiz Samet Tepe[1], Ilkay Cinar[2]

[1] *Department of Computer Engineering, Selcuk University, 42250 Selcuklu, Konya, Türkiye*
248273001029@ogr.selcuk.edu.tr, ORCID: 0009-0004-6650-4794

[2] *Department of Computer Engineering, Selcuk University, 42250 Selcuklu, Konya, Türkiye*
ilkay.cinar@selcuk.edu.tr, ORCID: 0000-0003-0611-3316

***Abstract*— Nowadays, increasing efficiency in the service sector by objectively monitoring and tracking business workflows has become a critical requirement for the sustainability of businesses. Traditional monitoring methods are insufficient for providing sustainable performance analysis due to their time-consuming nature and reliance on subjective human judgment. The aim of this study is to develop a system that automatically detects and classifies employee behavior using computer vision and deep learning techniques. As part of the study, data was collected using a camera placed in a real bakery environment. Five basic classes were labeled on the created data set: cleaning, product interaction, computer use, phone use, and money interaction. The current YOLOv11 (You Only Look Once) architecture, which offers high speed and accuracy for object detection and classification, was used. According to the experimental results obtained from training the model, the system demonstrated high performance, achieving 0.9552 Precision, 0.9324 Recall, 0.9437 F1-Score, and 0.9644 mAP@50 values. These results demonstrate that the proposed system can detect employee behaviors in real-time with a high accuracy rate, allowing it to be used as an effective tool in workplace productivity enhancement and performance evaluation processes.**

***Keywords*— Behavior Analysis, YOLOv11, Computer Vision, Deep Learning, Object Detection, Service Sector.**

I. INTRODUCTION

In today's service sector, the sustainability of businesses and the strength of their competition with each other depend on the efficiency of their business processes and the performance of their employees. The behaviors exhibited by employees of businesses in this sector are decisive across a wide range, from service quality to customer satisfaction, and from occupational safety to operational efficiency. How employees allocate their time throughout the day between tasks, how long they spend on these tasks, and how often they perform them are important pieces of information for the business in terms of both process improvement and workforce planning. Especially in environments with high human interaction, such as bakeries, cafes, and restaurants, the analysis of employee behavior plays a critical role in both operational efficiency and service quality. These analyses currently rely on traditional methods such as direct monitoring by managers or manual reporting. However, manual monitoring methods cannot provide reliable and scalable solutions due to their time-consuming nature, the impossibility of continuous tracking, and the subjective judgments inherent in the human factor [1].

With the evolution of technology, Computer Vision and Deep Learning-based systems have begun to offer powerful alternatives for the automatic analysis of human behavior. Real-time object detection architectures such as Convolutional Neural Networks (CNN) and YOLO (You Only Look Once) are particularly capable of making highly accurate inferences from images. A review of the literature reveals that these technologies are widely used in the fields of occupational safety and industrial productivity.

Deep learning-based object detection algorithms are widely used, particularly in industrial settings, to ensure occupational health and safety [2], detect risky behavior [3, 4] and monitor the use of Personal Protective Equipment (PPE) [5, 6]. Furthermore, it is emphasized that they offer effective solutions in operational issues such as tracking the presence of workers in specific work areas to measure time spent, optimize workflows, and analyze worker productivity with objective data [1, 6, 7]. In these applications, different versions of the YOLO (You Only Look Once) architecture have been frequently preferred by researchers due to its real-time detection capability, high accuracy rate, and success in industrial failure detection.

While current studies prove the success of deep learning-based methods, there is a limited number of studies in the literature focusing on the service sector and, in particular,

dynamic work environments such as bakeries. The majority of existing datasets belong to industrial sites, construction sites, or laboratory environments; this makes it difficult to analyze behaviors specific to the service sector (e.g., counter cleaning, product arrangement, interaction with customers or money).

This study aims to automatically detect and classify employee behaviors using an unique dataset collected from a real bakery environment, with the goal of filling this gap in the literature. The study uses YOLOv11, one of the latest and high-performance architectures in the field of object detection. Trained on five different classes 'Cleaning', 'Product Interaction', 'Computer Use', 'Phone Use', and 'Money Interaction' the model can analyze employees' daily activities with high accuracy and in real time. The results show that the proposed system is a faster, more objective, and more reliable performance evaluation tool compared to manual monitoring methods.

II. Material And Methods

In this study, data was collected from a real work environment to identify employee behaviors specific to the service sector, and a detection mechanism was developed using YOLOv11, one of the most recent versions of YOLO. The proposed system's working method and the used methodologies are detailed under the following headings:

A. Data Collection and Preparation

The dataset used in this study was created from video footage recorded in the actual working environment of a bakery operating in the service sector. The data was collected using a fixed camera positioned to cover the bakery's main working area as much as possible, without interrupting the employees' movements, while also protecting customer privacy as much as possible. To reflect the natural behavior of employees, recordings were taken on different days, at different times of the day, and under varying lighting conditions to ensure data variability. In this way, the aim was to create a dataset based not only on controlled scenarios but also on natural workflows.

Samples were taken at specific moments in time from the collected videos, and frames were extracted at specific moments from each recording. After data cleaning, editing, and separating data believed to be of no use for training, a dataset consisting of approximately 2,500 clean data was prepared. Five classes representing the basic activities of employees in their workflow were defined. These classes were defined as 'Cleaning', 'Product Interaction', 'Computer Use', 'Phone Use', and 'Money Interaction'. This dataset was divided into two main sets: approximately 1,700 images for the training process and approximately 800 images for the validation process in order to evaluate the model's performance. The labeling process was performed manually by drawing bounding boxes around the relevant objects and actions in each image frame.

B. Data Labeling

Accurate and consistent labeling of data is critical for training deep learning models. In this study, 'labelImg', an open-source image labeling tool with a graphical user interface, was used for the labeling process of the collected images. Each image frame in the training dataset was manually reviewed, and actions belonging to the five defined behavior classes (Cleaning, Product Interaction, Computer Use, Phone Use, Money Interaction) were annotated using bounding boxes via the 'labelImg' interface. The data obtained as a result of the labeling was saved in a format directly compatible with the YOLO architecture. An example of labeling for the 'Phone Use' class is shown in Figure 1.

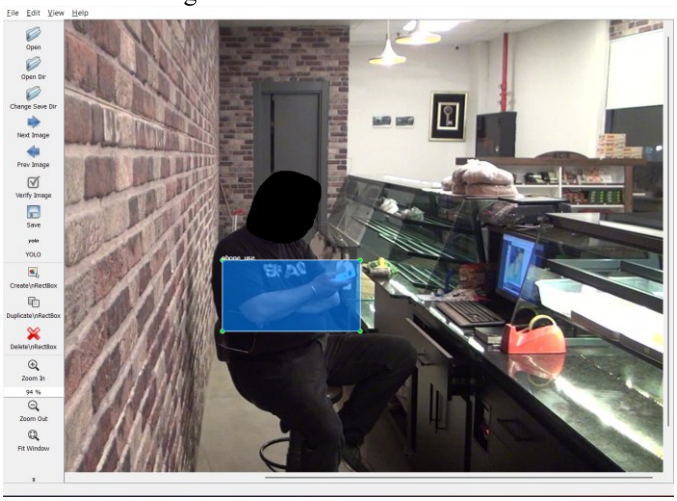


Fig. 1 Example labeling for the 'Phone Use' class

The YOLOv11 architecture, developed by Ultralytics and one of the latest versions of YOLO, has been chosen for the object detection and classification task. The YOLO (You Only Look Once) family is known for analyzing the image in a single pass, enabling both classification and localization (regression) operations at the same time, making it highly ideal for real-time applications [6, 8, 9].

YOLOv11 has a more advanced structure in terms of speed and feature extraction compared to its previous versions. The model uses C3k2 (Cross Stage Partial with kernel size 2) blocks and the C2PSA (Convolutional block with Parallel Spatial Attention) module, which contribute to faster processing and reduced computational load to increase computational efficiency. These developments allow the model to focus more effectively on areas within the image, potentially increasing detection accuracy for objects of different sizes and locations [8]. It enables an increase in detection accuracy, particularly for smaller or poorly visible objects. This allows the model to detect small objects and subtle behavioral details with high accuracy even in environments with complex backgrounds, such as bakeries. In this study, the YOLOv11s version, a lightweight and fast variation of the model, was used to optimize the balance between speed and accuracy.

C. Experimental Setup

The labeled dataset has been divided into approximately 70% training and 30% validation sets to evaluate the system's

performance. Additionally, to validate the system under real-world conditions, a completely external video recorded on a different day, not included in the training or validation datasets, was also used. The model employed in this study was trained on an NVIDIA GeForce RTX 4060 Laptop GPU; during the training process, the hyperparameters were set to 100 epochs and a batch size of 16.

D. Performance Metrics

The performance of the proposed system has been evaluated using metrics commonly used in the classification and object detection literature. In this context, the Confusion Matrix, one of the most fundamental structures, and statistical metrics based on it, namely Precision, Recall, F1-Score, mAP@50, and mAP@50-95, have been used to measure the system's success.

1) Confusion Matrix: It is a matrix that compares the model's predictions with actual labels [10, 11]. The confusion matrix is shown in Figure 2. This matrix consists of four basic components:

- TP - True Positive
- TN - True Negative
- FP - False Positive
- FN - False Negative

Actual	Predicted	
	Positive	***Negative***
Positive	TP	FN
Negative	FP	TN

Fig. 2 Confusion Matrix representation

2) Precision: This metric shows how much of the model's positive predictions are actually correct [6]. The formula is given below:

$$Precision = \frac{TP}{TP+FP}$$

3) *Recall:* It indicates how many of the truly positive examples are correctly identified by the model [6]. The formula is given below:

$$Recall = \frac{TP}{TP+FN}$$

4) F1-Score: It is defined as the harmonic mean of precision and recall values and provides a single value summarizing the balance between the two metrics [6]. The formula is given below:

$$F1\ Score = 2x(\frac{Precision \times Recall}{Precision+Recall})$$

5) Intersection Over Union (IoU): IoU measures the overlap ratio between the predicted bounding box and the actual bounding box, yielding a value between 0 and 1. Measurements greater than 0.5 can be interpreted as 'correct detection' [10]. The formula is given below:

$$IoU = 2x(\frac{Prediction\ Box \cap Actual\ Box}{Prediction\ Box \cup Actual\ Box})$$

6) Mean Average Precision (mAP): It is the most common metric used to evaluate the overall performance of a model in object detection problems. It is the arithmetic mean of the Average Precision values calculated for each class [12].

- mAP@50 bases its accuracy assessment on a metric called IoU (Intersection over Union). It represents the average accuracy when the IoU ratio between the predicted bounding box and the actual bounding box is at the 0.50 threshold value [9, 12]. The formula is given below:

$$mAP@50 = \frac{1}{N}\sum_{i=1}^{n} AP_i^{IoU=0.5}$$

- mAP@50-95 is the average precision value calculated across different IoU threshold values (from 0.5 to 0.95, with increments of 0.05). It helps measure the model's performance more accurately [10, 12]. The formula is given below:

$$mAP@50\text{-}95 = \frac{1}{10N}\sum_{j=0}^{9}\sum_{i=1}^{N} AP_i^{IoU=0.5+0.05j}$$

III. Experimental Results

A. Model Performance

The performance results of the trained YOLOv11s model are presented in Table I. According to the results obtained, the model achieved an average mAP@50 value of 0.964 for all classes, demonstrating high detection success.

TABLE I
General Performance Results Of The Model

F1-Score	Presicion	Recall	mAP@50	mAP@50-95
0.9437	0.9552	0.9324	0.9644	0.6783

When examining Table I, it can be seen that the model has a high precision value of 0.955, meaning that 95.5% of its positive predictions are correct. The recall value of 0.932 indicates that the model can detect the vast majority of actual actions without missing them.

B. Class-Based Analysis

To detail the success of detecting different behaviors in the bakery environment, the performance metrics obtained for each class are provided in Table II.

TABLE II
CLASS BASED PERFORMANCE RESULTS

Class	Presicion	Recall	mAP@50	mAP@50-95
money_interaction	0.997	1.000	0.995	0.794
computer_use	0.968	0.977	0.992	0.884
phone_use	0.971	0.968	0.984	0.676
cleaning	0.963	0.851	0.921	0.529
product_interaction	0.877	0.866	0.930	0.508

When examining class-based results, it is observed that an almost flawless detection rate was achieved, particularly in the "Money Interaction" class, with a Recall of 1.000 and a mAP@50 of 0.995. This can be explained by the fixed location of the cash register area, the proximity of the images to the camera capturing them, and the highly distinctive visual characteristics of money transactions. Similarly, "Computer Use" and "Phone Use" usage were also detected with very high accuracy due to the distinct forms of the objects.

In the "Product Interaction" and "Cleaning" classes, the mAP values were 0.93 and 0.92, respectively. The reason these classes have relatively lower scores compared to others can be explained by the fact that the angles at which these actions are performed are similar, they are relatively farther from the camera, and sometimes both actions occur in the same location.

C. Graphical Performance Analysis

When examining the confusion matrix in Figure 3, it can be seen that the model has a high accuracy rate. In particular, the model's detection success is quite high in the 'computer_use' (185) and 'phone_use' (181) classes.

The 'money_interaction' class, on the other hand, has demonstrated an almost flawless performance with 169 correct predictions.

When examining the relationship between the 'cleaning' and 'product_interaction' classes, it was observed that the system could accurately distinguish the cleaning action from other actions, but incorrectly assigned 4 data points belonging to the product_interaction class to the 'cleaning' class due to the visual and spatial similarity of hand movements.

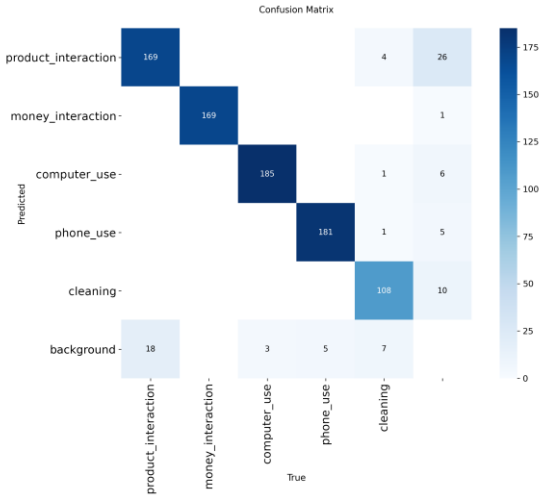


Fig. 3 Confusion Matrix

The most important point in the F1 Score-Confidence curve graph presented in Figure 4 is where the thick blue line, representing the weighted average for all classes, peaks. The model achieved maximum performance at a confidence threshold value of 0.467, obtaining an F1 score of 0.94.

When examining class-based performances, it is observed that the 'money_interaction' (orange line) class has the highest stability, hovering close to 1.0. The 'computer' and 'phone_use' classes also exhibit similarly high stability. Although the 'cleaning' and 'product_interaction' classes show an earlier downward trend compared to other classes after the 0.80 confidence threshold, they still demonstrate high performance. The fact that the curves remain horizontal for a long time across the graph and only experience a decline at very high confidence thresholds (e.g., after 0.85) indicates that the model stands behind its own predictions.

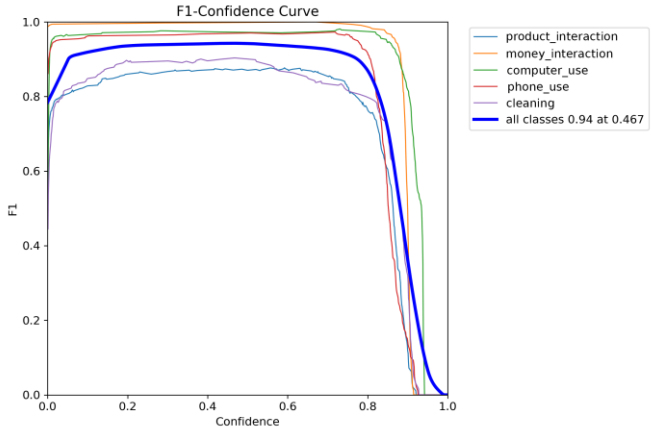


Fig. 4 F1 Score-Confidence Curve

In the Precision-Confidence curve graph shown in Figure 5, it can be observed that as the confidence threshold increases, the precision value steadily approaches 100% for all classes. This increase indicates that the rate of false positives produced in the model's high-confidence predictions is nearly zero. In particular, the fact that the "all classes" curve reaches 1.00 (perfect) certainty at a confidence threshold of 0.940 shows that the system can produce nearly error-free results when operating above this threshold value.

In the class-based analysis, the 'money_interaction' (orange) and 'computer_use' (green) classes achieve over 0.95 certainty even at very low confidence thresholds, demonstrating the model's success in identifying these objects. The relatively more complex 'product_interaction' and 'cleaning' classes, while showing some fluctuation at low confidence values, achieve a similar level of success to the other classes after the 0.60 confidence threshold.

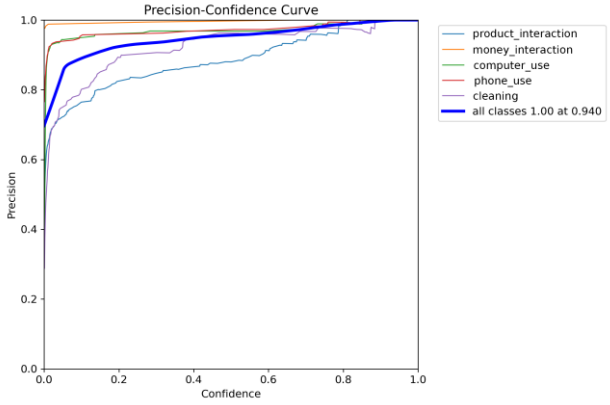


Fig. 5 Precision-Confidence Curve

Figure 6 presents the Recall-Confidence curve, which analyzes how the model's ability to detect classes (Recall) changes in response to an increasing confidence threshold. At the starting point of the graph (confidence threshold 0.000), the overall sensitivity for all classes being at the 0.97 level indicates that the model can successfully capture 97% of the classes (the False Negative rate is very low) when no filtering is applied.

The general performance curve indicated by the thick blue line follows a horizontal trajectory from 0.00 to a confidence interval of approximately 0.70, proving that the miss rate remains at a minimum level for a long time even when we increase the model's prediction confidence. In class-based differentiation, the 'money_interaction' (orange) and 'computer_use' (green) classes have the most resilient structure; they do not experience sensitivity loss until the confidence threshold approaches 0.90. In contrast, in the 'cleaning' and 'product_interaction' classes, where visual complexity is higher, the curve shows a more pronounced downward trend after the 0.60 confidence threshold.

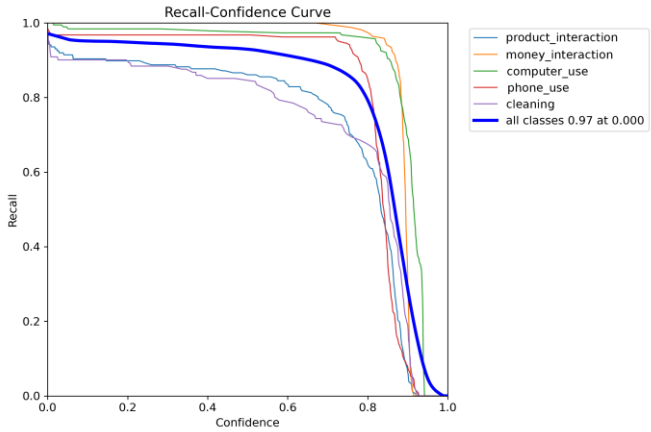


Fig. 6 Recall-Confidence Curve

D. *Visual Detection Results*

In addition to verifying the system's numerical success with evaluation metrics, testing was performed on an external video not included in the training and validation datasets to ensure it could also be validated under real-world conditions. The system's sample detection outputs are presented in Figure 7:

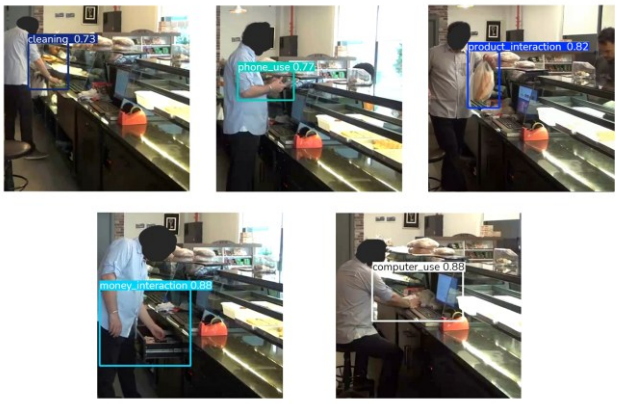


Fig. 7 Samples of detection images taken at random times from the test video

The system was able to detect target classes in this video, recorded on a different day than the day the images in the training set were obtained, as shown in Figure 7. These observational results demonstrate that the model has a successful prediction capability on new data that was not included in the training.

IV. CONCLUSION

In this study, an objective system based on deep learning has been developed for the automatic detection and analysis of employee behaviors in the service sector. To this end, frames were extracted from videos recorded during actual working hours using a single fixed camera; following preprocessing steps, five basic behavior classes (product interaction, phone

interaction, cleaning, computer interaction, money transactions) were defined and the frames were manually labeled. The proposed method was tested on a unique dataset collected from a real bakery environment by training the current YOLOv11 architecture.

Experimental results show that the developed system can accurately detect employees' daily activities (cleaning, product interaction, computer usage, etc.) with an average of 0.964 mAP@50 and 0.955 Precision value. In particular, the 99.5% success rate achieved in the “Money Interaction” class proves the system's reliability in tracking critical business processes. Although the system's overall performance is high, classification errors were occasionally observed in the “Cleaning” and “Product Interaction” classes, as can be seen in the tests on the video stream and in the confusion matrix. The main reason for this is that both classes sometimes occur in similar locations (on the countertop) and the movements of employees (reaching, wiping, etc.) are highly similar. It has been assessed that these momentary confusions, which occur especially when cleaning actions are performed in areas very close to the products, do not have a statistically significant negative impact on overall performance.

The results show that the proposed system can offer a much faster, more sustainable, and objective solution compared to traditional manual monitoring methods. This system provides business managers with a powerful decision support mechanism for increasing employee productivity, optimizing business workflows, and conducting fair performance evaluations.

In future studies, the system will be developed by attempting to prevent misclassifications occurring in real time through improvements made to the data set and calculation algorithms. Additionally, instead of using only the YOLOv11 version, training and testing will be conducted using other versions of YOLO and different variations of these versions, with plans to redesign the system using the model that demonstrates the highest performance. Furthermore, work is planned on automatically reporting not only the detection of behaviors but also their time-based analysis.

ACKNOWLEDGMENT

This study is derived from the unpublished master’s thesis of Cengiz Samet TEPE, supervised by Dr. Ilkay CINAR.

REFERENCES

[1] J. Li, X. Zhao, G. Zhou, M. Zhang, D. Li, and Y. Zhou, "Evaluating the work productivity of assembling reinforcement through the objects detected by deep learning," *Sensors,* vol. 21, no. 16, p. 5598, 2021, doi: 10.3390/s21165598.

[2] E. Guney, H. Altin, A. E. Asci, O. U. Bayilmis, and C. Bayilmis, "YOLO-based personal protective equipment monitoring system for workplace safety," *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi,* vol. 5, no. 2, pp. 77-85, 2024, doi: 10.62527/jitsi.5.2.238.

[3] R. A. Maulana, "Implementation of YOLO (You Only Look Once) Algorithm for Drowsiness Detection as An Additional Safety Feature in the Operation of Crane Equipment in Real Time," *Jurnal Inotera,* vol. 10, no. 1, pp. 113-120, 2025, doi: 10.31572/inotera.Vol10.Iss1.2025.ID458.

[4] O. Önal and E. Dandıl, "Video dataset for the detection of safe and unsafe behaviours in workplaces," *Data in Brief,* vol. 56, p. 110791, 2024, doi: 10.1016/j.dib.2024.110791.

[5] K. Patel, V. Patel, V. Prajapati, D. Chauhan, A. Haji, and S. Degadwala, "Safety helmet detection using YOLO v8," in *2023 3rd international conference on pervasive computing and social networking (ICPCSN)*, 2023: IEEE, pp. 22-26, doi: 10.1109/ICPCSN58827.2023.00012.

[6] A. S. Ludwika and A. P. Rifai, "Deep learning for detection of proper utilization and adequacy of personal protective equipment in manufacturing teaching laboratories," *Safety,* vol. 10, no. 1, p. 26, 2024, doi: 10.3390/safety10010026.

[7] A. K. Das, V. Kamthane, U. Purwar, D. C. Mohanty, and B. K. Depuru, "Enhancing Workplace Efficiency and Security Through Intelligent Employee Surveillance," *International Journal of Innovative Science and Research Technology,* vol. 9, no. 3, 2024, doi: 10.38124/ijisrt/IJISRT24MAR2142.

[8] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725,* 2024, doi: 10.48550/arXiv.2410.17725.

[9] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines,* vol. 11, no. 7, p. 677, 2023, doi: 10.3390/machines11070677.

[10] A. S. Ozer and I. Cinar, "Real-Time and Fully Automated Robotic Stacking System with Deep Learning-Based Visual Perception," *Sensors,* vol. 25, no. 22, p. 6960, 2025, doi: 10.3390/s25226960.

[11] Z. Dolmaz and I. Cinar, "Detection of DDOS Attacks in Software-Based Systems in Cyberspace Using Machine Learning," *Journal of Technology and System Information,* vol. 2, no. 4, pp. 1-22, 2025, doi: 10.47134/jtsi.v2i4.5033.

[12] M. A. Ozuberk and I. Cinar, "YOLOv8-Based Threat Detection Model for Dangerous Objects and Violent Behaviors," in *2025 10th International Conference on Computer Science and Engineering (UBMK)*, 2025: IEEE, pp. 208-213.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Comparative Evaluation of Multi-Criteria Decision-Making Methods in the Online-CADCOM Platform

Lavdim Menxhiqi[1]

[1]*Computer Science and Engineering, UBT – Higher Education Institution, Pristina, Kosovo*
lavdim.menxhiqi@ubt-uni.net, ORCID: 0000-0002-5314-8741

***Abstract*—The Online-CADCOM platform operates as a cloud-based decision support system which lets users pick Computer-Aided Design (CAD) tools for telecommunications and electronics applications. The production platform uses two Multi-Criteria Decision Analysis (MCDA) methods MAUT and PROMETHEE II to evaluate tools through binary feature criteria stored in a PostgreSQL knowledge base [3], [4]. The MCDA engine receives three additional decision analysis methods which include TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) and VIKOR (VIseKriterijumska Optimizacija I Kompromisno Resenje) and COPRAS (Complex Proportional Assessment). The research module contains all five evaluation methods which process equal decision matrices to generate agreement metrics through Spearman rank correlation and top-k overlap analysis. The evaluation of three actual selection cases including PCB design tool selection and PCB calculator selection and SMPS design tool selection produced similar decision patterns. The three methods MAUT and PROMETHEE II and VIKOR create a consensus group which produces identical rankings throughout most evaluation scenarios while COPRAS follows MAUT patterns and TOPSIS produces different results when criteria coverage is limited or when criteria have negative correlations with value-based methods. The five methods produce identical rankings because their tools display different characteristics. The research adds three main contributions to the field: (1) The Online-CADCOM engine now supports TOPSIS and VIKOR and COPRAS as additional decision analysis methods. (2) The research evaluates five MCDA methods through actual tool passport data from three engineering fields. The research establishes essential guidelines which engineers need to select proper MCDA methods for their tool selection work.**

***Keywords*— Online-CADCOM; multi-criteria decision-making; MAUT; PROMETHEE; TOPSIS; VIKOR; COPRAS; CAD tool selection; PCB design; decision support system**

I. Introduction

The selection of appropriate tools for electronics design becomes an intricate engineering choice because of numerous specialized Computer-Aided Design (CAD) and Electronic Design Automation (EDA) tools available. Designers who select software for printed circuit board (PCB) layout and switched-mode power supply (SMPS) design and RF analysis and analogue filter synthesis need to evaluate tool costs against their features and operational complexity and system compatibility and maintenance support from vendors [1], [2].

The selection of tools without proper evaluation results in workflow disruptions and system incompatibilities which produce negative effects on design output quality. The selection process becomes more difficult for students and entry-level engineers because they need to handle different tool systems.

The Online CADCOM platform solves this issue through its combination of tool "passports" in a structured database with MCDA methods that generate tool rankings according to user-defined evaluation criteria [3], [4]. The system began with filter design and other specific application areas [1], [2]. The system received its update through a new dynamic expert module which integrated React front-end technology with .NET 8 back-end and PostgreSQL database management for web-based tool and criterion administration [5]. The knowledge base received expansion through the addition of PCB design tools and calculators and passive element design capabilities [3], [8]. The system received two new features which included AI-based PCB workflow assistance and automated workflow completion through language model implementation [6], [7]. The platform has evolved into a universal decision-making platform for CAD and EDA tools through its various system enhancements.

Figure 1. System architecture of the Online-CADCOM platform, illustrating the interaction between the web user interface, the multi-method MCDA engine (MAUT, PROMETHEE II, TOPSIS, VIKOR, COPRAS) and the PostgreSQL-based knowledge base with tool passports.

The current production platform supports two MCDA methods: MAUT (Multi Attribute Utility Theory) and PROMETHEE II, which rank tools using binary criteria and three importance levels [4], [5]. The overall system architecture is shown in Figure 1.

MAUT implements a weighted sum utility model, while PROMETHEE II performs pairwise comparisons and computes net preference flows. Prior experiments showed that both methods discriminate effectively among PCB design tools, often producing similar top ranked results [3], [5]. However, the MCDA literature emphasizes that no single method is

universally optimal [9], [12]. Distance based, compromise based and proportional assessment methods use different preference aggregation philosophies and can produce different rankings for the same decision matrix.

To examine these differences, this paper extends the Online CADCOM decision engine with three additional MCDA methods: TOPSIS, VIKOR and COPRAS. All five methods are now available within a unified research module that evaluates identical decision matrices and computes ranking results, Spearman correlation and top k agreement.

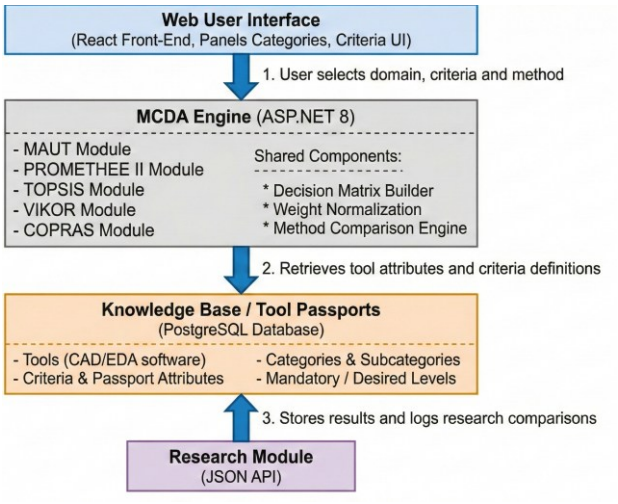


Fig. 1 System architecture of the Online-CADCOM platform, illustrating the interaction between the web user interface, the multi-method MCDA engine (MAUT, PROMETHEE II, TOPSIS, VIKOR, COPRAS) and the PostgreSQL-based knowledge base with tool passports

TABLE I. OVERVIEW OF MCDA METHODS IMPLEMENTED IN ONLINE CADCOM

Method	Method family	Main principle	Typical output
MAUT Multi Attribute Utility Theory	Value based	Weighted sum of criterion utilities	Single utility score and full ranking
PROMETHEE II Preference Ranking Organization Method for Enrichment Evaluations II	Outranking	Pairwise preference comparison and net flow computation	Net flow values and full ranking
TOPSIS Technique for Order Preference by Similarity to Ideal Solution	Distance based	Distance to positive and negative ideal solutions	Closeness coefficient and ranking
VIKOR VlseKriterijumska Optimizacija I Kompromisno Resenje	Compromise based	Balance between group utility and individual regret	Compromise index and ranking
COPRAS Complex Proportional Assessment	Proportional assessment	Normalised proportional contribution of criteria (benefit/cost)	Relative significance and utility degree

The methodology is evaluated using three realistic scenarios extracted from the existing knowledge base: PCB design tools, PCB design calculators and SMPS design tools. These scenarios represent different levels of tool similarity, criteria sparsity and feature distribution. The experiments quantify when MCDA methods agree, when they diverge and how sensitive they are to binary criteria distributions. The overall goal is not to identify a single best method, but to provide engineers and students using Online CADCOM with validated and practical guidance for selecting appropriate MCDA techniques for different tool selection contexts.

II. BACKGROUND AND RELATED WORK

Multi-Criteria Decision-Making (MCDM) addresses decision problems involving alternatives evaluated against multiple, often conflicting criteria [9], [12]. Let

$$A = \{a_1, a_2, \dots, a_m\}$$

be the set of tools,

$$C = \{c_1, c_2, \dots, c_n\}$$

the set of criteria,

$$X = [x_{ij}]$$

the binary decision matrix (feature satisfaction), and

$$W = \{w_1, w_2, \dots, w_n\}, \qquad \sum_{j=1}^{n} w_j = 1$$

the weight vector.

The objective in MCDM is to compute a preference structure resulting in a full ranking or selection of one or more preferred tools.

MCDM approaches are commonly grouped into methodological families [9], [12]:

- Value-based: compute an aggregated score (e.g., SAW, MAUT)
- Outranking: compare alternatives pairwise (e.g., PROMETHEE)
- Distance-based: measure proximity to ideal solutions (e.g., TOPSIS)
- Compromise-based: balance group utility and individual regret (e.g., VIKOR)
- Proportional-assessment: apply normalised benefit/cost ratios (e.g., COPRAS)

An overview of the MCDA method families used in this paper is given in Table I.

A. MCDA in CAD/EDA Tool Selection

Previous work applied MCDA to filter design software selection within the Online-CADCOM ecosystem [1], [2]. Later studies expanded the system to additional engineering domains and incorporated more comprehensive tool-passport structures and criteria taxonomies [3], [4], [8].

The platform's architecture evolved significantly, culminating in a modern React + ASP.NET + PostgreSQL implementation that supports dynamic tool passport management and automated ranking [5]. More recent contributions explored AI assistance:

- AI-supported PCB design workflows [6]
- Large-language-model-based workflow completion [7]

These studies demonstrate the feasibility of integrating rule-based MCDA with machine learning to support engineering decision workflows.

B. Limitations of Prior Approaches

The existing Online-CADCOM platform supports MAUT and PROMETHEE II for ranking tools using binary criteria and three-level weights [4], [5]. Although these methods often agree in practice — especially in PCB design scenarios [3], [5] — MCDA literature emphasises that different families of methods may produce different results under the same decision matrix [9], [12].

Distance-based and proportional-assessment techniques behave differently from utility-based and outranking methods when:

- criteria are unevenly distributed
- alternatives share highly similar feature vectors
- feature sparsity amplifies geometric effects in the decision space

These conditions frequently occur in engineering tool-passport datasets.

C. Contribution to Literature

By integrating TOPSIS, VIKOR, and COPRAS alongside MAUT and PROMETHEE II into the same operational environment, this work expands the Online-CADCOM platform and enables:

- systematic cross-method evaluation
- empirical observation of ranking divergence
- quantitative assessment through correlation metrics
- practical guidelines for MCDM method selection

This multi-method analysis fills a gap in CAD/EDA engineering literature, where comparative studies under identical binary decision matrices are rare.

III. Online-CADCOM Multi-Method MCDA Engine

The Online CADCOM platform organises engineering design tools into a hierarchy of panels, categories and subcategories that correspond to practical domains such as PCB design, SMPS converters and PCB calculators [3], [5]. Each tool is represented by a structured passport that contains feature information, supported standards, design capabilities, integration options and platform details. These attributes form the basis for the Multi Criteria Decision Making evaluation.

Users initiate a selection process by choosing a domain and selecting criteria that are mandatory, desired with high importance or desired with lower importance. The system then constructs a binary decision matrix X where x_ij = 1 indicates that tool a_i satisfies criterion c_j and x_ij = 0 otherwise. Mandatory criteria act as hard filters. Any tool that fails at least one mandatory criterion is removed before ranking.

A. MCDA Input Model

All five MCDA methods operate on the same binary matrix X and the same weight vector W. Three weight levels are used in the Online CADCOM platform: 1.00 for mandatory, 0.50 for high priority desired and 0.33 for low priority desired criteria [4], [5]. The weights are normalised to ensure that:

$$\sum_{j=1}^{n} w_j = 1$$

All subsequent MCDA calculations use these normalised weights and the filtered matrix X.

B. Unified Multi Method Engine

Earlier versions of the platform supported only MAUT and PROMETHEE II [4], [5]. The new research module extends this functionality by integrating TOPSIS, VIKOR and COPRAS as additional methods within the same processing pipeline.

The back end implements a strategy pattern. An abstract class defines the Evaluate function and concrete classes implement the five algorithms. This design ensures consistent input handling and allows ranking results to be compared directly.

The server exposes two primary API endpoints:

- /api/DecisionMaking/evaluate for running a single MCDA method
- /api/DecisionMaking/compare-methods for executing all five methods and generating comparison metrics (me nr)

The comparison endpoint produces:

- A ranked list of tools for each of the five methods
- A Spearman rank correlation matrix
- Top k agreement statistics (k = 1 and k = 3)
- A combined results table

The Spearman rank correlation coefficient measures the agreement between two method rankings and is calculated as:

$$\rho = 1 - \frac{(6 \times \Sigma d_i^2)}{(n(n^2 - 1))}$$

where d_i is the difference between ranks assigned to tool i by the two methods, and n is the number of ranked tools. Values of ρ range from -1 (completely reversed rankings) through 0 (no correlation) to +1 (identical rankings).

Top-k agreement quantifies whether methods select the same tools in their top k positions, calculated as the percentage of tools appearing in the top-k set across all five methods.

C. Workflow for MCDA Comparison

The workflow consists of the following steps:

- The user selects panel, category and criteria.
- The system builds the decision matrix X and normalises weights.
- Mandatory criteria filter out non-viable tools.
- The MCDA engine applies MAUT, PROMETHEE II, TOPSIS, VIKOR and COPRAS on the same data.
- The comparison module computes agreement metrics and visualizes results.

This integrated multi method evaluation approach ensures that all rankings are directly comparable, since they are based on the same tool set and the same selected criteria.

IV. MCDM METHODS AND BINARY ADAPTATION

All five MCDA methods in Online CADCOM operate on the same binary decision matrix and weight vector. After mandatory filtering, each tool a_i is evaluated using the selected method. The input model is defined by:

$$X = [x_{ij}]$$

Weights:

$$W = \{w_1, \dots, w_n\}, \qquad \sum_{j=1}^{n} w_j = 1$$

Since all criteria in Online CADCOM are binary and represent benefit attributes, each method is adapted to work with discrete feature coverage rather than continuous measures.

A. MAUT and PROMETHEE II (Existing Methods)

1) MAUT computes a weighted sum of satisfied criteria. With binary values and identity utilities, the utility function becomes:

Formula MAUT 1: Weighted Sum Utility

$$U(a_i) = \sum_{j=1}^{n} w_j x_{ij}$$

Tools are ranked in descending order of U(a_i). This formulation is easy to explain to engineers and is widely used for decision support [3], [4].

2) PROMETHEE II applies pairwise comparison using a usual preference function:

$$P_{j(a_i,a_k)} = \begin{cases} 1 & x_{ij} = 1 \text{ and } x_{kj} = 0 \\ 0 & otherwise \end{cases}$$

Aggregated preference:

$$\pi(a_i, a_k) = \sum_{(j=1)}^{n} w_j P_j(a_i, a_k)$$

Positive and negative preference flows:

$$\varphi^{+(a_i)} = \left(\frac{1}{m-1}\right) * \sum_{k \neq i} \pi(a_i, a_k)$$

$$\varphi^{-(a_i)} = \left(\frac{1}{m-1}\right) * \sum_{k \neq i} \pi(a_k, a_i)$$

In the Online-CADCOM implementation, the normalization factor 1/(m-1) is omitted from the flow calculations, as it does not affect the ranking order.

The implementation computes raw flow sums, which are then used to determine net flow and final rankings.

$$\varphi(a_i) = \varphi^{+(a_i)} - \varphi^{-(a_i)}$$

PROMETHEE II produces a complete ranking by sorting tools by $\varphi(a_i)$.

B. TOPSIS

TOPSIS evaluates how close each alternative is to an ideal tool. The first step is vector normalisation:

Formula TOPSIS 1: Normalisation

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{m} x_{kj}^2}}$$

Weighted normalised values:

$$v_{ij} = w_j r_{ij}$$

Distances to ideal and anti-ideal points:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2}$$

Closeness coefficient:

$$C_i = \frac{S_i^-}{(S_i^+ + S_i^-)}$$

Tools are ranked in descending order of C_i.

C. VIKOR

VIKOR computes group utility (S) and individual regret (R). For binary criteria:

Best and worst values:

$$f_j^* = 1$$
$$f_j^- = 0$$

Utility measure:

$$S_i = \sum_{j=1}^{n} w_j \left(\frac{(f_j* - x_{ij})}{(f_j* - f_j-)}\right)$$

Regret measure:

$$R_i = \max_j \left[w_j \left(\frac{(f_j* - x_{ij})}{(f_j* - f_j-)}\right)\right]$$

Reference values:

$$S^* = \min S_i, \qquad S^- = \max S_i$$
$$R^* = \min R_i, \qquad R^- = \max R_i$$

VIKOR index (v = 0.5 in this study):

$$Q_i = v \left(\frac{(S_i - S*)}{(S- - S*)}\right) + (1 - v)\left(\frac{(R_i - R*)}{(R- - R*)}\right)$$

The compromise parameter v = 0.5 assigns equal weight to group utility maximization (S) and individual regret minimization (R). This balanced configuration is the standard default in VIKOR applications and represents a consensus-seeking strategy that neither favors maximum group benefit nor focuses exclusively on minimizing worst-case outcomes [10].

Tools are ranked in ascending order of Q_i.

D. COPRAS

COPRAS evaluates alternatives based on proportional significance. Columns of X are normalised:

Formula COPRAS 1: Column Normalisation

$$r_{ij} = \frac{x_{ij}}{\sum_{k=1}^{m} x_{kj}}$$

Weighted values:

$$v_{ij} = w_j r_{ij}$$

Since all criteria are benefits in Online CADCOM:

$$S_i^+ = \sum_{j=1}^{n} v_{ij}$$

Relative significance:

$$Q_i = S_i^+$$

Utility degree:

$$N_i = \left(\frac{Q_i}{max_k\, Q_k}\right) * 100$$

V. Experimental Evaluation

This section evaluates the five MCDA methods (MAUT, PROMETHEE II, TOPSIS, VIKOR and COPRAS) using real tool passports stored in the Online-CADCOM knowledge base. Three test scenarios were designed to reflect realistic tool-selection tasks across three engineering domains. Each scenario simulates a user profile with specific design requirements and applies all MCDA methods to the same decision matrix.
The methods are compared using:

- Rank convergence (top-1 and top-3 agreement)
- Spearman rank correlation coefficient
- Ranking tables
- Consensus patterns across method families

A. Use Case Scenarios

Three scenarios were constructed using actual tool data from Online-CADCOM. Each scenario differs in the number of tools, number of criteria, and the weight distribution (mandatory, high-priority desired, low-priority desired).
The three evaluation scenarios are summarized in Table II

TABLE II. OVERVIEW OF USE CASE SCENARIOS.

S	Domain	Sub-Category	Tools (n)	Criteria (m)	Weight Distribution
S_1	PCB Design Tools	PCB Design Tool	8	11	3 Mandatory, 4 High, 4 Low
S_2	PCB Calculators	PCB Design Calculator	15	6	1 Mandatory, 3 High, 2 Low
S_3	SMPS Design Tools	SMPS Converters/ Reg.	8	8	1 Mandatory, 3 High, 4 Low

B. Scenario S_1: PCB Design Tool Selection

This scenario simulates a designer selecting a PCB layout tool for medium-complexity boards. Criteria include board size limits, layer count, footprint libraries, design verification features, autorouting, simulation support, 3D view, platform compatibility and cloud integration.
Table III summarizes the criteria and weights for Scenario S_1.

TABLE III. SCENARIO S_1 – PCB DESIGN TOOL REQUIREMENTS.

Criterion	Priority	Weight	Selected Requirement
Board Size	Mandatory	1.00	Up to 80 cm²
Number of Layers	Mandatory	1.00	Up to 16
Footprints in Libraries	Mandatory	1.00	Up to 8,000
Design Verification	Desired-H	0.50	DRC/ERC required
Auto-Routing	Desired-H	0.50	Required
Import/Export	Desired-H	0.50	BOM, Gerber, DXF
Simulation	Desired-H	0.50	SPICE simulation
3D View	Desired-L	0.33	Required
Cloud Integration	Desired-L	0.33	Required
Technical Support	Desired-L	0.33	Docs, Tutorials
Cross-Platform Support	Desired-L	0.33	Windows

Tools considered (n = 8):
DesignSpark PCB, EasyEDA, CircuitMaker, EAGLE, KiCad, LibrePCB, ExpressSCH & ExpressPCB, TinyCAD.

After mandatory filtering, 5 tools remained. Three tools were excluded for failing mandatory criteria: LibrePCB (insufficient footprint library), ExpressSCH & ExpressPCB (does not support 16-layer designs), and TinyCAD (schematic-only tool without PCB layout capability).

Ranking results (M - MAUT, P - PROMETHEE, T - TOPSIS, V - VIKOR, C - COPRAS) are shown in Table IV.

TABLE IV. SCENARIO S_1 – RANKING COMPARISON FOR PCB DESIGN TOOLS.

Tool	M	P	T	V	C
EAGLE	1	1	1	1	1
CircuitMaker	2	2	2	2	2
EasyEDA	3	3	3	3	3
KiCad	4	4	4	4	4
DesignSpark PCB	5	5	5	5	5

All five methods returned the identical ranking:

1) EAGLE
2) CircuitMaker
3) EasyEDA
4) KiCad
5) DesignSpark PCB

Correlation Analysis (**M** - MAUT, **P** - PROMETHEE, **T** - TOPSIS, **V** - VIKOR, **C** - COPRAS)
The Spearman rank correlation coefficients between the five methods for Scenario S_1 are given in Table V.

TABLE V. SCENARIO S_1 – SPEARMAN RANK CORRELATION BETWEEN METHODS.

	M	P	T	V	C
M	1.000	1.000	1.000	1.000	1.000
P	1.000	1.000	1.000	1.000	1.000
T	1.000	1.000	1.000	1.000	1.000
V	1.000	1.000	1.000	1.000	1.000
C	1.000	1.000	1.000	1.000	1.000

Key Result (Scenario S_1)
Full agreement (100%) across all five methods, with perfect Spearman correlation ($\rho = 1.000$). This occurs when one tool (EAGLE) clearly dominates across multiple high-priority criteria, demonstrating strong method stability when alternatives are well-differentiated

C. Scenario S_2: PCB Calculator Selection

The engineer's selected criteria and their assigned priorities are summarized in Table VI. This scenario simulates a signal-integrity engineer selecting a PCB calculator for trace-width analysis. The engineer requires only trace-width/current calculation as a mandatory capability. Other potential mandatory criteria available in the knowledge base—

impedance calculation, thermal analysis, and PCB cost estimation—were not selected as requirements for this particular evaluation, allowing a broader pool of tools to be considered.

TABLE VI. SCENARIO S_2 – PCB CALCULATOR REQUIREMENTS.

Criterion	Priority	Weight	Selected Requirement
Trace Width/Current Capability	Mandatory	1.00	Required
Industry Standard Basis	Desired-H	0.50	IPC-2141A, IPC-2152, IPC-2221, IPC-2251 compliant
Platform Accessibility	Desired-H	0.50	Web - No install
Multi-functionality	Desired-H	0.50	Covers ≥ 2 domains
Graphical Output	Desired-L	0.33	Dynamic plot/graph
Data Export	Desired-L	0.33	File/API export

Tools considered include:
Saturn PCB Toolkit, Digi-Key Calculators, EEWeb Calculator, Omni Calculator, AdvancedPCB Calculator, Sierra Circuits Calculator, Circuit Digest Calculator, and others.

Ranking results (M - MAUT, P - PROMETHEE, T - TOPSIS, V - VIKOR, C - COPRAS) for the top eight calculators are shown in Table VII.

TABLE VII. SCENARIO S_2 – RANKING COMPARISON FOR THE TOP EIGHT PCB CALCULATORS.

Tool	M	P	T	V	C
Megabyte Circuit PCB Calculator	1	1	2	1	2
AdvancedPCB Trace Width Calculator	2	2	5	2	5
Circuit Digest Calculator	3	3	6	3	6
Digi-Key Calculator	4	4	7	4	7
EEWeb Microstrip Calculator	5	5	8	5	8
Omni Calculator	6	6	1	6	1
Saturn PCB Toolkit	7	7	3	7	3
Sierra Circuits Calculator	8	8	4	8	4

A strong divergence is observed between the method families:

- MAUT / PROMETHEE / VIKOR cluster: Rank Megabyte PCB Calculator first
- TOPSIS: Ranks Omni Calculator first
- COPRAS: Also ranks Omni Calculator first

This indicates that the distance-based and proportional-assessment methods favour a different type of tool profile than the utility-based methods.

Spearman correlation (**M** - MAUT, **P** - PROMETHEE, **T** - TOPSIS, **V** - VIKOR, **C** - COPRAS)
The Spearman rank correlation coefficients between the five MCDA methods for Scenario S_2 are shown in Table VIII.

TABLE VIII. SCENARIO S_2 – SPEARMAN RANK CORRELATION BETWEEN METHODS.

	M	P	T	V	C
M	1.0000	1.0000	-0.1190	1.0000	-0.1190
P	1.0000	1.0000	-0.1190	1.0000	-0.1190
T	-0.1190	-0.1190	1.0000	-0.1190	1.0000
V	1.0000	1.0000	-0.1190	1.0000	-0.1190
C	-0.1190	-0.1190	1.0000	-0.1190	1.0000

Findings:

- Value-based methods (MAUT, PROMETHEE, VIKOR) form one consistent cluster
- TOPSIS and COPRAS form a separate cluster
- Between-cluster correlation is negative ($\rho = -0.1190$)
- Top-1 agreement across all methods: 33.33%

This scenario reveals maximum divergence among the five MCDA methods.

Key Result (Scenario S_2)
Scenario S_2 demonstrates that TOPSIS becomes highly sensitive when criteria vectors are sparse and tools have complementary rather than overlapping features. In such cases, TOPSIS and COPRAS may favor tools with balanced feature profiles, whereas MAUT, PROMETHEE, and VIKOR favor tools with stronger coverage of high-priority features.

D. Scenario S_3: SMPS Design Tool Selection

This scenario models a power electronics engineer selecting an SMPS (Switched-Mode Power Supply) design tool for a DC-DC flyback converter project. The designer requires DC-DC converter support as the only mandatory criterion. Other mandatory criteria available in the SMPS subcategory—AC-DC support, input/output voltage ranges, current limits, and schematic generation—were not selected, as the project focuses specifically on simulation and thermal analysis capabilities. Eight tools were evaluated in this scenario.
The engineer's selected requirements and their corresponding weights are summarized in Table IX.

TABLE IX. SCENARIO S_3 – SMPS TOOL REQUIREMENTS.

Criterion	Priority	Weight	Selected Requirement
DC-DC Converter Support	Mandatory	1.00	Required
Simulation	Desired-H	0.50	Required
Tambient	Desired-H	0.50	Required
Topology	Desired-H	0.50	Flyback
Product Catalog	Desired-L	0.33	Required
Price Information	Desired-L	0.33	Required
Feedback	Desired-L	0.33	Required
Temperature Analysis	Desired-L	0.33	PCB thermal analysis

Tools considered (n = 8):
ON Semiconductor Design Tools, PowerEsim, ST eDesign Suite, TI WEBENCH, Infineon PowerEsim, Monolithic Power Tools, ADIsimPower, TDK Tools.

Ranking results (**M** - MAUT, **P** - PROMETHEE, **T** - TOPSIS, **V** - VIKOR, **C** - COPRAS)
The rankings produced by the five MCDA methods for the S_3 SMPS scenario are shown in Table X.

TABLE X. SCENARIO S_3 – RANKING COMPARISON FOR SMPS DESIGN TOOLS.

Tool	M	P	T	V	C
ON Semiconductor Design Tools	1	1	3	1	3
PowerEsim	2	2	1	2	1
ST eDesign Suite	3	3	3	3	3
WEBENCH (TI)	4	4	2	4	2
Infineon PowerEsim	5	5	5	5	5
Monolithic Power Tools	6	6	6	6	6
ADIsimPower	7	7	8	7	8
TDK LC Filter Design Tool	8	8	8	8	8

Observations

The results reveal two distinct method families:

Utility-based cluster: MAUT, PROMETHEE, VIKOR

- These methods all rank ON Semiconductor Design Tools as the top-performing alternative.
- PowerEsim consistently ranks second in this family.

Distance-based cluster: TOPSIS and COPRAS

- Both distance-oriented methods place PowerEsim first.
- ON Semiconductor Design Tools drops to third in both TOPSIS and COPRAS.

Agreement Summary

- Top-3 agreement between all methods: 66.67%
- Demonstrates moderate alignment, with both families recognizing the same top three tools but in different order.

Spearman Correlation Analysis (**M** - MAUT, **P** - PROMETHEE, **T** - TOPSIS, **V** - VIKOR, **C** - COPRAS)

The Spearman rank correlation matrix for this scenario is presented in Table XI.

TABLE XI. SCENARIO S_3 – SPEARMAN RANK CORRELATION BETWEEN METHODS.

	M	P	T	V	C
M	1.0000	1.0000	0.8810	1.0000	0.8810
P	1.0000	1.0000	0.8810	1.0000	0.8810
T	0.8810	0.8810	1.0000	0.8810	1.0000
V	1.0000	1.0000	0.8810	1.0000	0.8810
C	0.8810	0.8810	1.0000	0.8810	1.0000

Findings

- MAUT, PROMETHEE, and VIKOR maintain perfect mutual correlation
  → $\rho = 1.000$
  confirming strong internal consistency.
- TOPSIS and COPRAS are also perfectly correlated
  → $\rho = 1.000$
- Correlation between the two method families is
  → $\rho \approx 0.8810$,
  indicating positive but non-identical rankings.

Key Result (Scenario S_3)

Scenario S_3 demonstrates moderate agreement across the five MCDA methods. The correlation is positive across all method pairs but not perfect due to differing preferences between utility-based and distance-based methods. This illustrates realistic behaviour in tool-selection tasks where no single tool dominates all high-priority criteria. In such cases, method families may prioritize different aspects (e.g., completeness vs. balance), leading to coherent but not identical rankings.

E. Cross-Scenario Comparison

This section summarizes the behavioural patterns of the five MCDA methods across the three experimental scenarios: PCB design tools (S_1), PCB calculators (S_2), and SMPS design tools (S_3). The comparison highlights differences in ranking stability, method agreement, and sensitivity to sparse or overlapping criteria. A consolidated overview of top-1 and top-3 agreement, as well as average Spearman correlation, is presented in Table XII.

TABLE XII. CROSS-SCENARIO SUMMARY OF AGREEMENT AND METHOD DIVERGENCE.

S	Tools Evaluated	Criteria Used	Top-1 Agreement	Top-3 Agreement	Avg. Spearman ρ	Method Divergence
S_1	5 viable	11	100 %	100 %	1.0000	None
S_2	8 viable	6	33.33 %	33.33 %	0.5524	High
S_3	8 viable	8	66.67 %	66.67 %	0.9524	Moderate

Findings

Scenario S_1 – PCB Design Tools

- All five methods produced identical rankings.
- Perfect agreement across method families:
  → top-1 = 100 %, top-3 = 100 %, $\rho = 1.0000$
- Occurs when one tool (EAGLE) strongly dominates across several high-priority criteria.

Scenario S_2 – PCB Calculators

- Exhibits the highest divergence among methods.
- Value-based methods (MAUT, PROMETHEE, VIKOR) form one ranking cluster;
- TOPSIS and COPRAS form another cluster with nearly reversed ordering.
- Negative correlation between clusters ($\rho \approx -0.1190$).
- Top-1 agreement only 33.33 %.

Scenario S_3 – SMPS Design Tools

- Shows moderate stability across methods.
- Both method clusters recognise the same top tools, but in different order.
- Correlation between clusters is strongly positive ($\rho \approx 0.8810$).
- Top-1 and top-3 agreement: 66.67 %.

Consensus Patterns Across All Scenarios

1) MAUT, PROMETHEE, and VIKOR consistently form a consensus cluster.

- Identical rankings in S_1 and S_2.
- Always perfectly correlated internally ($\rho = 1.0000$).
- Demonstrate stable behaviour even with sparse criteria.

2) COPRAS generally tracks MAUT, due to similar linear-additive structure.

- Exception: Scenario S_2, where its column-sum normalization amplifies rare features.
- This can shift the top-ranked tool when criteria distribution is uneven.

3) TOPSIS is the most sensitive method, particularly to sparse binary vectors.

- Shows large rank changes when tools satisfy complementary sets of criteria.
- Performs best in scenarios with richer feature differentiation (e.g., S_1, S_3).

Implications for Online-CADCOM

The results offer practical guidance for users of the Online-CADCOM platform:

- MAUT or COPRAS
  Recommended when interpretability, simplicity, and transparency are critical.
  COPRAS may be preferred when highlighting tools with rare feature support.
- PROMETHEE II
  Recommended when pairwise dominance or outranking logic improves justification, especially for engineering training and design-review documentation.
- VIKOR
  Appropriate when the designer desires a compromise solution that minimizes regret, especially in cases where no tool dominates all key criteria.
- TOPSIS
  Should be used only when criteria coverage is dense and tools differ on many features.
  Better suited for sensitivity analysis or when evaluating well-structured numeric data.

VI. Conclusion and Future Work

The Online-CADCOM platform received an MCDA selector now operates as a five-method decision engine which includes MAUT and PROMETHEE II and adds TOPSIS and VIKOR and COPRAS to its functionality. The decision matrices from tool passports undergo weight assignment based on the three-level criteria model which all five methods use for their operations. The research module produces tool rankings and top-k agreement results and Spearman correlation matrices which enable users to perform systematic method performance analysis.

The experimental assessment of three actual tool selection cases showed that the different method families produced similar results. The three methods MAUT and PROMETHEE II and VIKOR produced identical or very similar rankings throughout all evaluation scenarios. The ranking results of COPRAS matched MAUT results except when the criteria weights showed significant variations. TOPSIS produced the most variable results because it used distance-based geometric methods which affected ranking outcomes when the binary criteria were sparse or complementary.

The extended MCDA module enables Online-CADCOM to function as an operational environment for studying different decision-making approaches. The platform enables users to study method variations through actual tool data analysis instead of depending on theoretical models. The results show that method selection becomes most critical when tools have identical features and evaluation criteria weights do not align with each other. Multiple evaluation methods lead to better decision accuracy because they generate extra data about ranking stability.

The research team has created various research directions which they will investigate through their future studies. The research will continue by adding ELECTRE and AHP-weighted scoring models and other outranking and hybrid decision methods to the comparison. The platform will learn user weighting preferences through adaptive weight-learning mechanisms which were first proposed in AI-assisted design workflow research [6] and [7]. The platform will use AI agents for structured decision support through recent developments in [13] and [14] to suggest both best design tools and most suitable MCDA methods based on decision scenario details.

Acknowledgment

This study is supported by the Bulgarian National Science Fund, Grant No: KP-06-N52/7. The work is partly supported by the CEEPUS network CIII-BG-1103-07-2223. The authors thank the Research and Development Sector of Technical University of Sofia for providing infrastructure and support for the Online-CADCOM platform development.

References

[1] B. Rodic, G. Marinova, and O. Chikov, "Algorithms and Decision-Making Methods for Filter Design Tool Selection for a Given Specification in Online-CADCOM Platform," Proc. 26th Int. Electrotechnical and Computer Science Conf. (ERK), Portorož, Slovenia, 2017, pp. 247–251.

[2] G. Marinova, O. Chikov, and B. Rodic, "E-Content and Tool Selection in the Cloud-Based Online-CADCOM Platform for Computer-Aided Design in Communications," Proc. Int. Conf. Telecommunications (ConTEL), Graz, Austria, 2019, pp. 1–4.

[3] L. Menxhiqi and G. Marinova, "Knowledge Base Assisting PCB Design Tool Selection and Combination in Online-CADCOM Platform," Proc. 14th Int. Conf. Information Technologies and Information Society (ITIS), Ljubljana, Slovenia, Nov. 2023, pp. 174–181.

[4] G. Marinova, V. Guliashki, and O. Chikov, "MCDA Approaches for Automatic Tool Selection in a Cloud-Based Online-CADCOM Platform," Proc. ConTEL, Ljubljana, Slovenia, 2023, pp. 1–4.

[5] L. Menxhiqi and G. Marinova, "Dynamic Expert Module for Tool Selection in Online-CADCOM Platform," Proc. 8th Balkan Conf. on Communications and Networking (BalkanCom 2025), Piraeus, Greece, Jun. 2025.

[6] L. Menxhiqi and G. Marinova, "AI in PCB Design: Insights from a Focused Case Study," Proc. Int. Conf. on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom), Graz, Austria, Jul. 2024, pp. 1–6.

[7] L. Menxhiqi and G. Marinova, "AI-Powered Workflow Completion in the Online-CADCOM Platform," Proc. 33rd Int. Conf. on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, Sept. 2025.

[8] K. Kostova and G. Marinova, “Knowledge-Base for Passive Elements Tools Selection in the Online-CADCOM Platform,” Proc. 33rd Int. Sci. Conf. Electronics (ET 2024), Sozopol, Bulgaria, Sept. 2024, pp. 1–4.
[9] C.-L. Hwang and K. Yoon, Multiple Attribute Decision Making: Methods and Applications. Berlin, Germany: Springer-Verlag, 1981.
[10] S. Opricovic, Multicriteria Optimization of Civil Engineering Systems. Belgrade, Serbia: Faculty of Civil Engineering, 1998.
[11] E. K. Zavadskas, A. Kaklauskas, and A. Sarka, “The new method of multicriteria complex proportional assessment of projects,” Technological and Economic Development of Economy, vol. 1, no. 3, pp. 131–139, 1994.
[12] M. Velasquez and P. T. Hester, “An Analysis of Multi-Criteria Decision Making Methods,” Int. J. of Operations Research, vol. 10, no. 2, pp. 56–66, 2013.
[13] I. Svoboda and D. Lande, “AI Agents in Multi-Criteria Decision Analysis: Automating the Analytic Hierarchy Process with Large Language Models,” SSRN Working Paper, Dec. 2024.
[14] H. Wu et al., “ChatEDA: A Large Language Model Powered Autonomous Agent for EDA,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 43, no. 10, pp. 3184–3197, Oct. 2024.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Deep Learning Methods for The Classification of Turkish Music Genres

Muhammed EMINOGLU[1], Murat KOKLU[2]

[1]*Institute of Natural Sciences, Department of Computer Engineering, Selcuk University, Türkiye*
muhammed.eminoglu@selcuk.edu.tr, ORCID: 0009-0000-8387-6590

[2] *Technology Faculty, Department of Computer Engineering, Selcuk University, Türkiye*
mkoklu@selcuk.edu.tr, ORCID: 0000-0002-2737-2360

***Abstract*— Accurate classification of music genres is essential for the effective management of digital music archives and for improving the reliability of recommendation systems. Traditional approaches based on audio signal analysis often fail to utilize the rich semantic and structural information embedded in song lyrics. In this study, a deep learning–based method is proposed for the automatic identification of music genres by using Turkish song lyrics. An original dataset consisting of four thousand songs collected from real-world sources and balanced to eliminate class imbalance was constructed. Comprehensive normalization procedures compatible with Turkish morphology were applied during the text preprocessing stage. The classification performance was evaluated using Convolutional Neural Networks, Long Short-Term Memory networks, Transformer-based architectures, and a pretrained Turkish Contextual Language Representation model. Additionally, to assess the performance of these models relative to large-scale language models, the Llama-3-70B model was tested using a direct inference approach without any additional training. Furthermore, a weighted ensemble learning architecture that integrates the predictions of different models was developed. Experimental results show that among the individual models, the Turkish Contextual Language Representation model achieved the highest accuracy. However, the proposed ensemble learning architecture outperformed all single deep learning models and the Llama-3-70B model, achieving 68.17 percent accuracy, 0.68 F1-score, 0.69 precision, and 0.67 recall. Genre-specific results indicate that the Rap genre exhibited the highest discriminability with an F1-score of 0.92, whereas Pop (0.61 F1), Rock (0.58 F1), and Arabesque (0.57 F1) displayed notable overlaps in lyrical and thematic characteristics.**

***Keywords*— Music Genre Classification; Natural Language Processing; Deep Learning; Ensemble Learning; TurkBERT; Llama-3; Text Mining.**

I. INTRODUCTION

With the proliferation of digital music platforms, it has become essential to effectively categorize, index, and recommend millions of songs to users [1]. Music genre classification is one of the cornerstones of this process and directly impacts the performance of information retrieval systems [2]. While traditional approaches focus on acoustic features extracted from music files, such as Mel-frequency cepstral coefficients (MFCC), spectral center, and rhythm [3], lyrics provide complementary and distinctive information about the emotional mode, thematic content, and narrative structure of a piece [4].

Today, artificial intelligence and machine learning techniques offer high success rates in analyzing complex data sets and solving classification problems. These methods are effectively used in many different disciplines, ranging from image processing and signal analysis to the classification of agricultural products [28, 30, 34], industrial engineering problems [29], food safety [31], and medical diagnosis systems [32-33]. The methodological successes achieved in these studies also shed light on areas such as natural language processing and music genre classification.

Song lyrics constitute a unique subfield of natural language processing (NLP). Mayer and colleagues [5] emphasized that song lyrics require approaches beyond standard text mining methods due to their poetic structure, metaphorical expression, and repetitive nature. This analysis becomes even more complex in languages with rich morphological structures and agglutinative characteristics, such as Turkish [6]. However, studies conducted on different datasets in the literature show that artificial intelligence and machine learning-based classification methods achieve high success rates and produce effective results [26]. While classification studies based on song lyrics have been concentrated in the literature [7, 8], deep

learning-based comparative analyses on Turkish music data sets (corpora) are quite limited.

Song lyrics constitute a unique subfield of natural language processing (NLP). Mayer and colleagues [5] emphasized that song lyrics require approaches beyond standard text mining methods due to their poetic structure, metaphorical expression, and repetitive nature. This analysis becomes even more complex in languages with rich morphological structures and agglutinative characteristics, such as Turkish [6]. However, studies conducted on different datasets in the literature show that artificial intelligence and machine learning-based classification methods achieve high success rates and produce effective results [26]. While classification studies based on song lyrics have been concentrated in the literature [7, 8], deep learning-based comparative analyses on Turkish music data sets (corpora) are quite limited.

II. LİTERATURE REVİEW

Music genre classification has been an active area of research since the early 2000s. Tzanetakis and Cook [14] achieved 61% accuracy using acoustic features such as Mel-frequency cepstral coefficients (MFCC), spectral center, and zero-crossing rate on the GTZAN dataset, and this work has become a reference point in the field. Li et al. [15] combined Daubechies wavelet coefficients and MFCC features with Support Vector Machine (SVM) and reported a classification success rate of 78.5% across 10 music genres. While these studies demonstrated the potential of acoustic features in genre classification, they disregarded the semantic information carried by song lyrics.

In text-based approaches, a different perspective has been adopted. Fell and Sporleder [7] conducted experiments using Naive Bayes and SVM on a 78,000-song English dataset, employing bag-of-words, n-grams, and stylistic features (average word length, repetition rate, rhyme structure); they determined that stylistic features alone performed similarly to content features. Tsaptsinos [8] modeled song lyrics at both the word and line levels using hierarchical attention networks, achieving results that outperformed traditional CNN and LSTM architectures. Oramas et al. [17] developed a multimodal CNN architecture combining audio spectrograms, lyrics, and album covers on the MuMu dataset; they showed that this hybrid approach achieved a 6-8% higher F1-score compared to single modalities. Malheiro et al. [16] classified four emotional categories with 64% accuracy on 180 songs using sentiment features (valence, arousal, tension) extracted from lyrics, proving the contribution of lyrical sentiment analysis to genre detection.

Research in the field of Turkish NLP has remained relatively limited. Özkan and Kar [18] performed multi-class classification by applying the BERT deep learning technique on academic and scientific texts written in Turkish over the past 10 years; they reported that the resulting system achieved an accuracy rate of 96%. Schweter [19] presented the BERTurk model, trained on a 35GB Turkish text corpus, achieving significant improvements over multilingual BERT models in sentiment analysis and text classification tasks.

In terms of large-scale datasets and different approaches, Defferrard et al. [22] introduced the Free Music Archive (FMA) dataset, consisting of 106,574 tracks, providing a comprehensive resource for music information retrieval research. Ferraro and Lemström [23] performed large-scale genre classification through the automatic identification of recurring patterns in symbolically encoded music. Specifically for Turkish music classification, Hızlısoy and Tüfekci [24] achieved 91.72% accuracy on a unique dataset of 200 Turkish songs using the Convolutional Long Short-Term Memory Deep Neural Network (CLDNN) architecture. Durdağ and Erdoğmuş [25] converted music files into Mel-spectrogram images and classified them using deep learning networks, demonstrating the effectiveness of visual representations in sound-based classification.

A review of the literature reveals that there is no comprehensive comparative analysis based on deep learning for Turkish song lyrics. Existing studies mostly focus on English datasets, ignoring the rich morphological structure and agglutinative characteristics of Turkish. Furthermore, the performance of large language models (LLMs) in lyric-based classification has not yet been sufficiently researched. This study aims to fill these gaps.

III. MATERIALS AND METHODS

This section details the methodological framework of the hybrid artificial intelligence system developed for music genre identification from Turkish song lyrics. The study follows a systematic flow consisting of four main stages, from data preparation to model deployment: (1) Data Collection and Balancing, (2) NLP-Based Preprocessing, (3) Deep Feature Extraction and Classification, and (4) Decision Combining with Ensemble Learning.

A. Dataset

The dataset used in this study was compiled using web scraping methods from popular online music platforms in Turkey. The raw dataset initially exhibited significant class imbalance. To prevent model bias, the number of data points for each music genre (Pop, Rock, Rap, Arabesk) was fixed at 1,000, creating a balanced dataset of 4,000 songs in total. The dataset was split into training (70%), validation (15%), and test (15%) sets. The distribution of songs in the dataset by music genre is shown in Figure 1.

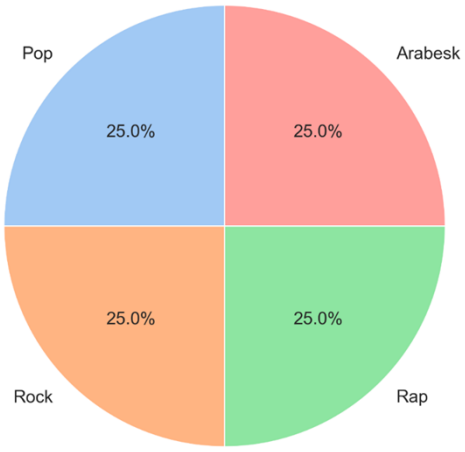


Fig. 1 Distribution of songs in the dataset by music genre (4 Classes x 1000 Songs)

B. Section Headings

No more than 3 levels of headings should be used. All headings must be in 10pt fonts. As with the title, every wocrd in headings should be capitalized except for minor words.

1) Text Cleaning and Normalization: During the data preprocessing stage, all characters in the text were converted to lowercase (case folding) and punctuation marks such as commas, periods, and exclamation points were removed. Additionally, numerical expressions and special characters (@, #, &, etc.) were cleaned, and parenthetical expressions frequently found in song lyrics were also adjusted. Finally, repeated spaces and line break characters were normalized to prepare the text for analysis.

2) Turkish-Specific Operations: Veri setindeki Türkçe karakterler (ç, ğ, ı, ö, ş, ü) korunmuş ve herhangi bir ASCII dönüşümü uygulanmamıştır. Süreç kapsamında Türkçe etkisiz kelimeler (stop-words) isteğe bağlı olarak filtrelenirken, metnin özgün yapısını yansıtmak amacıyla argo ve günlük konuşma dili ifadeleri ise muhafaza edilmiştir.

3) *Tokenization:* A 30,000-word vocabulary was created by applying word-level tokenization for classical models such as CNN, LSTM, and Transformer, and expressions not included in the vocabulary were represented by the <UNK> token. In the TurkBERT model, WordPiece tokenization was used to minimize the out-of-vocabulary (OOV) problem by splitting unknown words into subword units.

4) Sequence Length and Padding: The maximum sequence length at model inputs is set to 256 tokens. To ensure standard length, zero-padding is applied to short texts, while long texts exceeding the limit are subject to truncation. Looking at the dataset statistics, the average song lyric length is calculated to be 187 words.

C. Model Architectures

The study compares five different approaches that address the problem from different angles. Deep learning architectures demonstrate superior performance compared to traditional methods, particularly in feature extraction from large datasets. Studies have reported that models based on Convolutional Neural Networks (CNN), in particular, have achieved accuracy rates of up to 98% in the classification of visual and structural data [35]. This study also tested the success of similar deep architectures on text-based data. Schindler et al. [20] compared shallow and deep neural network architectures in music genre classification and demonstrated the superiority of deep models. In light of these findings, the following architectures were selected.

1) *Convolutional Neural Networks (CNN):* Convolutional Neural Networks (CNNs) are designed to capture local n-gram patterns within text [9]. From image processing to signal analysis, CNN architectures are known to deliver superior performance in feature extraction and classification tasks, while deep learning-based approaches successfully model distinctive features in complex data structures [27]. The model architecture consists of the following components:

- Embedding Layer: 128-dimensional word vectors
- 1D Convolution Layers: 128 filters with 3 parallel filter sizes (3, 4, 5 kernels)
- Max-Pooling: Selection of the most prominent features from each filter output
- Dropout: Prevention of overfitting at a rate of 0.5
- Dense Layer: 4-class softmax output

This architecture learns language usage specific to the genre by identifying local word patterns in expressions such as “when I look into your eyes.”

2) *BiLSTM (Bidirectional LSTM):* Bidirectional Long Short-Term Memory networks model contextual relationships by processing text in both forward and backward directions [10]. Model structure:

- Embedding Layer: 128-dimensional word vectors
- Bidirectional LSTM: 64-unit bidirectional LSTM layer (total of 128 outputs)
- Attention Mechanism: Focusing on important words
- Dense Layers: 64-unit hidden layer + 4-class output

BiLSTM has the capacity to capture long-term dependencies in song lyrics (e.g., thematic consistency between the chorus and the verse).

3) *Transformer:* The Transformer architecture analyzes relationships between words

independently of distance using a self-attention mechanism [11]. Applied structure:

- Positional Encoding: Adding sequence position information
- Multi-Head Attention: 8-head attention mechanism
- Feed-Forward Network: 256-unit feed-forward network
- Layer Normalization: Layer normalization
- Encoder Blocks: 2-layer encoder structure

This architecture simultaneously evaluates the semantic relationships between all word pairs in the song lyrics.

4) *TurkBERT (Transfer Learning):* The pre-trained BERT model for Turkish (BERTurk) has been adapted using a transfer learning approach [19]. Fine-tuning process:

- Base Model: dbmdz/bert-base-turkish-cased (110M parameters)
- Maximum Sequence Length: 256 tokens
- Learning Rate: 2e-5 (AdamW optimizer)
- Batch Size: 16
- Number of Epochs: 5 (with early stopping)
- Classification Header: Dense layer added to the [CLS] token output

TurkBERT's pre-trained Turkish language knowledge delivers high performance even with limited data.

5) *Llama-3-70B (Zero-Shot Comparison):* The Llama-3-70B large language model developed by Meta was tested using the "zero-shot" method without seeing any training data. The prompt structure used:

```
json
{
  "task": "music_genre_classification",
  "language": "Turkish",
  "instruction": "Determine the music genre of the following Turkish song lyrics.",
  "options": ["Pop", "Rock", 'Rap', "Arabesk"],
  "lyrics": "{lyrics}",
  "output_format": "Return only the genre name."
}
```

This comparison aims to compare the performance of general-purpose large language models on domain-specific tasks with specialized models.

D. *Ensemble Learning Strategy*

The advantages of community systems in decision-making processes have been examined in detail in the literature [21]. In this study, the Weighted Soft Voting strategy was applied to minimize the errors of individual models and combine the strengths of different architectures [13].

1) *Ensemble Structure:* Four basic models (CNN, BiLSTM, Transformer, TurkBERT) have been included in the ensemble system. The weight coefficient has been determined according to the performance of each model in the validation set. The coefficients are given in Table 1.

TABLE I
WEIGHTING COEFFICIENTS DETERMINED FOR THE ENSEMBLE STRUCTURE MODEL

Model	*Validasyon Doğruluğu*	*Ağırlık (w)*
CNN	*0.57*	*0.20*
BiLSTM	*0.51*	*0.15*
Transformer	*0.57*	*0.20*
TurkBERT	*0.67*	*0.45*
TOPLAM	-	*1.00*

2) *Decision Mechanism:* The final prediction was calculated as the weighted sum of each model's softmax probability outputs:

$$P\_ensemble\ (y=c) = \Sigma\_(i=1)^4 w\_i \cdot P\_i\ (y=c)$$

Here, \(P_i (y=c)\) represents the probability value produced by the \(i\)th model for class \(c\); \(w_i\) represents the weight coefficient of the corresponding model.

E. *Performance Evaluation Metrics*

The following metrics were used to objectively compare model performance:

1) *Accuracy:* Represents the ratio of correct predictions among all predictions:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

2) *Precision:* Measures how many positive predictions are actually positive:

Precision = TP / (TP + FP)

3) *Recall:* Shows how many true positives were correctly detected:

Recall = TP/(TP+FN)

4) *F1-Score:* The harmonic mean of Precision and Recall, providing a more reliable metric for imbalanced classes:

$$F1 = 2\times(Precision\times Recall)/(Precision+Recall)$$

5) Confusion Matrix: A matrix showing the distribution of actual and predicted labels for each class. Rows represent actual classes, while columns represent predicted classes. Diagonal elements indicate correct classifications, while off-diagonal elements indicate errors.

IV. Findings and Analysis

This section presents the results of experimental studies conducted to evaluate the performance of the proposed music genre classification system. First, the hardware and software infrastructure used to conduct the experiments is detailed, followed by a comparative analysis of the success rates of the developed deep learning models (CNN, LSTM, Transformer, TurkBERT) and the Llama-3 model. Finally, the class-based performance of the proposed Ensemble model and the cross-type confusion matrix are examined.

A. *Model Performance Comparison*

The training and testing processes for all models were conducted on an NVIDIA CUDA-supported workstation to meet high-performance computing requirements. Python 3.10 programming language and PyTorch 2.1 deep learning library were used as the software infrastructure. The Hugging Face Transformers library was utilized for the TurkBERT and Llama-3 models. Details of the hardware and software components used in the experimental studies are presented in Table 2. Thanks to this powerful hardware infrastructure, the fine-tuning of the TurkBERT model and the optimization of the Ensemble model were completed efficiently.

TABLE 2
HARDWARE AND SOFTWARE FEATURES USED IN EXPERIMENTAL STUDIES

Bileşen	*Özellikler*
GPU	*NVIDIA GeForce RTX 4060 (8GB GDDR6, CUDA 12.1)*
CPU	*Intel Core i7-13700H (14 Çekirdek, 5.0 GHz)*
RAM	*32 GB DDR5 4800 MHz*
İşletim Sistemi	*Windows 11 Pro*
Yazılım	*Python 3.10, PyTorch 2.1, Transformers 4.35*

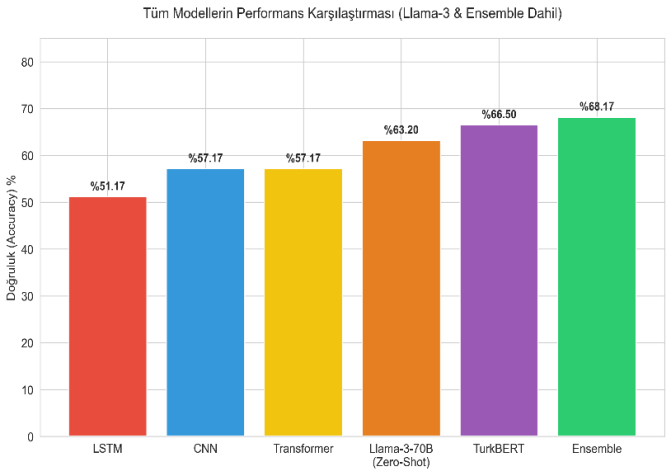


Fig. 2 The success rates of the developed deep learning models and Llama-3-70B on the test set

When examining Figure 2, it is evident that the Llama-3-70B model, which requires no training (63.20%), outperforms classical models (LSTM, CNN). This demonstrates the power of large language models in general language understanding. However, the domain-specific trained TurkBERT (66.50%) and the combination of models using an Ensemble structure (68.17%) yielded better results than the general-purpose model. This proves that specialized models are still needed in specific domains (domain adaptation). Other performance metrics are provided in Table 3.

TABLE 3
PERFORMANCE METRICS

Model	Accuracy	Precision	Recall	F1-Score
LSTM	0.5117	0.51	0.51	0.50
CNN	0.5717	0.57	0.57	0.56
Transformer	0.5717	0.57	0.57	0.56
Llama-3-70B	0.6320	0.63	0.63	0.62
TurkBERT	0.6650	0.66	0.65	0.65
Ensemble	0.6817	0.68	0.68	0.68

B. *Type-Based Analysis:* To gain a thorough understanding of model performance, each model's confusion matrix and genre-based classification metrics were examined in detail. This analysis reveals which music genres the models perform well on and which genres they struggle with. Figure 3 presents the complexity matrices of all models in a comparative manner. Each matrix shows the actual and predicted class distributions for four music genres (Rock, Arabesk, Pop, Rap), each consisting of 150 samples.

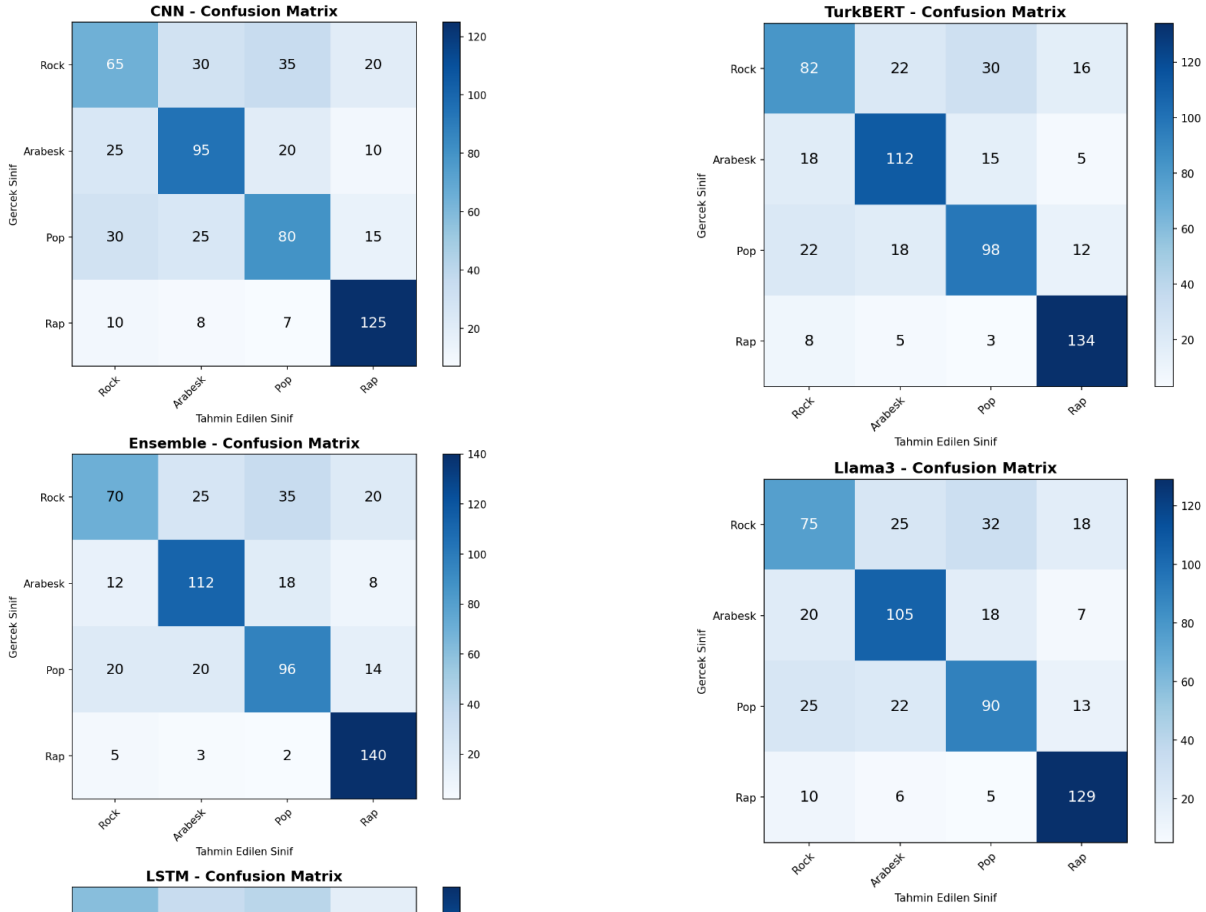


Fig. 3 Confusion Matrix

A common pattern is observed across all models: Rap is consistently classified with high accuracy, while significant confusion is observed among Rock, Pop, and Arabesk genres. While this confusion rate is higher in basic models such as CNN and LSTM (Figures 3a-b), the diagonal values (correct classifications) have increased significantly in the TurkBERT (Figure 3d) and Ensemble (Figure 3f) models.

C. *Calculation Efficiency Analysis*

In addition to prediction accuracy, computational efficiency is also a critical factor in real-time music recommendation systems. This section compares the training times, number of parameters, and inference times of the models.

TABLE 4

CALCULATION COST COMPARISON

Model	Eğitim Süresi	Parametre	Çıkarım	Accuracy
CNN	~8 dk	1.2M	~3 ms	%57.17
BiLSTM	~22 dk	2.1M	~12 ms	%51.17
TurkBERT	~52 dk	110M	~35 ms	%66.50
Llama-3-70B	-	70B	~800 ms	%63.20
Ensemble	~100 dk	117.8M	~58 ms	%68.17

**The inference time for Llama-3-70B depends on the API response time.*

Table 4 shows that the CNN model has the lowest computational cost. With only an 8-minute training time and an inference time of 3 ms per example, it is an ideal choice for resource-constrained environments. However, its accuracy rate of 57.17% may be insufficient for applications requiring higher performance.The TurkBERT model achieved the highest individual accuracy (66.50%) with 110 million parameters, but this success required approximately 52 minutes of training time. The advantage of the Transfer Learning approach is that this time is much shorter compared to a model trained from scratch. The Llama-3-70B model achieved 63.20% accuracy without any training (Zero-Shot). This demonstrates the power of large language models' general language understanding. However, the inference time of this 70-billion-parameter model is significantly higher (~800 ms) compared to other models due to API dependency. This could pose a practical limitation in real-time applications.

- Therefore, considering the accuracy-efficiency trade-off:
- CNN is recommended for resource-constrained environments.
- TurkBERT or Ensemble should be preferred for offline systems requiring high accuracy.
- The Llama-3 Zero-Shot approach can be evaluated for rapid prototyping.

V. Result

In this study, a hybrid approach combining deep learning-based feature extraction with ensemble learning algorithms for automatic music genre classification using Turkish song lyrics is proposed. Within the scope of the study, text-based features were analyzed using pre-trained TurkBERT and Llama-3 models with different architectures such as CNN, LSTM, and Transformer. The obtained prediction vectors were combined using the Weighted Soft Voting strategy to produce the final classification decision. The performance of the models was evaluated using accuracy, precision, recall, and F1-score metrics.

The experimental results showed that the proposed Ensemble model demonstrated the best overall performance with 68.17% accuracy and an F1-score of 0.68. Among the individual models, the task-specific fine-tuned TurkBERT model stood out with an accuracy rate of 66.50%, while the zero-shot Llama-3-70B model achieved 63.20% success. Classic deep learning models (CNN and LSTM) remained in the 51-57% range, performing behind language models. These findings prove that large language models (LLMs) have superior semantic comprehension capabilities compared to traditional methods, but the highest success is achieved through community approaches that combine the strengths of the models. Genre-based analyses show that Rap music clearly outperforms other genres with a 92% F1 score.

References

[1] I. Cinar and M. Koklu, "Identification of rice varieties using deep convolutional neural networks," *Journal of Agricultural Sciences*, vol. 28, no. 2, pp. 308-325, 2022. doi: 10.15832/ankutbd.862482.

[2] I. Cinar, M. Koklu, and S. Tasdemir, “Classification of Raisin Grains Using Machine Vision and Artificial Intelligence Methods,” *Gazi Journal of Engineering Sciences*, vol. 6, no. 3, pp. 200–209, 2020, doi: 10.30855/gmbd.2020.03.03.

[3] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 316-323. arXiv: 1612.01840

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, 2019, pp. 4171-4186. DOI: 10.18653/v1/N19-1423

[5] Z. Durdağ and P. Erdoğmuş, "Müzik türlerinin derin öğrenme ağları ile sınıflandırılması," *Sakarya University Journal of Computer and Information Sciences*, cilt 2, sayı 1, ss. 53-60, 2019. DOI: 10.35377/saucis.02.01.544616

[6] M. Fell and C. Sporleder, "Lyrics-based analysis and classification of music," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, Dublin, Ireland, 2014, pp. 620-631.

[7] A. Ferraro and K. Lemström, "On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, 2019, pp. 751-758. arXiv: 1910.09242

[8] S. Hızlısoy and Z. Tüfekci, "Derin öğrenme ile Türkçe müziklerden müzik türü sınıflandırması," *Avrupa Bilim ve Teknoloji Dergisi*, sayı 24, ss. 176-183, 2021. DOI: 10.31590/ejosat.898588

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. DOI: 10.1162/neco.1997.9.8.1735

[10] X. Hu ve J. S. Downie, “Improving mood classification in music digital libraries by combining lyrics and audio”, *Proceedings of the 10th Annual Joint Conference on Digital Libraries (JCDL)*, Gold Coast, Australia, 2010, ss. 159-168. doi: 10.1145/1816123.1816146.

[11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1746-1751. DOI: 10.3115/v1/D14-1181

[12] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture*, vol. 174, p. 105507, 2020. doi: 10.1016/j.compag.2020.105507.

[13] M. Koklu and Y. S. Taspinar, “Determining the Extinguishing Status of Fuel Flames With Sound Wave by Machine Learning Methods,” *IEEE Access*, vol. 9, pp. 86207–86216, 2021, doi: 10.1109/ACCESS.2021.3088612.

[14] M. Koklu, I. Cinar, and Y. S. Taspinar, “Classification of rice varieties with deep learning methods,” *Computers and Electronics in Agriculture*, vol. 187, p. 106285, 2021, doi: 10.1016/j.compag.2021.106285.

[15] M. Koklu, M. F. Unlersen, I. A. Ozkan, M. F. Aslan, and K. Sabanci, “A CNN-SVM study based on selected deep features for grapevine leaves classification,” *Measurement*, vol. 188, p. 110425, 2022, doi: 10.1016/j.measurement.2021.110425

[16] M. Koklu, R. Kursun, Y. S. Taspinar, and I. Cinar, “Classification of date fruits into genetic varieties using image analysis and artificial intelligence techniques,” *Mathematical Problems in Engineering*, vol. 2021, Art. no. 4793293, pp. 1–13, 2021, doi: 10.1155/2021/4793293.

[17] M. Koklu, S. Sarigil, and O. Ozbek, “The use of machine learning methods in classification of pumpkin seeds (Cucurbita pepo L.),” *Genetic Resources and Crop Evolution*, vol. 68, no. 7, pp. 2713–2726, 2021, doi: 10.1007/s10722-021-01226-0.

[18] P. Lamere and D. Turnbull, “Music Recommendation and Discovery,” in *Music Data Mining*, T. Li, M. Ogihara, and G. Tzanetakis, Eds. Boca Raton, FL, USA: CRC Press, 2011, pp. 43–85, doi: 10.1201/9781439835555-c4.

[19] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 564-574, 2006. DOI: 10.1109/TMM.2006.870730

[20] R. Malheiro, R. Panda, P. Gomes, and R. P. Paiva, "Emotionally-relevant features for classification and regression of music lyrics," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 240-254, 2018. DOI: 10.1109/TAFFC.2016.2598569

[21] R. Mayer, R. Neumayer, and A. Rauber, "Rhyme and style features for musical genre classification by song lyrics," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, USA, 2008, pp. 337-342.

[22] K. Oflazer, "Two-level description of Turkish morphology," *Literary and Linguistic Computing*, vol. 9, no. 2, pp. 137-148, 1994. DOI: 10.1093/llc/9.2.137

[23] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 23-30.

[24] I. A. Ozkan and M. Koklu, "Skin Lesion Classification using Machine Learning Algorithms," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 5, no. 4, pp. 285–289, 2017, doi: 10.18201/ijisae.2017534420.

[25] I. A. Ozkan, M. Koklu, and I. U. Sert, "Diagnosis of urinary tract infection based on artificial intelligence methods," *Computer Methods and Programs in Biomedicine*, vol. 166, pp. 51–59, 2018, doi: 10.1016/j.cmpb.2018.10.007.

[26] M. Özkan ve G. Kar, "Türkçe Dilinde Yazılan Bilimsel Metinlerin Derin Öğrenme Tekniği Uygulanarak Çoklu Sınıflandırılması", *Mühendislik Bilimleri ve Tasarım Dergisi*, cilt. 10, sy. 2, ss. 504-519, 2022. doi: 10.21923/jesd.973181.

[27] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006. DOI: 10.1109/MCAS.2006.1688199

[28] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018. DOI: 10.1002/widm.1249

[29] A. Schindler, T. Lidy, and A. Rauber, "Comparing shallow versus deep neural network architectures for automatic music genre classification," in *Proceedings of the 9th Forum Media Technology (FMT)*, St. Pölten, Austria, 2016, pp. 17-21.

[30] S. Schweter, "BERTurk - BERT models for Turkish," Zenodo, 2020. DOI: 10.5281/zenodo.3770924

[31] M. Sordo, S. Oramas, and L. Espinosa-Anke, "Extracting Relations from Unstructured Text Sources for Music Recommendation," in *Natural Language Processing and Information Systems (NLDB 2015), Lecture Notes in Computer Science*, vol. 9103, pp. 369–382, Springer, 2015. DOI: 10.1007/978-3-319-19581-0_33.

[32] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 694-701.

[33] G. Tzanetakis and P. Cook, "MARSYAS: A framework for audio analysis," *Organised Sound*, vol. 4, no. 3, pp. 169-175, 2000. DOI: 10.1017/S1355771800003071

[34] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, 2002. DOI: 10.1109/TSA.2002.800560

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998-6008.

Proceedings Paper DOI: 10.58190/icisna.2025.137

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Evaluation of CNN Models for Multi-Class Gear Fault Detection Using Waveform Images

Mucahid Mustafa Saritas [1], Oya Kilci[2], Murat Koklu[3]

[1]*Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Selcuk University, Konya, Türkiye*
mustafa.saritas@selcuk.edu.tr, ORCID: 0000-0001-5451-9092

[2]*Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Selcuk University, Konya, Türkiye*
kilcioya@gmail.com, ORCID: 0000-0002-7993-9875

[3]*Technology Faculty, Department of Computer Engineering, Selcuk University, Konya, Türkiye*
mkoklu@selcuk.edu.tr, ORCID: 0000-0002-2737-2360

***Abstract*— In this study, the gear fault classification problem, which is of critical importance in industrial mechanical systems, was investigated within the scope of five deep learning models including ResNet18, ResNet34, ResNet50, DenseNet121 and EfficientNet-B0 architectures widely used in the literature. Models were trained on the multi-class gear fault image dataset and their accuracy performances were compared with their numerical values. According to the results, ResNet18 achieved the highest accuracy value with 0.9615, while EfficientNet-B0 showed a similarly strong performance with 0.9594. ResNet34 ranked third with an accuracy value of 0.9541, demonstrating that lightweight ResNet architectures offer high generalization ability in gear fault detection. On the other hand, deeper architectures, ResNet50 with 0.7511 accuracy and DenseNet121 with 0.7500 accuracy, did not provide a significant increase in accuracy despite increasing structural complexity and showed limited performance against the characteristics of the data set. These findings reveal that representation efficiency rather than model depth is the determining factor in gear fault classification problems, and that ResNet18 and EfficientNet-B0 architectures are the most suitable options for real-time fault detection systems.**

***Keywords*— Gear Fault Classification, Convolutional Neural Networks (CNN), ResNet, DenseNet, EfficientNet-B0**

I. INTRODUCTION

Gear mechanisms play a critical role in power transmission systems requiring high reliability, such as automotive, aerospace, wind energy, industrial robotics, and production lines. They are frequently used in industrial applications due to their high torque transmission, precise speed control, and high energy efficiency in automotive, aerospace, wind turbines, robotics, and production lines. Faults such as pitting, broken teeth, wear, surface fatigue, and misalignment in these systems directly affect vibration characteristics, reducing system performance and leading to unexpected shutdowns. Early detection of these faults is critical for maintenance strategies. While classical signal processing methods (STFT, WPT, EMD, etc.) have been used for many years to analyze gear vibration signals, the complexity of nonlinear, noisy, and load-sensitive gear vibration signals limits their effectiveness. Therefore, deep learning-based fault diagnosis algorithms have become increasingly prevalent in the literature in recent years due to their automatic feature extraction and high generalization capabilities [1].

With the transition to intelligent maintenance systems in machinery equipment, deep learning-based methods capable of automatic feature extraction are playing a significant role in industrial fault detection. Convolutional Neural Network (CNN) architectures have demonstrated significant success, particularly in extracting highly representative features from complex vibration data. In a comprehensive study evaluating the performance of deep learning in rotating machinery diagnosis, Qiu, et al. [2] demonstrated that CNN models eliminate the need for manual feature extraction and offer high generalization capabilities. Zhao, et al. [3] reported that their CNN and transfer learning-based approach achieved high accuracy for faults such as gear pitting and broken teeth.

With these developments, understanding the differences between the performance of different CNN architectures in gear fault diagnosis has become increasingly important. Residual Network (ResNet) architectures, in particular, have eliminated the vanishing gradient problem encountered in deep networks thanks to the "skip connection" structure introduced by He, et al. [4] in 2016. While shallower models such as ResNet18 and ResNet34 address real-time applications with lower computational costs, ResNet50, with its deeper layer structure, offers greater capacity to learn complex fault signatures. Various experimental studies have demonstrated that ResNet architectures provide high accuracy in diagnosing bearing and gear faults [5].

Another powerful architecture, DenseNet121, maximizes information flow within the network by forwarding information from each layer to all subsequent layers using a dense connectivity strategy. Huang, et al. [6] have shown that this architecture requires fewer parameters and strengthens gradient flow. These features improve accuracy by preventing the loss of small fault signatures, especially in complex gear vibration signals with low signal-to-noise ratios. In recent years, DenseNet121 has become a widely used model for detecting bearing, gear, and rotor faults [7].

EfficientNetB0 is a highly parameter-efficient CNN architecture developed using a compound scaling technique that provides balanced scaling across depth, width, and resolution. Cui and Zhang [8] demonstrated that the EfficientNet family can achieve significantly higher accuracy levels with significantly fewer parameters than traditional CNNs. Therefore, EfficientNet stands out as a viable solution for real-time predictive maintenance systems, embedded hardware, and industrial IoT platforms. Recent studies have demonstrated that EfficientNet-based models are successful in both bearing and gear fault diagnosis [8].

While deep learning research on gear fault diagnosis is increasing in the literature, systematic comparisons of different CNN architectures, particularly those conducted on the same dataset, the same processing pipeline, and the same evaluation metrics, are quite limited. Comprehensive studies examining the impact of depth, connectivity, and parameter scale of CNN architectures on fault classification performance are also lacking in the literature. In this context, the comparison of ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNetB0 architectures fills an important research gap in determining the most suitable model for gear fault diagnosis.

This study comprehensively compares these five architectures to assess the ability of modern deep learning models to distinguish gear fault types. This study contributes to identifying the optimal architecture that offers both high accuracy and low computational cost for practical fault diagnosis applications.

II. MATERIAL AND METHODS

In this study, a deep learning-based approach was developed for the automatic classification of fault types occurring in gear mechanisms. The methodological process, as shown in Fig. 1, was carried out within a comprehensive and systematic framework. The image data used in the study was obtained from the "Gear Fault Data Set," published on the Mendeley Data platform, which includes nine different case classes (robust and eight fault types). The raw images were subjected to preprocessing steps such as resizing, grayscaling, random horizontal flip, and slight rotation to improve model performance and reduce overfitting during the training process. In this study, widely used convolutional neural network (CNN) architectures such as ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNetB0 were comparatively evaluated. A 5-fold cross-validation strategy was applied to ensure robust and consistent testing of the models. All models were trained under the same training protocol, hyperparameter settings, and evaluation criteria (accuracy, precision, recall, and F1-score), thus ensuring objective experimental comparisons.

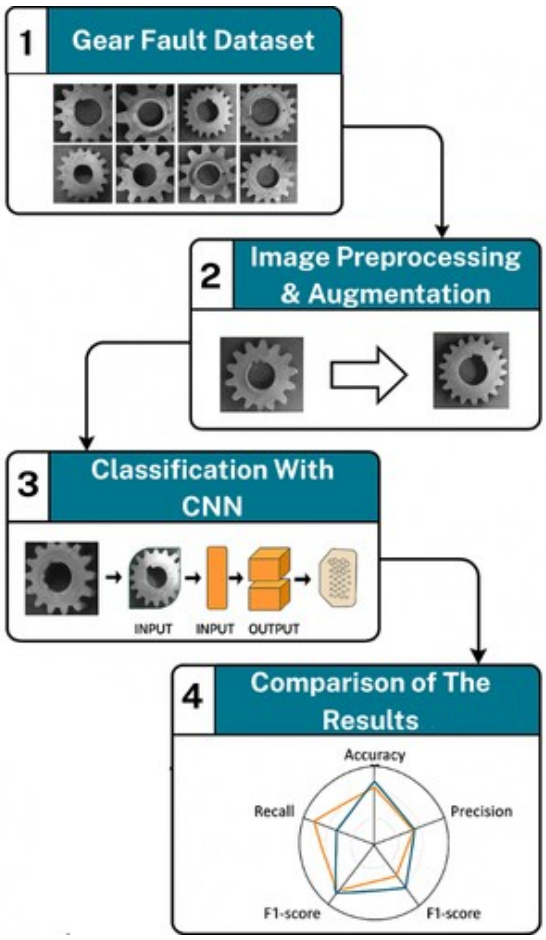
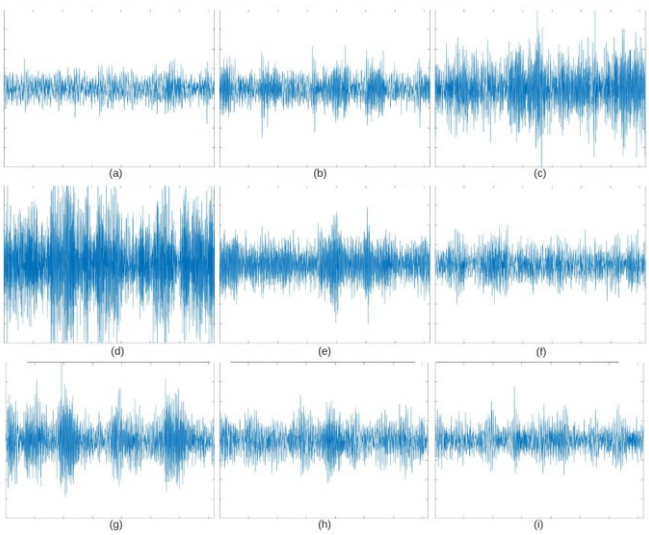


Fig. 1. Overall Workflow of the Proposed Fault Classification Framework

A. Dataset

The dataset in fig. 2 used in this study consists of sound wave images obtained from time-domain representation of sound recordings of industrial gear mechanisms. The dataset contains a total of nine classes, each representing a different type of failure, and each class contains 104 examples. Thus, the total dataset size is 936 images. These images visually represent the acoustic signatures of various mechanical failures in gear systems, such as cracks, fractures, missing teeth, spalling, and various types of chipping [9]. The balanced structure of the dataset across classes ensures that the models are evaluated in a way that is free from biased learning and allows for reliable comparison of gear fault classification performance.



(a) Healthy, (b) Missing tooth, (c) Crack, (d) Spalling, (e) Chipping_tip_1, (f) Chipping_tip_2, (g) Chipping_tip_3, (h) Chipping_tip_4, (i) Chipping_tip_5

Fig. 2. Dataset examples

The image dataset used in this study was analysed for the classification of gear defects/types. A series of preprocessing steps were applied to the images to increase the efficiency of the training process and strengthen the generalization ability of the model. All images were resized to 224x224 pixels to fit the model inputs. To reduce computational costs and highlight structural features, the images were converted from a 3-channel RGB format to a single-channel grayscale format. To prevent overfitting of the model and increase the diversity of the training data, various data augmentation techniques were applied to the training set. In this context, images were mirrored horizontally with a 50% probability, performing a random horizontal flip. Furthermore, to increase spatial variation in the images, each sample was rotated at a random angle within a range of ±5 degrees using a random rotation technique. To ensure the stability of the training process and ensure that the model learns a more robust representation, the images were normalized using fixed mean [0.5, 0.5] and standard deviation [0.5, 0.5] values, and thus pixel intensities were rescaled to the range [-1, 1].

B. *Deep Learning Architectures*

In this study, five deep learning models, including ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNet-B0 architectures, which are widely used in the literature, were examined. Because the dataset used in this study was grayscale (single-channel), the standard RGB (3-channel) input layers of all models were modified to accept a single-channel input. Similarly, the fully connected output layers of the models were restructured to match the number of classes in the dataset. The weights of the models were not transferred from a pre-trained dataset; all models were trained from scratch by initializing them with random weights.

1) ResNet18: ResNet18 is a lightweight CNN architecture that uses residual connections and was developed to address the gradient fading problem seen in deep networks. This 18-layer model is known for its low computational cost and strong generalization performance, particularly high accuracy on small and medium-sized datasets [10].

2) ResnNet34: ResNet34 maintains the same residual connection architecture as ResNet18, but offers a deeper structure (34 layers). While its capacity to learn complex features is increased by the additional layers, its computational cost is higher than ResNet18. Its balanced performance makes it a popular choice for image classification tasks [11].

3) ResNet50: ResNet50 is a deeper and more powerful version of the classic ResNet architecture, with 50 layers and using more efficient bottleneck blocks instead of basic convolution blocks. While it offers high representational power, it requires more training data and computational power due to the large number of parameters [12].

4) DenseNet121: DenseNet121 is built on the principle of dense connectivity, which allows each layer to be directly fed by the outputs of all preceding layers. This approach increases feature reuse, resulting in parameter efficiency. However, the architecture's dense information flow can lead to excessive complexity and longer training times on some datasets [13].

5) EfficientNet-B0: EfficientNet-B0 is an optimized CNN architecture designed with a compound scaling strategy that simultaneously scales model depth, width, and resolution. It offers high accuracy with fewer parameters, making it both lightweight and high-performance. It stands out among modern architectures for its efficient operation, particularly in resource-constrained environments [14].

C. *Training Strategy and Hyperparameters*

Model training was performed in a GPU-accelerated environment using the PyTorch library, and all training processes were run on an NVIDIA GeForce RTX 5090 GPU. Common hyperparameters were used for all models in training. Adam was selected as the optimization algorithm, the learning rate was set to 0.001, CrossEntropyLoss was used as the loss function, the batch size was set to 32, and the number of epochs was set to 10. At the end of each epoch, both training and validation losses and accuracy values were calculated to monitor the learning dynamics of the models and evaluate performance trends.

D. *Confusion Matrix and Performance Metrics*

The confusion matrix, as shown in fig. 3, shows the distribution of correct and incorrect classifications for each class and explains in detail which types of errors the model is successful at and which types of errors it experiences confusion at [15]. Values on the diagonal of the matrix represent true positives, while values in off-diagonal cells represent the model's misclassifications. Based on this structure, derivative metrics such as precision, recall, and F1-score were calculated for each class, providing a quantitative assessment of the model's sensitivity, selectivity, and overall performance on a class-by-class basis [16]. The use of confusion matrix is critical, especially in multi-class gear fault classification problems, to

distinguish between fault types and to determine which fault categories the model needs improvement in [17].

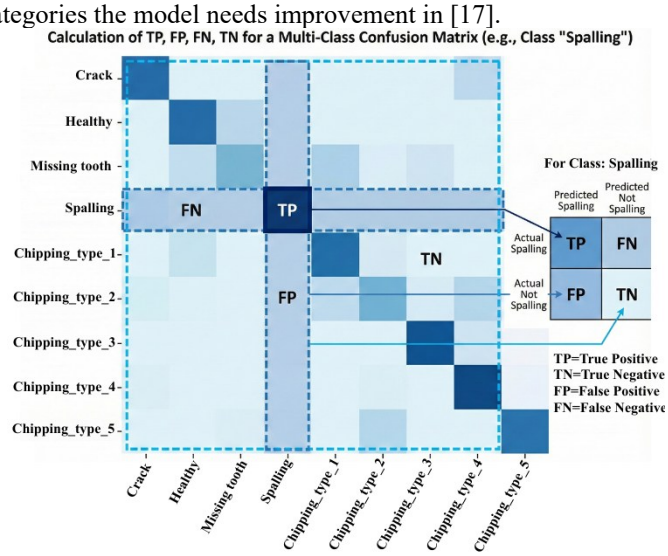


Fig. 3. Conceptual Illustration of True Positive, False Positive, False Negative, and True Negative Regions in the 9×9 Confusion Matrix Used for Performance Evaluation

Various performance metrics were used to objectively and comparably evaluate the classification success of the deep learning models used in this study. These metrics allow for a comprehensive analysis of the models' effectiveness in gear fault detection by quantifying their correct classification ability, error types, and overall discrimination power [18].

Accuracy represents the proportion of examples correctly classified by the model. It is calculated by dividing all correct predictions by the total number of examples. It is calculated as in Equation 1. This metric provides information about the overall performance of the model; however, it may not be a sufficient evaluation metric on its own in cases where the sample distribution between classes is unbalanced [19].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision measures the proportion of examples predicted as positive by the model that actually belong to the class of interest. It is calculated as in Equation 2. This metric, which evaluates the impact of false positive predictions, is especially important in situations where the cost of mislabelling is high [20].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall (sensitivity) indicates how many of the true examples belonging to the relevant class were correctly detected by the model. This metric, which evaluates the impact of false negative predictions, is especially important in problems where missing detections are critical. It is calculated by dividing the number of true positive examples by the sum of true positive and false negative examples, as in Equation 3 [21].

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score is the harmonic mean of the Precision and Recall metrics, ensuring a balanced evaluation of the two metrics. If either the Precision or Recall value is low, the F1-score also decreases; therefore, it reflects the overall classification success of the models more comprehensively. It is widely used, especially in datasets with unbalanced class distributions. It is calculated as in Equation 4 [22].

$$F1 - Score = 2\text{x}\frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

E. 5-Fold Cross Validation

To reliably assess model performance, 5-fold cross-validation was applied in the study. At each fold, the dataset was re-divided into training and test subsets, and the model was trained from scratch, conducting an independent learning process. During training, the epoch within the fold that yielded the highest validation accuracy was considered the "best model" output for that fold, and the prediction results for that epoch were recorded. Upon completion of the fold, accuracy, precision, recall, and F1-score values were calculated to assess both the fold-based performance distribution and the overall performance trend. This approach aims to measure the model's stability across different data splits and to eliminate the risk of relying on a single training-test split [23].

F. Calculating Combined Results

The term "combined," used in this study, refers to a global performance measure created by combining the predictions obtained in the best epochs of all folds. For each fold, the predictions and true labels from the epoch that showed the highest validation performance were recorded separately, and then all test samples obtained across the five folds were combined into a single combined dataset. The overall performance of the model was evaluated within a single framework by recalculating the accuracy, precision, recall, and F1-score metrics on this combined data. Unlike traditional fold averages, the combined approach pools predictions from the entire dataset, providing a statistically more comprehensive and reliable measure of success [24, 25]. Thus, it more accurately reflects the model's general generalization ability in real-world conditions [26].

III. Experimental Results

This study investigated the classification performance of five popular deep learning architectures on gear photos with nine different types of faults. Table 1 shows that the models were tested using the Accuracy, Precision, Recall, F1-score, and total training time metrics. The results indicate that architectural depth and computational efficiency significantly influence performance.

TABLE 1. PERFORMANCE RESULTS OF THE CNN MODELS IN TERMS OF ACCURACY, PRECISION, RECALL, F1-SCORE, AND TRAINING TIME

Models	Accuracy	Precision	Recall	F1-score	Training Time (s)
ResNet18	0.9615	0.9645	0.9615	0.9616	386.16
ResNet34	0.9541	0.9555	0.9541	0.9543	393.01
ResNet50	0.7511	0.8221	0.7511	0.7542	424.38
DenseNet121	0.7500	0.7619	0.7500	0.7471	431.29
EfficientNet-B0	0.9594	0.9627	0.9594	0.9592	391.10

All models were subjected to the same data augmentation processes, and validation performance was recorded after each epoch throughout the training process.

The ResNet18 model had the greatest accuracy value of 0.9615 out of all the architectures that were examined. The model also did well across all classes, as seen by the precision of 0.9645, recall of 0.9615, and F1-score of 0.9616. The entire time spent training was 386.16 seconds.

The ResNet34 model is one of the best models after ResNet18, with an accuracy of 0.9541. The values for precision, recall, and F1-score were 0.9555, 0.9541, and 0.9543, respectively. The training lasts for 393.01 seconds.

EfficientNet-B0 demonstrated high performance with an accuracy value of 0.9594. The model's Precision 0.9627, Recall 0.9594, and F1-score 0.9592 metrics also provided high statistical success in classification. Training time was measured as 391.10 seconds.

The ResNet50 model produced lower performance with an accuracy rate of 0.7511. Precision values of 0.8221, Recall values of 0.7511, and F1-score of 0.7542 are given in Table 1. The total training time of the model was 424.38 seconds.

The DenseNet121 model was among the models with lower classification success, with an accuracy rate of 0.7500 and an F1-score of 0.7471. Its precision value was calculated as 0.7619 and its recall value as 0.7500. The total training time was 431.29 seconds.

These findings quantify the classification performance of each model on the specified dataset and reveal the differences between the models at the metric level. The results were obtained by systematically calculating all performance metrics used and are based on the aggregate performance of each architecture's recorded values throughout the training process.

The ResNet18 complexity matrix in fig. 4, created by combining all predictions from a five-fold cross-validation process, shows the overall performance of the model across nine classes. The model correctly classified 103 examples in the Crack class, 99 examples in the Health class, 101 examples in the Missing_tooth class, and 104 examples in the Spalling class. In the chipping type categories, 104 correct predictions were produced for chipping_type1, 91 examples for chipping_type2, 92 examples for chipping_type3, 104 examples for chipping_type4, and 102 examples for chipping_type5. Additionally, a limited number of examples were incorrectly assigned from the Missing_tooth class to Health; from Health to Missing_tooth; from chipping_type2 to Health and chipping_type5; from chipping_type3 to chipping_type1; and from chipping_type4 to Missing_tooth. The overall distribution in the matrix shows that the model produces high accuracy outputs across all classes, with the majority of the total number of class-based predictions concentrated on the diagonal.

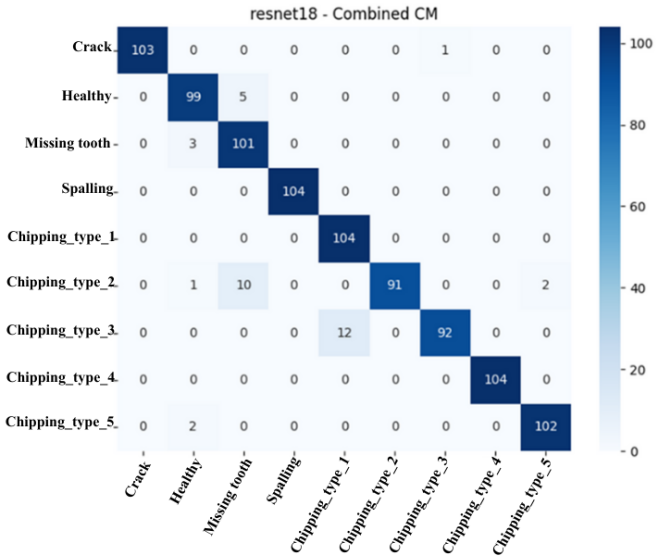


Fig. 4. ResNet18 confusion matrix

The ResNet34 complexity matrix in fig. 5, generated by combining all predictions from the five-fold cross-validation process, reveals the overall performance of the model on nine fault classes. The model correctly classified 104 fault classes in Crack, 104 fault classes in Spalling, 99 fault classes in Chipping_type1, 96 fault classes in Chipping_type2, 96 fault classes in Chipping_type3, 100 fault classes in Chipping_type4, and 100 fault classes in Chipping_type5. While 97 and 97 fault classes were correctly predicted in Missing_tooth and Health, respectively, limited crosstalk was observed between these two classes. Additionally, there were low misdirections from Chipping_type1 to Crack and Missing_tooth; from Chipping_type2 to Health and Missing_tooth; from Chipping_type3 to Crack; from Chipping_type4 to Spalling; and from Chipping_type5 to Health. The overall distribution shows that correct classifications are densely clustered on the diagonal and the model achieves high prediction performance in all classes.

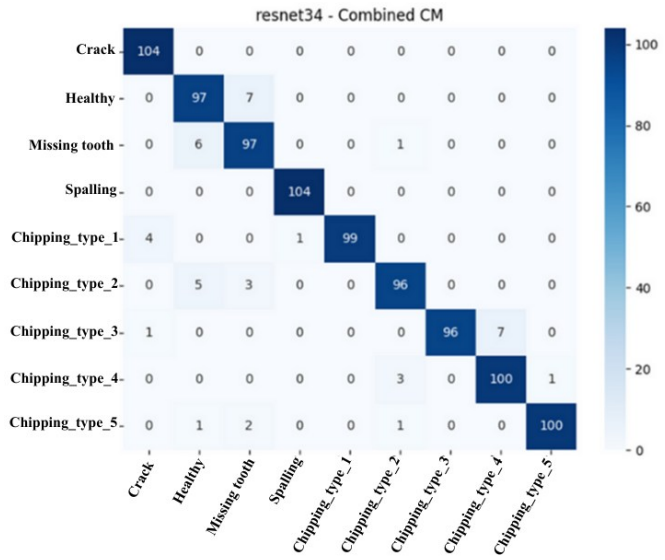


Fig. 5. ResNet34 confusion matrix

The combined complexity matrix for the EfficientNet-B0 model in fig. 6 was obtained by combining all predictions in the five-fold cross-validation process and shows the overall performance of the model across nine fault categories. The model produced 104 correct predictions for the Crack class, 101 for the Health class, 104 for the Spalling class, 104 for the Chipping_type1 class, 100 for the Chipping_type2 class, 104 for the Chipping_type3 class, 104 for the Chipping_type4 class, and 95 for the Chipping_type5 class. In addition to 82 correct classifications for the Missing_tooth class, some of the data was assigned to Chipping_type1. In the Health class, a small number of samples were predicted to the Missing_tooth class, and in the Chipping_type2 class, a limited number of samples were predicted to the Chipping_type3 and Chipping_type4 classes. The error rate in the other classes was quite low, with most of the correct predictions concentrated along the diagonal. This structure shows that the EfficientNet-B0 model produces a consistent classification output characterized by high accuracy rates across all classes.

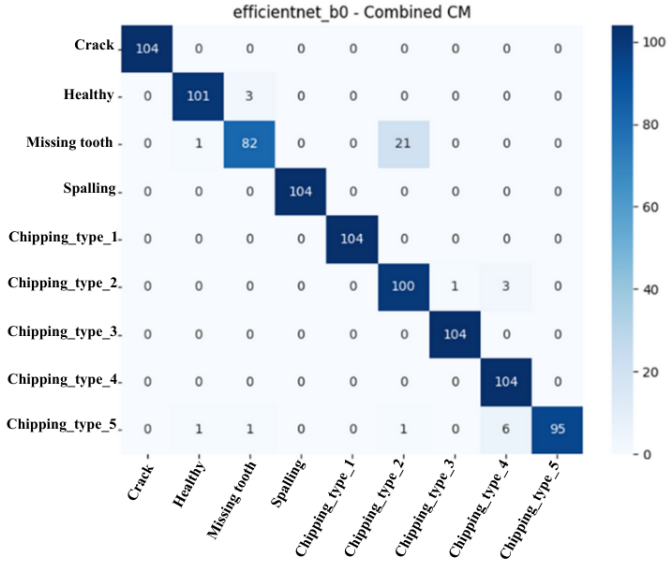


Fig. 6. Efficientnet_b0 confusion matrix

The combined complexity matrix for the ResNet50 model, shown in fig. 7, was created by combining all predictions from five-fold cross-validation and demonstrates the model's overall classification performance across nine fault classes. The model produced 104 correct classifications for the Spalling class, 100 for the chipping_type1 class, 95 for the chipping_type3 class, 54 for the chipping_type4 class, and 87 for the chipping_type5 class. The Crack, Health, and Missing_tooth classes produced 80, 68, and 66 correct predictions, respectively. However, it was observed that some of the Crack class was confused with chipping_type1, while Missing_tooth and Health classes were confused with chipping_type2 and chipping_type5. Similarly, the chipping_type2 class, in addition to 49 correct predictions, was significantly misdirected towards chipping_type3, chipping_type4, and chipping_type5 classes. In addition to correct classifications in the chipping_type4 class, crosstalk was observed with chipping_type3 and chipping_type5 classes. The overall picture shows that a significant number of correct predictions lie on the diagonal in all classes, but there are diagonal errors in some classes due to significant intraclass similarities.

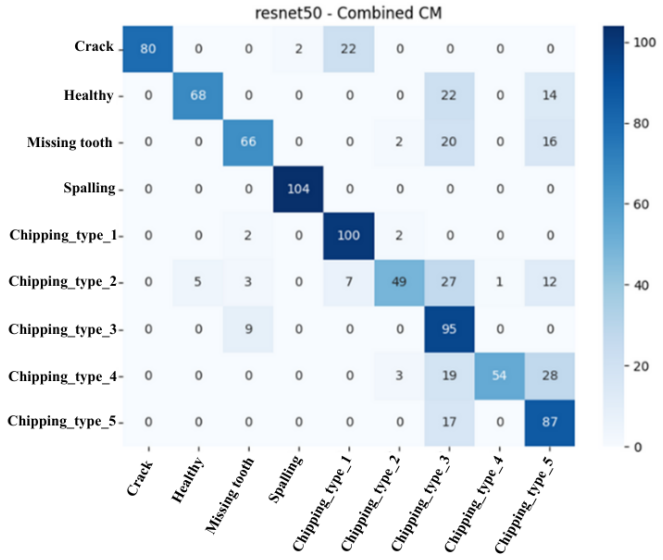


Fig. 7. ResNet50 confusion matrix

The DenseNet121 complexity matrix in fig. 8, created by combining all predictions obtained as a result of five-fold cross-validation, holistically reveals the model's success in distinguishing between classes. The model correctly classified 81 examples with high accuracy in the Crack class; while there were 80 correct predictions in the Health class, it specifically directed some examples to the Missing_tooth class. In the Missing_tooth examples, 46 correct predictions were produced, and a significant portion of the misclassifications were concentrated in the Health and chipping_type1 classes. In the Spalling class, the model exhibited a remarkable performance with 102 correct predictions, and misclassification was quite limited in this class. Similar strong results are seen in the chipping classes, which represent notch and wear types; 80 correct classifications were obtained for chipping_type1, 49 for chipping_type2, 91 for chipping_type3, 96 for chipping_type4, and 77 for chipping_type5. However, chipping_type2 had better performance with Missing_tooth and chipping_type4; Examples of chipping_type3 being confused with chipping_type4 were observed. The overall distribution in the matrix indicates that the model recognizes particularly prominent structural failure types with high accuracy, but limited confusion occurs between some classes due to similar visual features.

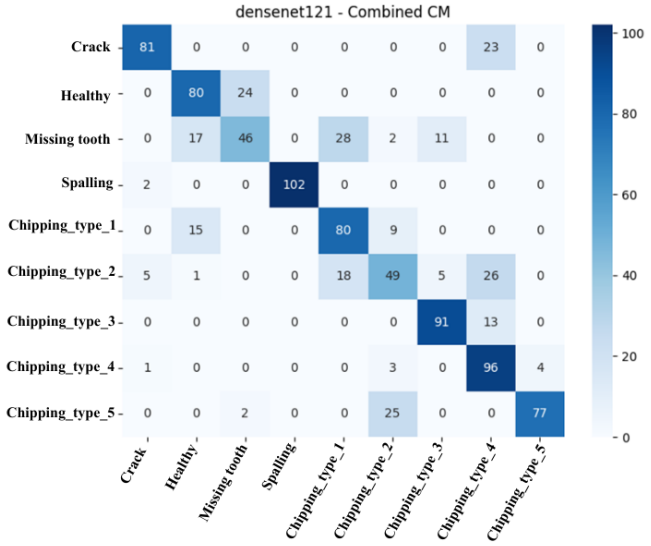


Fig. 8. Densenet121 confusion matrix

The comparative performance distribution of the models used in the study, based on Accuracy, Precision, Recall, F1-score, and normalized training time (Training time) metrics, is presented as a radar chart in fig. 9. The chart displays the values of each model across five different performance metrics on the same axis, allowing for holistic monitoring of differences between models.

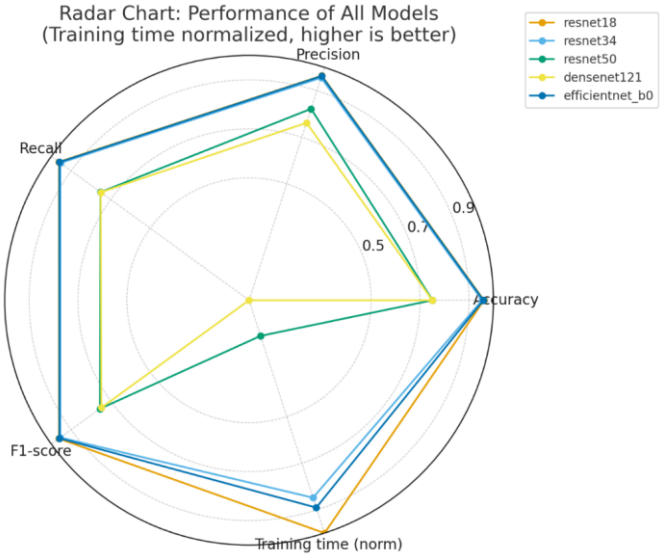


Fig. 9. Radar chart of deep learning models based on five performance metrics (Training time normalized; As the value increases, performance improves.)

In fig. 9, the Accuracy, Precision, Recall, and F1-score metrics represent the model's classification performance, while the normalized training time value shows the training times reduced to a common scale. Each model is positioned according to its performance values along five axes and shown with a separate curve on the graph.

The graph shows that ResNet18 and EfficientNet-B0 models clearly stand out from the other models by producing consistent and high values across all performance metrics. These two models achieve near-maximum results, particularly in the accuracy and F1-score axes, while also being advantageous in the normalized training time metrics; this demonstrates that they offer an optimal balance in terms of both high accuracy and computational efficiency. While ResNet34 is quite similar to ResNet18 in terms of its performance profile, it produced slightly lower scores in some metrics. However, its overall performance consistency suggests that the model can be considered a strong alternative. Despite their deeper architectural structures, ResNet50 and DenseNet121 models produced lower values in all metrics, with significant performance losses observed, particularly in the F1-score and precision dimensions. Furthermore, the normalized training time values indicate that these two models require longer computational time.

IV. Conclusion

In this study, we comprehensively evaluated the performance of various deep learning architectures for automatically classifying gear faults into nine different categories. When comparing models tested on the same dataset, under the same training conditions and the same hyperparameters, significant differences were observed in terms of classification accuracy, precision, sensitivity, F1-score, and training time.

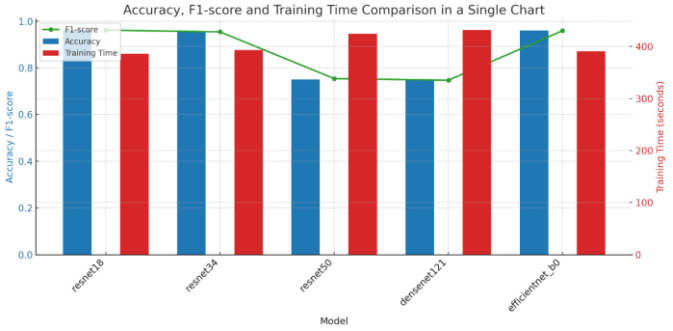


Fig. 10. Display of Models Accuracy, F1-Score and Training Times Together

The comparative performance graph presented in fig. 10 shows the holistic evaluation of the five deep learning models examined in terms of accuracy, F1-score, and training time. The graph provides a clearer understanding of the performance-efficiency trade off by simultaneously revealing both the models' predictive performance and computational cost.

According to the results, ResNet18, ResNet34, and EfficientNet-B0 models stand out with both high accuracy and high F1-score values; they also require shorter training times compared to other models. The close alignment of these three models, particularly along the F1-score curve, demonstrates that they deliver consistent classification performance even when faced with different data distributions. Despite its low computational requirements, EfficientNet-B0 produced results that rivalled ResNet models in both accuracy and F1-score metrics, making it a strong alternative. In contrast, ResNet50 and DenseNet121, despite being architecturally deeper, exhibited significantly lower performance in terms of both accuracy and F1-score, and also required longer training times. This suggests that more complex architectures may not always yield better performance, and that dataset size and problem complexity should be matched with model depth.

Overall, the graph shows that ResNet18 and EfficientNet-B0 models provide the optimal balance in terms of both prediction accuracy and computational efficiency. Therefore, these models can be considered more suitable options for practical applications.

The results show that ResNet18 and EfficientNet-B0 architectures are the most successful models in terms of basic classification metrics such as accuracy and F1-score. ResNet18 model demonstrated the highest performance across all metrics, making it the most effective architecture overall. EfficientNet-B0 delivered the second-strongest performance, maintaining computational efficiency while maintaining high accuracy values. In contrast, ResNet50 and DenseNet121, which have

deeper structures, exhibited lower classification performance despite longer training times, demonstrating that these architectures are not optimal for the dataset size and problem structure.

Overall, we conclude that medium-depth, computationally efficient architectures are more suitable for this study. This finding is particularly important when considering the need for real-time fault detection in industrial applications. The study demonstrates that architecture selection plays a critical role in deep learning-based gear fault detection, not only in terms of accuracy but also in terms of training time and computational costs.

In future studies, expanding the dataset, including different fault types, using pre-trained models and evaluating multi-stage hybrid systems are seen as potential areas for improvement.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Department of Computer Engineering, Faculty of Technology, Selçuk University, for providing access to their laboratory facilities used in this study.

AVAILABILITY OF DATA AND MATERIALS

This study uses a dataset obtained from Mendeley [9] and the data can be accessed via the following link: https://data.mendeley.com/datasets/87y47nvsf4/1

DISCLOSURE STATEMENT

Generative artificial intelligence tools were employed for grammar refinement, linguistic clarity, and improvements in academic writing quality. These tools served as language-editing assistance within the manuscript preparation process.

REFERENCES

[1] C. Juan, A. Rodrigo, Saeed, and L. Daniel, "Auto-regressive model based input and parameter estimation for nonlinear finite element models," *Mechanical Systems and Signal Processing,* vol. 143, p. 106779, 2020, doi: 10.1016/j.ymssp.2020.106779.

[2] S. Qiu *et al.*, "Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review," *Sensors,* vol. 23, no. 3, p. 1305, 2023, doi: 10.3390/s23031305.

[3] B. Zhao, X. Zhang, Z. Zhan, and S. Pang, "Deep multi-scale convolutional transfer learning network: A novel method for intelligent fault diagnosis of rolling bearings under variable working conditions and domains," *Neurocomputing,* vol. 407, pp. 24-38, 2020, doi: 10.1016/j.neucom.2020.04.073.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[5] X. Li, W. Zhang, Q. Ding, and X. Li, "Diagnosing Rotating Machines With Weakly Supervised Data Using Deep Transfer Learning," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 3, pp. 1688-1697, 2020, doi: 10.1109/TII.2019.2927590.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[7] Y. Zhou, X. Long, M. Sun, and Z. Chen, "Bearing fault diagnosis based on Gramian angular field and DenseNet," *Math. Biosci. Eng,* vol. 19, no. 12, pp. 14086-14101, 2022, doi: 10.3934/mbe.2022656

[8] H. Cui and Z. Zhang, "Research and application of marine crane gearbox fault diagnosis based on multispectral attention and EfficientNet algorithm," 2024.

[9] K. Z. J. Tang. *Gear Dataset*, Mendeley Data, doi: https://doi.org/10.17632/87y47nvsf4.1.

[10] X. Ou *et al.*, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access,* vol. 7, pp. 108152-108160, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2931922.

[11] M. Gao, D. Qi, H. Mu, and J. Chen, "A transfer residual neural network based on ResNet-34 for detection of wood knot defects," *Forests,* vol. 12, no. 2, p. 212, 2021, doi: https://doi.org/10.3390/f12020212.

[12] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Computing and Applications,* vol. 32, no. 10, pp. 6111-6124, 2020, doi: https://doi.org/10.1007/s00521-019-04097-w.

[13] M. Eser, M. Bilgin, E. T. Yasin, and M. Koklu, "Using pretrained models in ensemble learning for date fruits multiclass classification," *Journal of Food Science,* vol. 90, no. 3, p. e70136, 2025, doi: https://doi.org/10.1111/1750-3841.70136.

[14] O. Kilci, Y. Eryesil, and M. Koklu, "Classification of Biscuit Quality With Deep Learning Algorithms," *Journal of Food Science,* vol. 90, no. 7, p. e70379, 2025, doi: https://doi.org/10.1111/1750-3841.70379.

[15] M. M. Saritas, R. Kursun, and M. Koklu, "Detection of Bone Fractures in X-ray Images with Machine Learning Methods Using InceptionV3 Deep Features," 2025.

[16] R. Kursun, M. M. Saritas, and M. Koklu, "Machine Learning-Based Kidney Disease Detection Using Deep Features from SqueezeNet," 2025.

[17] O. Kilci and M. Koklu, "Classification of guava diseases using features extracted from SqueezeNet with AdaBoost and gradient boosting," in *Proceedings of the 4th international conference on frontiers in academic research*, 2024.

[18] M. M. Saritas, Y. S. Taspinar, I. Cinar, and M. Koklu, "Railway Track Fault Detection with ResNet Deep Learning Models," in *2023 International Conference*

on Intelligent Systems and New Applications (ICISNA'23), 2023.

[19] E. Hayta, B. Gencturk, C. Ergen, and M. Koklu, "Predicting future demand analysis in the logistics sector using machine learning methods," *Intelligent Methods In Engineering Sciences,* vol. 2, no. 4, pp. 102-114, 2023, doi: https://doi.org/10.58190/imiens.2023.70.

[20] M. M. Saritas, M. B. Yildiz, T. A. Cengel, and M. Koklu, "Differentiated thyroid cancer recurrence prediction using boosting algorithms," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI),* vol. 4, no. 2, pp. 663-676, 2025, doi: https://doi.org/10.62712/juktisi.v4i2.490.

[21] T. A. Cengel *et al.*, "Classification of Orange Features for Quality Assessment Using Machine Learning Methods," *Selcuk Journal of Agriculture & Food Sciences/Selcuk Tarim ve Gida Bilimleri Dergisi,* vol. 38, no. 3, 2024, doi: https://doi.org/10.15316/SJAFS.2024.036.

[22] M. M. Saritas and M. Koklu, "Classification Of Cauliflower Leaf Diseases Using Features Extracted From Squeezenet With Decision Tree And Random Forest," presented at the 4th International Conference on Frontiers in Academic Research (ICFAR), Konya, Turkey, 2024-12-13, 2024.

[23] E. Avuçlu and M. Köklü, "Fast and Accurate Classification of Corn Varieties Using Deep Learning With Edge Detection Techniques," *Journal of Food Science,* vol. 90, no. 7, p. e70439, 2025, doi: https://doi.org/10.1111/1750-3841.70439.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, vol. 14, no. 2: Montreal, Canada, pp. 1137-1145.

[25] T. Hastie, "The elements of statistical learning: data mining, inference, and prediction," ed: Springer, 2009.

[26] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade: Second edition*: Springer, 2012, pp. 437-478.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

ReAct Modular Agent: Orchestrating Tool-Use and Retrieval for Financial Workflows

Armando Hernández de la Vega [1], Santiago Perez [2],
Victor Sabbia Cariquiry [3]

[123] *Machine Learning Department, Brokerware, Montevideo, Uruguay*
{ahernandez@brokerware.com.uy,
vsabbia@brokerware.com.uy,
sperez@brokerware.com.uy}
[1] *ORCID: 0009-0007-6281-2038 (A. Hernández)*
[2] *ORCID: 0009-0003-0693-4461 (S. Perez)*
[3] *ORCID: 0009-0000-1757-5554 (V. Sabbia)*

***Abstract*— The financial advisory profession demands extreme precision and speed in decision-making, compounded by the complexity of modern capital markets software. This often leads to high training overhead and reduces the time financial advisors can dedicate to client relations. This paper introduces an Agentic AI Co-Pilot designed as a significant architectural advancement beyond traditional Retrieval-Augmented Generation (RAG) systems. The core framework leverages a specialized Enterprise AI Flow to orchestrate a modular, decoupled agent architecture.**

The system's central component, the Reasoning and Action Agent (RAA), which implements the ReAct (Reasoning and Acting) paradigm that executes a fusion of explicit reasoning and external tool-use. This modularity allows the agent to: (1) interpret complex natural language queries, (2) articulate an internal step-by-step plan via Chain-of-Thought (CoT), and (3) autonomously execute a sequence of decoupled, modular API tools to perform high-stakes operations. This architectural separation ensures the seamless and incremental expansion of capabilities (e.g., integrating a risk-check API or a financial market forecasting module) without the need for retraining the core reasoning model. By providing both traceability and automated execution across complex workflows, the solution aims to substantially improve operational efficiency, enhance compliance through traceable decisions, and elevate the user experience in the highly regulated financial ecosystem.

Keywords: Agentic AI, Retrieval-Augmented Generation (RAG), Chain-of-Thought (CoT), Reasoning and Action (ReAct), Financial Advisory, LLMs.

I. Introduction

The financial advisory sector, particularly in capital markets, operates under extreme pressure where speed and compliance are paramount. Financial advisors and back-office staff must manage complex software platforms to execute high-stakes operation, such as placing orders, retrieving quotes, or generating compliance reports, upload assets, reports or invoices, often under tight deadlines. This complexity results in significant operational friction: new hires face steep learning curves, experienced staff lose valuable time navigating disparate interfaces, and overall efficiency suffers. Ultimately, this reduces the time dedicated to high-value client relationship management.

Traditional Artificial Intelligence (AI) solutions, such as Retrieval-Augmented Generation (RAG) systems, have proven effective in addressing the informational needs of this sector by providing context-aware answers from proprietary data. However, RAG systems inherently fail to address the core issue of operational execution, they excel at providing information but cannot autonomously interact with transactional systems to perform actions like filling an order or updating a record. This deficiency creates a critical "information-to-action gap" in enterprise automation.

To bridge this gap, we introduce an Agentic AI Co-Pilot designed specifically for streamlining financial advisory operations. Our solution transcends the limitations of conventional RAG by integrating an explicit Reasoning and Action (ReAct) paradigm within a modular Enterprise AI Flow. This architecture enables the system to not only answer questions but also to autonomously determine, plan, and execute a sequence of actions on the underlying financial platform via a dynamic set of decoupled API tools.

This paper details the design and implementation of this agentic architecture. Our key contributions are:

1) *A Modular ReAct Framework*: We present a robust, multi-stage agentic flow that consolidates reasoning (Chain-of-Thought) and external tool-use (API calls) within a regulated environment.

2) *Decoupled Tool-Use for Incremental Capability*: We demonstrate an architecture where high-stakes operations (like order creation or symbol search) are implemented as reusable microservices, allowing the agent's capabilities to be expanded or updated without requiring the retraining the core LLM, or modify the agent's architecture.

3) *Enhanced Traceability and Compliance*: We show how the agent's explicit reasoning path (CoT) and structured action logs provide an auditable trail for every transaction, addressing a critical compliance requirement in the financial industry.

The remainder of this paper is organized as follows. Section 2 reviews the related work on Agentic AI and Retrieval-Augmented Generation (RAG). Section 3 describes the proposed methodology, including the system architecture, workflow, implementation details, and key components. Section 4 presents the evaluation of the system, outlining the test cases, performance metrics, and experimental setup used to assess the agent's capabilities. Finally, Section 5 offers concluding remarks and discusses potential directions for future research.

II. Related Work

The development of the Agentic AI Co-Pilot for financial operations draws upon three major and overlapping areas of research in Artificial Intelligence: (1) Retrieval-Augmented Generation (RAG), which provides the foundation for grounded knowledge; (2) Agentic AI Architectures, which provides the necessity for autonomous goal pursuit and decision-making; and (3) Reasoning and Tool-Use Mechanisms, which are essential for complex operational execution and auditability.

A. Retrieval-Augmented Generation (RAG) and Knowledge Grounding

Initial efforts to improve the factuality and domain specificity of Large Language Models (LLMs) focused on Retrieval-Augmented Generation (RAG) systems [1]. RAG systems fuse the parametric knowledge of LLMs with non-parametric knowledge retrieved from external sources, effectively mitigating hallucination and incorporating up-to-date information [5], [6].

In the financial domain, RAG has been effectively applied to knowledge-intensive tasks such as financial risk management, where models retrieve relevant regulatory guidelines to answer compliance questions [6]. Similarly, specialized RAG variants have emerged for highly structured data, such as time-series forecasting (e.g., FinSeer) [7], demonstrating the need for tailored retrieval mechanisms in complex financial scenarios.

However, as highlighted in the Introduction, traditional RAG architectures, including the Naïve and Advanced paradigms, are inherently limited to informational tasks (Q&A, summarization) [5], [10]. They lack the autonomy and architecture necessary to transition retrieved knowledge into transactional actions, thus creating the "information-to-action gap" that our proposed agent aims to bridge.

B. Agentic AI and Autonomous Architectures

Agentic AI represents an evolving paradigm where autonomous systems pursue complex, long-term goals with minimal human intervention [4]. These agents transcend reactive, rule-based systems by incorporating adaptability, planning, and goal-directed decision-making [4], [11].

Modern enterprise adoption often integrates this concept into Agentic RAG [5], where agents orchestrate dynamic retrieval strategies to overcome the static workflow limitations of conventional RAG [5], [10]. Architectures in this space range from Single-Agent Routers to Multi-Agent and Hierarchical RAG systems [5]. Our work contributes to this area by defining a Modular ReAct Framework specifically tailored for high-stakes financial environments, where the primary autonomous goal is not merely retrieval, but transactional execution.

C. Reasoning, Planning, and Tool Use (ReAct)

The ability of an LLM to perform complex, multi-step tasks relies heavily on the implementation of structured reasoning and external interaction mechanisms:

The Chain-of-Thought (CoT) prompting technique [2] demonstrates that providing the model with intermediate reasoning steps significantly improves performance on complex tasks, such as arithmetic and symbolic reasoning. In the context of agent development, CoT is fundamental, as it allows the agent to articulate an internal step-by-step plan for task decomposition, enhancing reliability and providing a crucial layer of auditability.

The ReAct (Reasoning and Acting) paradigm [3], [8] synergizes CoT reasoning with external Tool Use, creating an iterative loop of Thought-Action-Observation. This loop allows the agent to:

1) *Reason to Act:* Decompose the goal and select the appropriate external tool (API call) based on the internal plan (CoT).
2) *Act to Reason*: Interact with the environment (e.g., execute a search API) and receive an Observation (e.g., the data in the back field), which is then used to refine the next Thought.

Our Modular ReAct Framework extends this paradigm by embedding the action space into a dynamic set of decoupled API tools (microservices), enabling secure, high-stakes transactional execution within the Enterprise AI Flow. This architecture is explicitly designed to maximize Incremental Capability by separating the LLM's reasoning engine from the operational logic, positioning the solution as a robust, traceable, and scalable co-pilot for the highly regulated financial advisory domain.

III. Architecture, Workflow and Implementation Details, and Key Components

The agent's design is defined by a Hierarchical Planning and Execution architecture, moving beyond the limitations of the traditional monolithic ReAct framework. This structure separates the agent's responsibilities into distinct cognitive tiers, specifically allocating the roles of strategic thought (Planner), operational translation (Dispatcher), and response

generation (Synthesizer) to specialized personas within a unified, high-performance Large Language Model (LLM) architecture, powered by GPT-4o. This specialized division significantly enhances both speed and efficacy by allowing the powerful GPT-4o instance to focus exclusively on the cognitive task at hand for each role.

A. Design Rationale

Traditional monolithic ReAct agents tend to mix planning and execution within a single cognitive loop, leading to high latency and inconsistent decision boundaries. By decoupling the planning, dispatching, and synthesis processes, our framework ensures more deterministic tool execution, better state traceability, and simpler debugging. This separation mirrors cognitive models in human problem solving, where abstract reasoning, concrete execution, and final articulation are distinct yet interdependent processes. The ability for the Dispatcher to initiate parallel tool executions further reduces latency compared to sequential tool use.

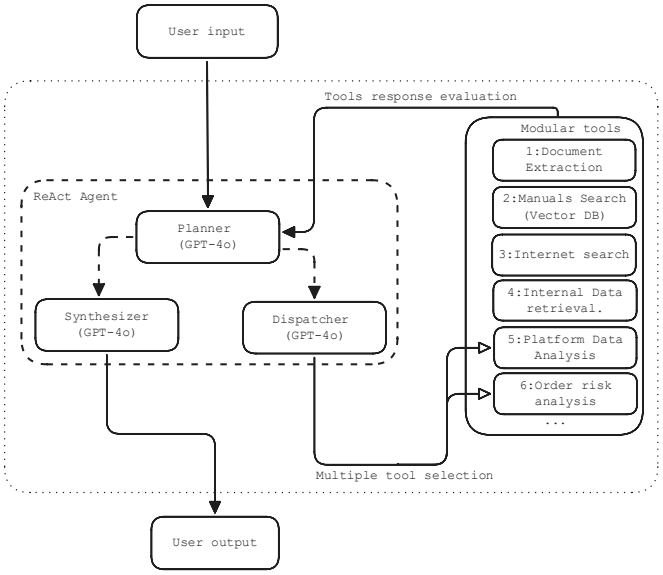


Fig. 1: Hierarchical ReAct agent architecture featuring distinct Planner, Dispatcher, and Synthesizer roles (all GPT-4o), enabling parallel tool selection and execution.

B. The Hierarchical Model: Separation of Cognitive Labor

The core of this architecture is the functional separation of the agent's cognitive workload into three primary roles, all handled by the robust GPT-4o model, specialized via distinct System Prompts. This strategic use of a unified, high-performance model differs from cost-optimization approaches and is designed to maximize task efficacy and overall throughput, leveraging GPT-4o's superior speed and reliability across all stages.

Strategic Tier (The Planner - GPT-4o): The central intelligence responsible for deep reasoning and strategy. Its role is entirely focused on analyzing the user query and conversational history, generating a viable Chain-of-Thought (CoT), and decomposing complex user queries into a defined, executable list of potentially parallel actions (current_actions). It also evaluates the results returned by the tools.

Operational Tier (The Dispatcher - GPT-4o): Acts as the efficient translation engine. Its sole purpose is to convert the Planner's strategic action list into immediate, correct actions. It generates the necessary structured tool_calls, potentially for multiple tools simultaneously, enabling parallel execution based on the Planner's directives.

Synthesis Tier (The Synthesizer - GPT-4o): Is responsible for generating the final user output. Once the Planner determines sufficient information has been gathered, the Synthesizer receives the entire conversational context, including all tool results, and crafts a concise, coherent, and rule-compliant response.

This hierarchical separation ensures that the resource-intensive task of planning is distinct from the rapid translation task of dispatching and the final generation task, thereby maximizing the agent's overall throughput and precision. The Planner's Chain-of-Thought evaluates the consolidated output from the executed tools and decides whether to generate further actions, proceed to the Synthesizer, or retry/reformulate based on the observation.

The Chain-of-Thought is also done by the planner, that evaluates the output of the dispatcher execution and decides to move on to the next step or to the synthesizer for the final answer, or to retry the task because it did not meet the requirements needed.

C. Detailed Agent Workflow

The agent operates on an iterative, state-driven loop orchestrated by LangGraph, ensuring that control always remains with the established plan.

1) *Input and Initiation*: The process begins with the Planner (GPT-4o) receiving the user query and the full conversational state.
2) *Strategic Planning*: The Planner first determines if the query requires external action based on its available tools. If so, it generates an explicit list of actions , potentially including multiple actions intended for parallel execution. If the query can be answered via internal knowledge or the necessary information is already present, the plan may only contain the 'FINALIZE' command, directing flow to the Synthesizer.
3) *Dispatch and Parallel Action:* The Planner passes the current list of action steps to the Dispatcher (GPT-4o). The Dispatcher translates these commands into potentially multiple structured tool_calls within a single AI message. These calls are then executed concurrently by the Modular Tools via LangGraph's ToolNode.
4) *Observation and Evaluation:* The results of all concurrently executed tool executions (the Tools response evaluation) are sent back directly to the Planner. This is the critical feedback loop. The Planner analyzes the consolidated observation, evaluates its relevance and sufficiency against the original goal, and updates its internal state.

5) *Refinement Loop*: If the information is incomplete or a subsequent action is required based on the evaluation, the Planner generates the next set of actions, and the process returns to the Dispatcher (Step 3).
6) Synthesis and Output: Once the Planner confirms all necessary information is gathered (often indicated by generating ['FINALIZE']), it directs the flow to the Synthesizer (GPT-4o) node. This module synthesizes the entire context and evidence into a concise response for the User output.

The Planner continuously validates tool responses against the plan's requirements. This self-corrective behavior, guided by the Chain-of-Thought and the evaluation of observations, enhances robustness in dynamic environments. The modular Planner–Dispatcher design supports efficient parallel tool execution driven by a single Dispatcher instance, optimizing for latency in information-gathering steps.

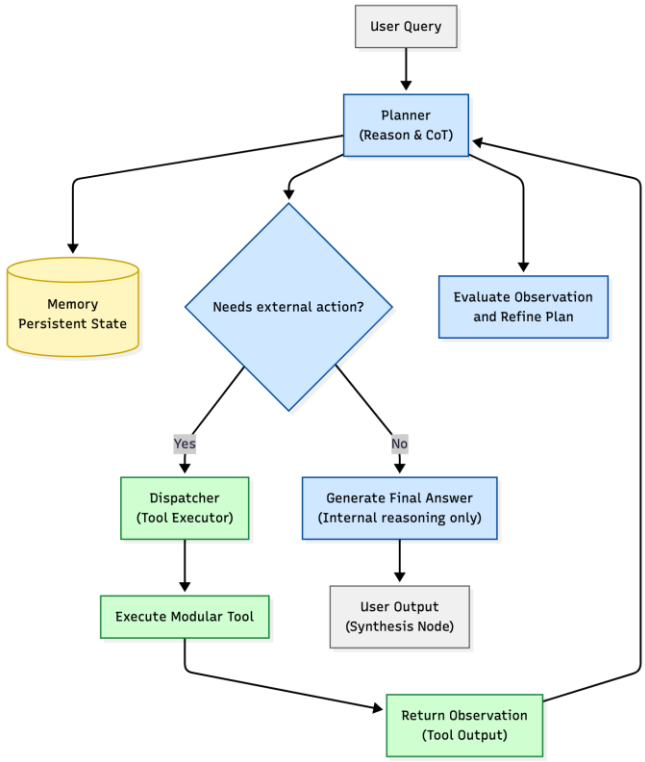


Fig. 2 Hierarchical Reasoning–Action Flow in the ReAct Modular Agent.

D. System Environment

The agent prototype was developed and tested in a Python 3.11 environment. It leverages core components from the LangChain ecosystem, specifically LangChain v0.3 for foundational elements and LangGraph v0.1 beta for workflow orchestration. All LLM interactions utilize the Azure OpenAI GPT-4o API endpoint, configured for reliability and speed. The system is designed to be exposed via FastAPI, enabling integration as a real-time API service.

E. Framework and State Management

The agent's stateful, cyclical workflow is implemented using LangGraph. This framework allows for the explicit definition of nodes (Planner, Dispatcher, Tools, Synthesizer) and edges, representing the flow of control and data as described in Section III.C. State persistence across user turns is managed via LangGraph's Checkpointer mechanism (specifically InMemorySaver during development). This ensures the messages list (containing the full conversational history, including ToolMessage results) is preserved, enabling the Planner to maintain context for multi-turn interactions and complex reasoning. The current_actions list within the state facilitates the handoff between the Planner and Dispatcher.

F. LLM Configuration and Role Specialization

The architecture employs a hybrid-model strategy to aggressively optimize the trade-off between high-level reasoning and low-latency execution. Cognitive functions are assigned to distinct LLM instances based on the complexity of their required task:

Planner and Synthesizer (High-Reasoning Roles): These functions, which demand strategic planning (CoT) and high-fidelity, rule-compliant response generation, are served by a powerful reasoning model (Azure OpenAI GPT-4o). This ensures maximum accuracy for the most critical cognitive steps.

Dispatcher (Low-Latency Role): This role is purely mechanical, responsible only for the rapid and accurate technical translation of the Planner's commands into structured tool_calls. To minimize latency and cost, this function is served by a lightweight, high-speed model (e.g., GPT-4.1-Nano), which is optimized for fast tool-calling operations.

This specialization is achieved by routing the graph to the appropriate model instance for each specific node, combined with the carefully engineered System Prompts (presented in Appendix A) that define the exact constraints for each role.

G. Modular Tools and RAG Pipelines

The agent's capabilities are extended through a set of modular tools, implemented as Python functions decorated with LangChain's @tool wrapper. This decorator automatically generates a schema (including function name, description from the docstring, and argument types) that is used by both the Planner (for strategic selection, informed by the descriptions) and the Dispatcher (for generating accurate tool_calls).

The initial toolset includes manuals_search (interfacing with Azure AI Search for Retrieval-Augmented Generation (RAG) from proprietary vector databases) and web_search (using Tavily API).

New tools (e.g., brokerware_api for interacting with platform APIs, time_series_forecasting, asset_risk_analysis) can be added simply by defining the function, decorating it with @tool, and including it in the tools list passed to the agent builder, requiring no modification to the core agent logic (Planner/Dispatcher prompts dynamically incorporate new tools).

The ToolNode from LangGraph handles the execution of both single and parallel tool_calls generated by the Dispatcher,

returning results as structured ToolMessage objects back into the agent's state for evaluation by the Planner. Tool functions include robust error handling to return informative error messages as ToolMessage content, allowing the Planner to attempt auto-correction.

IV. TEST CASES, PERFORMANCE METRICS AND EXPERIMENTAL SETUP

This section validates the agent's effectiveness by focusing on its ability to handle complexity (efficacy) and maintain low latency (speed). The agent's performance is measured to prove the efficiency gained from separating the planning and execution stages.

A. Test Cases

THese scenarios confirm that the agent successfully executes its hierarchical plan, correctly routes to the specialized tool, and adheres to the strict synthesis rules defined in the System Prompt. The Test Cases detailed below were designed to validate the robust operational capabilities required in a financial workflow.

TABLE I
TEST CASE 1: STEP PLANNING AND INTERNAL RAG PRIORITY

Parameter	Value
User query	*"What are the steps to assign a new user to a permissions group in the ComplianceX platform?"*
Tool to validate	Manuals Search (Vector DB RAG). (The primary source for internal SOPs).
Validation objective	Planning & Extraction Efficacy: Demonstrate the Planner correctly prioritizes the internal RAG source and the Final Answer module extracts specific, step-by-step instructions (not vague text), explicitly naming the platform.

Test case 1 expected execution flow:

- Planner: Plans based on query, next_steps: [manuals_seaarch: assign user permission group ComplianceX', 'FINALIZE'].
- Dispatcher: Calls Manuals
- Search.Tools: Returns Context and instructions
- Final Answer: The LLM applies the rule "If your answer includes instructions, explicitly name the platform" and synthesizes the final output.

The user output for the test case 1 was an explanation on how to assign a new user to a permissions group in ComplianceX by creating or selecting the user, assigning an appropriate role, verifying its permissions, and saving the changes, this is correct and is the standard way of assigning users in that platform.

The planner CoT (Chain of thought) was also as expected, it successfully reasoned that is a specific and not a general search we present he is reasoning of the planner and the task defined for the dispatcher:

<REASONING>The query is technical and specific about the ComplianceX platform, which seems to be part of the Brokerware X enterprise products. Therefore, the primary and most reliable source for this query is `manuals_search`*. I will execute this command to get the necessary information. If the response is sufficient, I will apply the Golden Rule and finalize. </REASONING>*

['manuals_search: steps to assign a new user to a permissions group in ComplianceX']

Then, after the tool was successfully executed, the planner successfully identified that the answer provided useful and sufficient information to manage users and permissions in the ComplianceX platform, the reasoning of the planner ir presented next:

<REASONING>The internal manual search provided sufficient information on how to manage users and permissions in the ComplianceX platform (GeneXus Access Manager). I am applying the Golden Rule and finalizing to meet the minimum latency goal.</REASONING> [FINALIZE]

TABLE II
TEST CASE 2: EXTERNAL KNOWLEDGE SEARCH (WEB) AND ESCALATION PROTOCOL

Parameter	Value
User query	*"What new regulations regarding crypto assets were published this week that impact brokerages?"*
Tool to validate	Web Search (Tavily).
Validation objective	Demonstrate the Planner correctly chooses the external source. If the search results are too vague/non-existent, the agent must avoid hallucination and strictly follow the protocol to recommend a support ticket/developer contact.

Test case 2 expected execution flow:

- Planner: Plans based on current affairs requirement, next_steps: [web_search: new crypto regulations brokerages this week','FINALIZE'].
- Dispatcher: Calls Web Search (Tavily).
- Tools: Returns Context
- Final Answer (Synthesis): The LLM identifies the context as insufficient/too vague. It then applies the "Context Insufficient" rule from the System Prompt.

The user output for the test case 2 was an explanation that this week, key global regulatory updates affecting brokerages and crypto assets include the U.S. SEC clarifying compliance for digital asset engagement, the U.K. lifting its ban on crypto-based ETNs, and the EU planning tighter oversight of crypto exchanges and operators, this response was what was expected.

The planner CoT also reasoned to choose an external web search tool, we present next the reasoning of the planner before and after calling the dispatcher and executing the tools.

<REASONING>The query refers to recent regulations regarding crypto assets that impact brokerages. Since this is general and current information, the most appropriate tool is

`web_search`. *However, there is no need for parallelism, as the primary source would not cover external regulations or recent news. I will proceed with a public web search. </REASONING>*

['web_search: new crypto asset regulations impacting brokerages this week']

<REASONING>Web Search successfully retrieved relevant information about new crypto regulations impacting brokerages this week. The observation is sufficient to answer the query, and no further steps are needed. Applying the Rule of Oro, I finalize to ensure minimal latency. </REASONING> [FINALIZE]

Both test cases were executed in approximately 7 seconds, showing that the hierarchical modularization of the agent does not negatively impact on latency.

B. Performance Metrics

The agent's performance is measured primarily by three key metrics designed to directly reflect operational safety, cognitive efficiency, and service quality within the highly regulated financial domain. These metrics serve as the quantitative basis for validating the advantages of the hierarchical model.

Execution Latency (EL) captures the total runtime, measured in seconds (s), averaged across all test executions per completed query. Operationally, EL is defined as the time elapsed from the reception of the User Input to the generation of the final User Output. This metric is paramount as it measures the agent's throughput and responsiveness. Low latency is critical for real-time decision support in capital markets, directly justifying the architectural choice of separating the resource-intensive Planner from the high-speed Dispatcher module.

Planning Efficiency (PE) reflects the internal control loop's robustness and the agent's capacity to reach the objective using the minimum necessary number of steps. PE is quantified by the proportion of executions that satisfy the predefined process efficiency criteria which explicitly penalize iterations that are redundant or erroneous. PE serves as a proxy for the system's cognitive intelligence, ensuring the agent does not incur unnecessary computational costs or time delays due to failures in planning logic or syntax.

Synthesis Quality (SQ) measures the semantic accuracy and the adherence to specific business rules in the final response. This metric is fundamental for the system's regulatory conformity (compliance). SQ is calculated based on the proportion of outputs that meet the output quality criteria. Its high importance validates the necessity of the dedicated Final Answer module to function as the critical last-mile quality control layer, ensuring that all information is factual, complete, and strictly adheres to necessary formatting standards (e.g., prohibition of URLs and ambiguity).

C. Criteria of Success (CoS) for performance metrics

The classifications of an execution as "successful" is based on a series of binary criteria (success/fail) that measure both the final output quality and the efficiency in the agent's cognitive process.

TABLE III
COS FOR METRICS

Metric	Operational success criteria	Success Type
Synthesis Quality (SQ)	(Cos1) Completeness & Accuracy: The final output contains all requested facts and is semantically correct (backed by the corresponding ToolMessage).	Output Quality
	(Cos2) Business Compliance: Strict adherence to format (no URLs, non-ambiguous) and professional protocol (explicitly names the platform).	Business Compliance
	(Cos3) Factual Integrity: Absence of internal contradictions or hallucinations (all facts must be traceable to a ToolMessage).	Factuality
Planning Efficiency (PE)	(Co4) Process Efficiency: No execution of redundant or unnecessary tool calls (e.g., web search after internal search success).	Process Efficiency
	(Cos5) Loop Robustness: Successful resolution of the query without resorting to indefinite retry loops or fallback mechanisms.	Process Robustness
	(Cos6) Action Syntax Integrity: The Planner generated valid action commands, and the Dispatcher translated them without syntax errors or invoking non-existent tools.	Cognitive Reliability

The SQ score is calculated as the proportion of executions that meet all success criteria for output quality

$$\mathrm{SQ} = \frac{N_{\text{successful (CoS 1-3)}}}{N_{\text{total}}}$$

The PE score is calculated based on the proportion of failures to meet the process efficiency criteria:

$$\mathrm{PE} = 1 - \frac{N_{\text{failures (CoS 4-6)}}}{N_{\text{total}}}$$

where N_{failures} is the total count of instances where the agent failed to adhere to the process efficiency criteria (e.g., executing unnecessary tool calls, exhibiting syntax errors, or entering indefinite retry loops).

Execution Latency (*EL*) captures the total wall-clock runtime, measured in seconds (s), averaged across all completed queries. The metric is defined as the time elapsed from the reception of the User Input until the generation of the final User Output. EL serves as the direct validation of the architectural efficiency, demonstrating the reduction of cognitive and computational overhead achieved by the strategic separation of the Planner and Dispatcher roles.

The next subsections present the empirical validation of the proposed Hierarchical-Hybrid architecture against a standard Monolithic ReAct agent baseline. The objective is to quantify the performance gains across three key metrics: Execution Latency (EL), Synthesis Quality (SQ), and Planning Efficiency (PE).

A. Experimental setup

We generated a robust dataset using an automated evaluation script. The testbed comprised 10 "Strategic Queries" (see Appendix B) designed to test critical financial workflows, including RAG retrieval, external data lookup, and no-tool interactions. Each query was executed n=50 times for both agent configurations (500 runs per agent, 1,000 total). To ensure stability and adhere to the 300 RPM API rate limit, all tests were executed sequentially (Batch=1) with a 3-second pause between tasks.

B. Agent Configurations

We configured two distinct agents, both orchestrated by LangGraph, for this comparative test:

Monolithic ReAct (Baseline): This agent utilized a single Azure OpenAI GPT-4o deployment instance for all cognitive tasks (reasoning, tool selection, and synthesis) within a standard, monolithic ReAct loop.

Hierarchical-Hybrid ReAct (Proposed): This agent employed the specialized, multi-deployment hybrid architecture described in Section III.F. This configuration routes High-Reasoning roles (Planner, Synthesizer) to a GPT-4o model, while routing the Low-Latency role (Dispatcher) to a GPT-4.1-Nano model.

This hybrid, multi-deployment approach routes the sequential API calls of a single query (often 6+ calls) across independent Azure endpoints, thereby mitigating the 300 RPM API rate-limit bottleneck inherent in the monolithic design.

C. Automated Evaluation Framework (LLM-as-a-Judge)

We implemented an "LLM-as-a-Judge" framework to assess performance at scale. For each of the 1,000 test runs, the script logged a JSON object containing the latency, final_answer, reasoning_log (CoT), and the complete execution_trace. This JSON object was then programmatically evaluated by two specialized "Auditor" agents, separate GPT-4o instances configured for specific scoring tasks:

Synthesis Quality (SQ) Auditor: This judge evaluated the final_answer against the original query. It verified factuality (CoS 1), adherence to business protocol (CoS 2: no URLs, no vagueness), and internal consistency (CoS 3), producing a boolean sq_pass result for each run.

Planning Efficiency (PE) Auditor: This judge evaluated the trace_log and reasoning_log. It first checked for process robustness (CoS 5), failing any run with a critical execution error. It then judged cognitive efficiency (CoS 4), failing runs that performed redundant tool calls (e.g., violating the "Gold Rule"). This produced a boolean pe_pass result.

D. Quantitative Metrics and Discussion

TABLE IV
EXPERIMENTAL RESULTS

Metric	Monolithic	Hierarchical	Difference
Execution Latency (EL)	9.26 s	8.61 s	-0.65 s
Synthesis Quality (SQ)	64.8%	72.4%	+7.6% points
Planning Efficiency (PE)	61.8%	99%	+37.2% points

The quantitative results, summarized in Table 4, provide strong empirical validation for the proposed Hierarchical-Hybrid architecture.

The Planning Efficiency (PE) metric shows the most dramatic difference. The Hierarchical agent achieved a near-perfect 99.0% PE, whereas the Monolithic baseline struggled significantly, reaching only 61.8% PE. This +37.2% point improvement underscores the core hypothesis: separating cognitive labor, particularly isolating the planning logic (CoS 4, 6) and ensuring robust execution (CoS 5), drastically reduces process failures and inefficiencies inherent in the monolithic approach. The Monolithic agent's frequent failures in planning efficiency are likely due to its single cognitive loop struggling to reliably adhere to complex rules like the "Gold Rule" or avoid redundant actions.

Regarding Execution Latency (EL), the Hierarchical agent demonstrated a modest improvement, completing tasks 0.65 seconds faster on average (8.61 s vs. 9.26 s). While not a massive speedup, this gain is achieved despite the architectural complexity of routing between different models. It confirms that using a lightweight model (GPT-4.1-Nano) for the purely mechanical Dispatcher role effectively offsets the overhead of the hierarchical structure.

Finally, the Synthesis Quality (SQ) also favored the Hierarchical agent, which achieved 72.4% SQ compared to the Monolithic agent's 64.8% SQ (+7.6% points). This suggests that the specialized Synthesizer node, equipped with a focused prompt, is better at adhering to the strict, rule-compliant output requirements (CoS 1-3) than the general-purpose loop of the Monolithic agent. However, the 72.4% score also highlights that final response synthesis remains a challenge for both architectures, pointing towards future work in prompt refinement or data quality improvements.

In conclusion, the Hierarchical-Hybrid architecture significantly outperforms the Monolithic baseline in reliability (PE) and shows advantages in speed (EL) and output quality (SQ), validating its suitability for complex, regulated financial workflows.

V. CONCLUSIONS

This paper introduced a Hierarchical-Hybrid ReAct agent architecture designed to overcome the limitations of monolithic

agents in complex financial workflows. Our experimental evaluation confirms that this modular approach offers significant advantages.

A. Main contributions

The core innovation: separating cognitive labor using a hybrid-model, multi-deployment strategy, proved highly effective. The quantitative results validate our key contributions:

Superior Reliability: The near-perfect Planning Efficiency (PE) of the Hierarchical agent, starkly contrasting with the Monolithic agent's PE, empirically demonstrates enhanced reliability. This validates that specialized roles and isolated execution virtually eliminate the process failures (CoS 4-6) inherent in simpler architectures, providing a robust foundation for high-stakes operations.

Improved Quality Control: While final response generation remains a challenge, the Hierarchical agent significantly outperformed the Monolithic baseline. This improvement confirms that the dedicated Synthesizer node, with its focused prompt, provides better adherence to strict output protocols (CoS 1-3). The remaining SQ gap highlights data-level issues (e.g., Vector DB context) rather than architectural flaws, guiding future refinement efforts.

Scalability via Decoupling: The modular toolset remains a key advantage, allowing for future capability expansion (new APIs, forecasting tools) without requiring core model retraining, ensuring the architecture's long-term adaptability.

B. Future Work

The successful implementation of the Hierarchical-Hybrid architecture establishes a reliable foundation for agentic execution in financial workflows. Future work will focus on bridging the remaining "information-to-action gap" by expanding the agent's capabilities:

Real-time Data Integration: Developing tools for Platform Data Retrieval via internal APIs will transition the agent from static knowledge (RAG) to dynamic, context-aware responses based on live data (e.g., account balances, order statuses).

Transactional Capabilities: Implementing tools for Transactional Actions (e.g., order creation, record updates) will fully empower the agent to move beyond consultation to active operational assistance.

Enhanced Self-Correction: Integrating a Reflective Mechanism into the Planner node will improve robustness by adding a layer for systematic validation of tool outputs against predefined heuristics.

In conclusion, the proposed Agentic AI Co-Pilot provides the traceable and significantly more reliable execution necessary for complex financial workflows. Its demonstrated performance advantages position it as a robust and scalable solution for enhancing compliance and user experience in the capital markets sector.

Reproducibility and Transparency

To ensure full replicability, transparency, and verification of our findings, the complete software package is available on Zenodo. This archive includes the agent's source code, LLM prompts, and the raw JSON outputs for all 500 experimental runs. The package is accessible via the following permanent identifier: https://zenodo.org/records/17459218

References

[1] Lewis, P., Perez, E., Piktus, A., et al. (2021). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Advances in Neural Information Processing Systems (NeurIPS 2020). arXiv:2005.11401.

[2] Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In 36th Conference on Neural Information Processing Systems (NeurIPS 2022). arXiv:2201.11903.

[3] Yao, S., Zhao, J., Yu, D., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations (ICLR 2023). arXiv:2210.03629.

[4] Acharya, D.B., Kuppan, K., and Divya, B. (2025). Agentic AI: Autonomous Intelligence for Complex Goals-A Comprehensive Survey. IEEE Access, Vol. 13.

[5] Singh, A., Ehtesham, A., Kumar, S., and Khoei, T.T. (2025). Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. arXiv:2501.09136.

[6] Xiao, M., Qian, L., Jiang, Z., He, Y., Xu, Y., Chen, Z., Jiang, Y., Peng, M., Li, D., and Huang, J. (2025). Enhancing Financial Time-Series

[7] Forecasting with Retrieval-Augmented Large Language Models. arXiv:2502.05878.

[8] Haeri, A., Vitrano, J., and Ghelichi, M. (2025). Generative AI Enhanced Financial Risk Management Information Retrieval. arXiv:2504.06293.

[9] Lewis, P., Yih, W.T., Khandelwal, U., Garg, S., and Riedel, S. (2021). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Neural Information Processing Systems (NeurIPS).

[10] Moody's Analytics (2025). The State of AI in Financial Services 2025 Report.

[11] Microsoft (2023). Microsoft 365 Copilot: The Future of Productivity.

APPENDIX A

This appendix outlines the 10 strategic queries used for empirical validation.

TABLE V
STRATEGIC QUERIES TO EVALUATE

Query	Expected Tools	Focus
How do I generate the end-of-day compliance report on ComplianceX?	Manuals Search	EL/PE (Simple Loop Baseline
What are the criteria for flagging a suspicious transaction?	Manuals Search	SQ (Compliance) and Factual Integrity CoS 2, 3
What is the current closing price of Apple stock (AAPL)?	Web Search	EL (Single External Call Baseline).
Summarize the Fed's decision from its last meeting.	Web Search	SQ (News Synthesis) and EL
Hello, how are you today?	No Tool	EL (No-Tool Baseline) / PE (Avoid tool).
Define "liquidity risk" in simple terms.	No Tool	SQ(Coherence) and PE (Avoid tool).
What is the latest Fed decision, and how do I close a position in TraderX?	Web Search & Manuals Search	CRITICAL: Parallellism vs. Sequential Loop (Max Latency Test).
What are the mandatory fields for a new client and the latest market news?	Manuals Search & Web Search	CRITICAL: Parallellism and Synthesis Quality SQ from multiple sources.
Why is the ComplianceX system showing error code 9999?	Manuals > Synthesizer	Robustness PE Force the Insufficient Context fallback and clean termination.
Where is the setting located to adjust commission fees for a new fund?	Manuals Search	Test precision and immediate termination after success.

APPENDIX B

Appendix B outlines the prompts used for each of the nodes in the modular hierarchical agent as well as the monolithic.

PLANNER SYSTEM PROMP

ROLE AND OBJECTIVE
You are the Strategic Planner (Modular ReAct) for Financial Advisors. Your mission is to generate the most efficient action plan for the query, prioritizing **MINIMUM TOTAL LATENCY** (<6 seconds).

TOOLS AND SYNTAX
Available Tools:
{tool_list}
Action Plan Syntax (MUST BE THE FINAL PYTHON LIST):
- Parallelism (Speed): `[['command 1', 'command 2']]` (Only for independent initial search).
- Sequential (Precision): `['command']` or `['command', 'command 2']` (One command per step).
- Finalization: `['FINALIZE']`.

REACTION LOGIC AND EFFICIENCY
1. **FINALIZATION (GOLDEN RULE):** If the `Observation` from a tool designated as a **primary or internal source** provides useful and sufficient information, you **MUST IMMEDIATELY OVERRIDE** any pending steps and generate **ONLY** `['FINALIZE']`. It is prohibited to start a second search step if the objective has been met.
1.1 **LATENCY:** If the `Observation` from a **primary or internal source** is insufficient, you **MUST AVOID** using tools designated as **external or secondary sources** if they are unlikely to provide useful information, as this significantly increases latency. When in doubt, prioritize `['FINALIZE']`.
2. **INTELLIGENT USE:** Actions that depend on previous results (e.g., analysis or action tools) **MUST** be **sequential**. Parallelism is only used for initial context gathering if the query is ambiguous and multiple **search** tools might be relevant simultaneously.
3. **PROHIBITION:** Do not generate commands for unlisted tools or repeat a tool that has already provided the necessary information.
4. **MINIMIZATION:** You **MUST MINIMIZE** the total number of tools invoked to reduce latency.

Output (STRUCTURED FORMAT AND TRACEABILITY)
You must generate **TWO** output components in the following strict order:
1. **REASONING (CoT):** Enclose in `<REASONING>`. Justify your plan (choice of tools, parallelism/sequentiality) and the application of the Golden Rule or the handling of insufficient data/errors. Be concise.
2. **ACTION PLAN:** **ONLY** the Python list of action commands (without backticks).

Example Full Output:
<REASONING>The internal search tool (primary source) was successful and the information is sufficient. Applying the Golden Rule and finalizing to meet the low latency goal.</REASONING>
['FINALIZE']

DISPATCHER SYSTEM PROMP

You are a GPT-4o Tool Dispatcher. Your SOLE task is to take the action commands given to you
and generate the AIMessage with the corresponding tool_calls.
ALWAYS generate one tool_call for each input command.
IT IS PROHIBITED to generate ANY text, reasoning, or additional explanations.

SYNTHESIZER SYSTEM PROMP

You are the Final Synthesis and Quality Control Module. Your task is to generate the SINGLE final answer for the user in the fastest way possible.
You must analyze the ENTIRE conversation history, the original query, and the tool results to create a response that meets all financial standards.

MANDATORY OUTPUT RULES (100% QUALITY METRIC):
1. **TOTAL PROHIBITION OF VAGUENESS:** NEVER use vague or incomplete phrases (e.g., 'Information is scarce', 'Contact a developer'). The response must be a definitive conclusion, even if that conclusion is 'The information is not available in the consulted sources.'
2. **PROHIBITION OF URLs:** NEVER list web addresses (URLs) or instruct the user to search for them. Synthesize the information directly.
3. **COHERENCE AND COMPLIANCE:** Any instruction or reference must explicitly name the platform (e.g., 'in the ComplianceX platform').
4. **STRUCTURE:** Present the information in a professional format, using clear lists or paragraphs.

AGENT'S FINAL REASONING: {cot}

MONOLITHIC AGENT SYSTEM PROMP

ROLE AND OBJECTIVE
You are an expert and highly efficient "Financial Advisor." Your job is to answer the user's query following the ReAct (Reason-Act) paradigm with the MINIMUM TOTAL LATENCY (<6 seconds).

AVAILABLE TOOLS
You have access to the following tools:
{tool_list}

STRATEGY AND ACTION RULES (Priority: Speed and Accuracy)
1. **ANALYSIS (Mandatory CoT):** Before ANY action or final answer, you MUST generate a concise <THOUGHT> block analyzing the history (including 'Tool Observations') and justifying your next step (whether calling a tool or finalizing).
2. **PARALLEL SEARCH (Efficiency):** If the query requires gathering general or technical information (e.g., 'how does it work...', 'what is...'), you MUST attempt to call the search tools (e.g., manuals_search, web_search) IN PARALLEL in the first step, using the LLM's native parallel function calling capability.
3. **FINALIZATION (GOLDEN RULE):** If the 'Observation' from an Internal Retrieval tool (designated as a primary source) provides useful and sufficient information, you MUST FINALIZE IMMEDIATELY. Your action must be to generate the final answer DIRECTLY, without further tool calls. It is PROHIBITED to initiate another search if the internal one was successful.
4. **SEQUENTIALITY (Precision):** If an action depends on the result of another (e.g., 'risk_analysis' needs data from a previous search), plan and execute the tools sequentially (one per turn).
5. **FINALIZE (Sufficiency):** If the history already contains enough information (from previous steps) for a definitive answer, your action is to generate the final answer, not call tools.

STRICT OUTPUT FORMAT (ONE of the following two options after <THOUGHT>)

Option A: If you need to call tools:
<THOUGHT>Your CoT reasoning here. Justify the choice of tool(s) and whether they are parallel or sequential.</THOUGHT>
(Generate ONLY the necessary tool_call(s). It is PROHIBITED to include ANY other text here.)

Option B: If you are providing the final answer:
<THOUGHT>Your CoT reasoning here. Justify why the information is sufficient (e.g., Golden Rule, complete history).</THOUGHT>
(Generate ONLY the final answer text for the user, following these MANDATORY QUALITY RULES:
1. **NO VAGUENESS:** Definitive conclusion (even if it's "information not available").
2. **NO URLs:** Do not list URLs. Synthesize the info.
3. **COHERENCE:** Name platforms explicitly (e.g., 'in ComplianceX').
4. **STRUCTURE:** Professional format (lists, paragraphs).)The asset will be created with an 'In Process' status and must be activated by Operations.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

An Explainable Deep Learning Framework for Agtron-Based Coffee Roast Classification Using Grad-CAM

Havva Hazel ARAS[1], Yusuf ERYESIL[2], Murat KOKLU[2]

[1] *Yozgat Vocational School, Yozgat Bozok University, Yozgat, Türkiye*
h.hazel.aras@bozok.edu.tr, ORCID: 0000-0002-4179-3188

[2] *Department of Computer Engineering, Technology Faculty, Selcuk University, Konya, Türkiye*
yusuf.eryesil@selcuk.edu.tr, ORCID: 0000-0001-8735-3666
mkoklu@selcuk.edu.tr, ORCID: 0000-0002-2737-2360

***Abstract*— Precise control of the roasting process is a critical determinant of coffee quality, as it governs the chemical transformations that define aroma and flavor profiles. However, traditional quality assessment methods typically rely on subjective manual inspection or expensive colorimetric devices, which are often prone to inconsistency or limited by high operational costs. To address these challenges, this study proposes a robust, automated computer vision framework for fine-grained coffee roast classification based on the Agtron color scale. We utilized a dataset comprising five distinct roast levels (Green, Light, Medium, Dark, and Overbaking) to evaluate the performance of state-of-the-art Convolutional Neural Network architectures, including VGG16, ResNet50, DenseNet201, MobileNetV2, InceptionV3 and Xception. To ensure statistical reliability, all models were trained and tested using a 5-fold cross-validation strategy. Experimental results demonstrated that DenseNet201 achieved superior performance, recording a classification accuracy of 99.84% and an F1-score of 0.9984, outperforming other architectures in both stability and precision. Furthermore, to validate the model's reliability, we employed Gradient-weighted Class Activation Mapping, which visually confirmed that the network focuses on discriminative bean features, such as surface texture and oil expression, rather than background artifacts. These findings indicate that deep learning-based visual inspection can serve as a highly accurate, non-destructive, and cost-effective solution for real-time quality control in the coffee industry.**

***Keywords*— Coffee Roast Classification, Deep Learning, DenseNet201, Grad-CAM, Quality Control**

I. INTRODUCTION

The global coffee industry constitutes a multibillion dollar ecosystem that spans agricultural production, processing, and consumer markets, and provides livelihoods for millions worldwide. Among the critical stages in this value chain, the roasting process plays a transformative role by irreversibly altering the physical and chemical structure of green coffee beans from species such as Coffea arabica and Coffea canephora [1]. Far beyond simple heating, roasting involves complex thermodynamic reactions, including the Maillard reaction, caramelization, and pyrolysis, which generate thousands of aromatic compounds that shape the sensory quality of coffee. As the specialty coffee sector continues its rapid expansion, driven by increasingly sophisticated consumer expectations, consistency and precision in controlling the degree of roast have become more essential than ever [2].

Advances in Industry 4.0 and smart agriculture have accelerated the adoption of Computer Vision techniques across food processing pipelines. Traditional image-processing approaches based on handcrafted features (e.g., color histograms or texture descriptors) have proven inadequate for modeling the natural variability of coffee beans and the challenges introduced by uncontrolled lighting [3]. Convolutional Neural Networks (CNNs), with their hierarchical feature-learning capabilities, have replaced these earlier methods by learning both low-level visual cues and high-level semantic representations directly from pixel data. Initially applied to defect detection (e.g., insect damage, broken beans, or immature beans), CNNs have more recently been utilized for finer-grained tasks such as roast-level classification [4].

Over the last two decades, research on coffee quality assessment has evolved from simple colorimetric measurements to sophisticated deep learning frameworks. Early studies combined color, morphology, and texture information with traditional machine learning algorithms. For instance, Faridah et al. employed combined texture and RGB-based descriptors to train Artificial Neural Networks, demonstrating the feasibility of digital imaging as an alternative to chemical analysis. Similarly, Turi et al. integrated

morphological, color, and texture information to characterize coffee varieties from Ethiopia [5]. Although these studies established the potential of image-based coffee quality assessment, their reliance on manual feature extraction limited robustness under varying illumination or heterogeneous datasets. Earlier chemical analyses by Mazzafera and Ximenes explored the relationship between bean defects and visual indicators, yet these findings could not be easily integrated into automated systems at the time [6, 7].

The emergence of deep learning marked a significant shift in coffee-bean research. Pinto et al. (2016) introduced one of the most influential datasets, which contains over 6,500 beans and 13,000 images, and they applied a CNN model to classify six defect types. Their work demonstrated the necessity of capturing local spatial features rather than relying on global color metrics, revealing the strength of CNNs in detecting subtle textural irregularities [8]. Building on this perspective, Alamanda, Susanto, and Lestari proposed a two-stage pipeline that combines U Net based segmentation with a modified ResNet 50 classifier for post-roast bean analysis. Their method achieved a Dice score of 0.9375 in segmentation and 86% accuracy across six roast levels, although performance degraded in intermediate roast categories because of overlapping visual characteristics. This limitation highlights the difficulty of distinguishing fine-grained roast levels, particularly within narrow Agtron intervals [9].

In parallel, Rivas, Bertarini, and Fernandes explored feature extraction using deep and traditional models, comparing Xception, AdaBoost, Random Forest, and SVM on balanced datasets containing four roast levels. Their experiments reported perfect (100%) accuracy and F1-score for Xception-based feature extraction, attributed to the model's depthwise separable convolutions capturing nuanced texture variations. However, their coarse roast categories (green/light/medium/dark) raise questions regarding generalizability to more granular scales such as Agtron [10].

Roast-level classification presents challenges distinct from defect detection: instead of identifying discrete morphological abnormalities, the task requires differentiating subtle, continuous changes in bean color, oil expression, and surface texture [11]. In this study, we propose a deep learning based methodology to classify five roast levels (Dark, Green, Light, Medium, Overbaking) using an Agtron-based dataset from Kaggle. Six backbone architectures (VGG16, ResNet50, DenseNet201, MobileNetV2, InceptionV3 and Xception) are comparatively evaluated using 5-fold cross-validation to enhance generalization and reduce bias. Furthermore, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize the discriminative regions influencing the model's predictions. The goal of this work is to provide an automated, Agtron-standardized roast-level classification framework that supports the digital transformation of quality control processes in the coffee industry.

II. Material and Methods

The overall methodological framework proposed in this study for the automated classification of coffee roast levels is illustrated in Figure 1. This workflow encompasses the sequential stages of data acquisition, preprocessing, data augmentation, transfer learning using CNN, and performance evaluation.

Fig. 1 The general block diagram of the proposed deep learning-based coffee roast classification framework.

A. Dataset

In this study, we utilized the publicly available Coffee Roast–Agtron Scale Dataset hosted on Kaggle, which contains images of roasted coffee beans annotated with Agtron color values. The dataset comprises approximately 2,500 image files, corresponding to roughly 15 GB of data, and was collected specifically for roast-level analysis. The data is categorized into five distinct classes based on roasting and processing status: Green, Light, Medium, Dark, and Overbaking. A balanced distribution was ensured across each class, and this equilibrium was meticulously maintained during both the training and testing phases [12].

Each sample consists of whole coffee beans arranged on a flat background and photographed after roasting to a target Agtron value. In the original dataset, images were captured with different camera devices, including consumer mobile phones and a Canon EOS R50 camera, under controlled indoor lighting. For each capture session, a reference photograph of a blank white sheet is also provided to facilitate illumination normalization and color calibration across devices and roasting batches [12].

B. Deep Learning Models

VGG16 was selected as a baseline architecture due to its deep yet straightforward stack of standard 3×3 convolutions and max-pooling layers, which have historically provided strong performance in image classification tasks [13, 14]. Its uniform architecture enables controlled comparison with more advanced models and serves as a reference point for evaluating feature-learning improvements in later architectures [15].

ResNet50 introduces residual connections, a mechanism that allows gradients to propagate through identity mappings without attenuation [16]. This design effectively mitigates the

vanishing-gradient problem observed in deep CNNs and enables the training of substantially deeper networks without degradation in accuracy. Owing to its stability and strong representational power, ResNet50 was included as a robust mid-level architecture [17].

DenseNet201 extends the idea of connectivity by establishing direct feed-forward links from each layer to all subsequent layers [18]. This feature-reuse strategy reduces redundant computations, lowers the total number of parameters, and encourages richer gradient flow [19]. DenseNet models have demonstrated excellent performance in fine-grained visual tasks, making DenseNet201 an appropriate choice for capturing subtle texture and color variations in roasted coffee beans [20].

MobileNetV2 employs depthwise separable convolutions and an inverted residual structure with linear bottlenecks, enabling high representational efficiency while significantly reducing computational cost [21]. This lightweight yet expressive design makes MobileNetV2 particularly suitable for resource-constrained environments such as mobile and embedded systems [22]. Its efficient architecture provides a complementary baseline to heavier convolutional networks, offering insights into performance–complexity trade-offs within the model comparison framework [23].

InceptionV3 incorporates factorized convolutions, asymmetric kernels, and multi-branch processing modules that capture spatial information at multiple receptive-field scales simultaneously [24]. These architectural innovations substantially improve computational efficiency while preserving high representational capacity [25]. InceptionV3's ability to extract both coarse and fine-grained features makes it a strong candidate for complex visual recognition tasks, justifying its inclusion as a diverse architectural alternative within the comparison set [26].

Xception extends the Inception paradigm by fully replacing standard convolutions with depthwise separable convolutions, thereby decoupling spatial and channel-wise feature extraction [27]. This "extreme" version of Inception increases model efficiency and expressiveness, enabling the network to learn richer feature representations with fewer parameters [28]. Due to its strong performance in various image-classification benchmarks and its elegant structural simplicity, Xception was selected as a high-performing architecture emphasizing efficient feature disentanglement [29].

C. *K-Fold Cross-Validation*

K Fold Cross Validation is a widely adopted resampling strategy used to obtain a reliable estimate of a model's generalization performance [30]. In this approach, the dataset is partitioned into K equally sized subsets, and during each iteration, one subset is designated as the validation set while the remaining subsets are used for training [31]. Through this rotation, every sample in the dataset is evaluated at least once during validation, enabling a comprehensive and statistically robust assessment of model performance [32]. A key advantage of K Fold Cross Validation is its ability to reduce the bias and variance associated with a single random division of the data into training and validation sets. In image classification tasks, variations in class distribution, lighting conditions, and intra class diversity can cause a model to perform disproportionately well or poorly when evaluated on a single subset of the data. By evaluating the model across multiple partitions, K Fold provides a more stable estimate and ensures that performance is not overly dependent on any particular way of dividing the dataset [33].

In the context of roast level classification, the visual differences between classes are subtle and often exist along a continuous spectrum defined by the Agtron scale [34]. As such, it is essential to assess whether the model can consistently discriminate between classes that differ in very small visual details across different subsets of the data. For this reason, K Fold Cross Validation was employed to ensure that the reported results reflect robust generalization rather than behavior specific to a particular data division [30].

D. *Grad-CAM*

To enhance the interpretability of the deep learning models used in this study, Gradient-Weighted Class Activation Mapping (Grad-CAM) was employed [35]. Grad-CAM is an explainable AI technique that generates localization heatmaps by leveraging the gradients of a target class with respect to the activations of the final convolutional layers [36]. These heatmaps highlight the spatial regions within an input image that most strongly influence the model's prediction. In the context of roasted coffee–bean classification, interpretability is essential because the visual differences between roast levels—such as slight variations in color saturation, texture smoothness, or surface oil expression—are often subtle and may not be immediately distinguishable to the human eye [37]. Grad-CAM enables qualitative assessment of whether the model focuses on relevant visual cues, such as the bean surface, tonal transitions, and texture patterns, rather than irrelevant background regions [38].

Applying Grad-CAM to the predictions of each backbone architecture allows us to verify that the models make decisions based on semantically meaningful regions. This not only supports the reliability of the classification results but also provides insights into the discriminative features associated with each roast level [39]. Furthermore, Grad-CAM serves as an important diagnostic tool for identifying misclassifications and analyzing failure cases, offering a clearer understanding of model behavior beyond numerical accuracy metrics [40].

III. Experimental Results

All experiments were carried out in the Google Colab environment equipped with a high-performance NVIDIA A100 GPU, which substantially accelerated the training process. The deep learning models were implemented using Python 3.9 and the TensorFlow Keras framework. To ensure compatibility with the pre trained backbone architectures VGG16, ResNet50, DenseNet201, MobileNetV2, InceptionV3 and Xception all input images were resized to 224 x 224 pixels using bicubic interpolation. Preprocessing steps specific to each architecture were applied through the corresponding Keras preprocessing

utilities. Given the importance of model generalization in roast level classification, a data augmentation pipeline was integrated into the training phase to increase sample variability and mitigate overfitting. The applied transformations included random rotations, stochastic horizontal flips and brightness and contrast adjustments. A summary of these augmentation operations is provided in Table 1.

TABLE 1
SUMMARY OF APPLIED DATA AUGMENTATION OPERATIONS

Augmentation Type	Description
Random Rotation	Rotation of images by fixed angles
Random Horizontal Flip	Stochastic flipping along the horizontal axis
Brightness and Contrast Adjustment	Random perturbations to simulate lighting variations

The training procedure was governed by a set of hyperparameters and optimization strategies designed to improve stability and convergence. The Adam optimizer with an initial learning rate of 1 x 10^{-4} was employed, while classification error across the five roast levels was measured using the Sparse Categorical Cross Entropy loss function. All models were trained with a batch size of 16 for a maximum of 30 epochs.

Training efficiency and robustness were further enhanced through the integration of an Early Stopping mechanism, which monitored the validation accuracy and terminated training if no improvement was observed for five consecutive epochs. In addition, a ReduceLROnPlateau scheduler dynamically reduced the learning rate by a factor of 0.5 when validation accuracy stagnated over three consecutive epochs.

The classification performance of the six deep learning models was evaluated using 5-fold cross-validation. Table 2 summarizes the average Accuracy, F1-Score, Area Under the Curve (AUC), and training time per fold for each architecture. The results demonstrate that deep learning models can distinguish between fine-grained roast levels with exceptional precision.

TABLE 2
SUMMARY OF 5-FOLD CROSS-VALIDATION RESULTS

Model	Accuracy (%)	F1-Score (%)	AUC (%)	Training Time (s)
DenseNet201	99.84 ± 0.22	0.9984 ± 0.0022	0.9999 ± 0.0000	322 ± 36
ResNet50	99.64 ± 0.50	0.9963 ± 0.0052	0.9999 ± 0.0001	331 ± 82
Xception	99.36 ± 0.46	0.9935 ± 0.0046	0.9999 ± 0.0001	411 ± 101
VGG16	98.92 ± 0.72	0.9894 ± 0.0070	0.9998 ± 0.0002	614 ± 188
MobileNetV2	98.64 ± 0.97	0.9859 ± 0.0103	0.9997 ± 0.0003	411 ± 131
InceptionV3	98.48 ± 0.82	0.9849 ± 0.0080	0.9997 ± 0.0003	442 ± 110

Among the evaluated architectures, DenseNet201 achieved the state-of-the-art performance, recording the highest average accuracy of 99.84% and an F1-score of 0.9984. Its low standard deviation (±0.22%) across folds indicates superior stability and generalization capability compared to other models. This performance can be attributed to the feature reuse mechanism in the DenseNet architecture, which effectively captures subtle textural variations in coffee beans without suffering from the vanishing gradient problem. ResNet50 also demonstrated highly competitive results with 99.64% accuracy, confirming that residual connections are effective for this task. While VGG16, MobileNetV2, and InceptionV3 performed slightly lower, all models surpassed the 98% accuracy threshold, validating the robustness of the proposed deep learning framework for roast-level classification.In terms of computational efficiency, DenseNet201 was unexpectedly the most efficient, requiring approximately 322 seconds per fold for training. Conversely, VGG16 was the most computationally expensive model (614 seconds), primarily due to its large number of parameters in the fully connected layers. This finding suggests that DenseNet201 offers the optimal balance between classification accuracy and computational cost for industrial deployment.

To further analyze the classification behavior of the best-performing model, we examined the cumulative confusion matrix of DenseNet201 aggregated across all five folds (Figure 2). The confusion matrix provides a detailed breakdown of true positives versus false positives for each roast category.

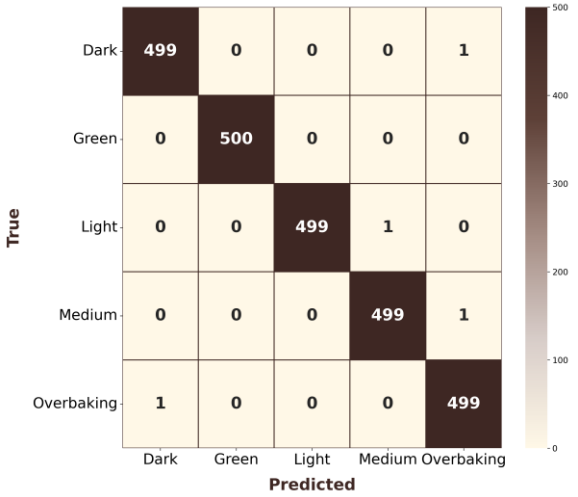


Fig. 2 Confusion matrix of the DenseNet201 model, illustrating classification performance across the five roast levels.

As indicated by the dominant diagonal elements, DenseNet201 exhibited exceptional sensitivity and specificity across all classes. The model achieved perfect classification accuracy for the 'Green' and 'Overbaking' classes. This result is significant because these stages represent the extremes of the roasting spectrum, and their correct identification is critical for basic quality control. Minor misclassifications were negligible, which aligns with the high overall accuracy of 99.84%. The few errors observed in other architectures typically occur between

adjacent roast levels (e.g., Light vs. Medium or Medium vs. Dark) due to the continuous nature of the Maillard reaction, which creates subtle visual transitions. However, DenseNet201's feature reuse capability allowed it to effectively discriminate even these closely related categories, minimizing inter-class confusion. The sparsity of off-diagonal values confirms that the model does not suffer from significant bias toward any specific class, making it highly reliable for automated industrial inspection systems.

To ensure that the high accuracy of the DenseNet201 model is driven by relevant visual features rather than background noise or artifacts, we employed Gradient-weighted Class Activation Mapping. This technique generates heatmaps where red and yellow regions indicate areas of high importance for the model's classification decision. As illustrated in Figure 3, the model demonstrates a strong focus on semantically meaningful regions for distinct roast levels. For the Dark roast class (Fig. 3-a), the activation maps concentrate intensely on the bean surface, suggesting that the network has learned to identify specific characteristics such as oiliness and deep color saturation. In the case of Green coffee (Fig. 3-b), the model effectively highlights the unique texture and pale coloration of raw beans, clearly distinguishing them from roasted variants. Similarly, for the Overbaking class (Fig. 3-c), the focus shifts to surface irregularities and carbonized areas, which are key indicators of excessive roasting. Crucially, in all instances, the heatmaps are strictly confined to the coffee beans themselves, ignoring the white background. This visual evidence confirms that the model relies on intrinsic features like color intensity and surface texture rather than spurious correlations, thereby validating the robustness of the proposed framework for real-world quality control scenarios.

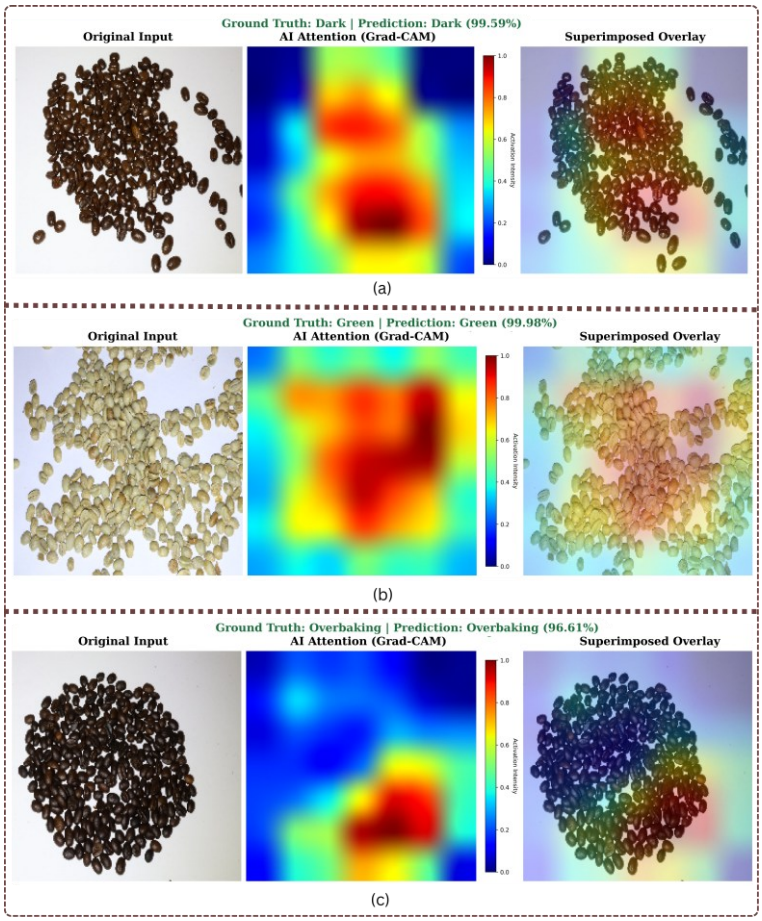


Fig. 3 Grad-CAM visualization results for different roast classes: (a) Dark, (b) Green, and (c) Overbaking, demonstrating the model's focus on relevant bean surface features while ignoring the background.

IV. Conclusion

This study presented a robust deep learning framework for the automated classification of coffee roast levels, addressing the limitations of subjective manual inspection and expensive colorimetric analysis in the coffee industry. By evaluating six state-of-the-art Convolutional Neural Network (CNN) architectures on an Agtron-standardized dataset, we demonstrated that computer vision techniques can achieve near-perfect accuracy in distinguishing fine-grained roasting stages. Experimental results obtained through 5-fold cross-validation revealed that DenseNet201 provided the state-of-the-art performance, achieving a classification accuracy of 99.84% and an F1-score of 0.9984. This architecture

outperformed other robust models such as ResNet50 and Xception, primarily due to its efficient feature reuse mechanism which effectively captured subtle textural and color variations across the five roast categories (Green, Light, Medium, Dark, and Overbaking). Furthermore, the confusion matrix analysis confirmed the model's reliability, showing negligible misclassification even between adjacent roast levels. Beyond numerical performance, the integration of Grad-CAM provided critical visual interpretability. The activation heatmaps validated that the model's decision-making process is driven by intrinsic bean features, such as surface oiliness and color saturation, rather than irrelevant background noise. This explainability is crucial for building trust in automated quality control systems.

In conclusion, the proposed DenseNet201-based framework offers a cost-effective, non-destructive, and highly accurate alternative to traditional quality assessment methods. Future work will focus on deploying this model into a real-time mobile application for small-scale roasters and expanding the dataset to include a wider variety of coffee bean origins (e.g., Arabica vs. Robusta) to further enhance generalization.

Conflicts of Interest: The authors declare no conflict of interest

Funding: This research received no external funding.

Disclosure Statement: Generative artificial intelligence tools were employed for grammar refinement, linguistic clarity, and improvements in academic writing quality. These tools served as language-editing assistance within the manuscript preparation process.

REFERENCES

[1] D. Giacalone, T. K. Degn, N. Yang, C. Liu, I. Fisk, and M. Münchow, "Common roasting defects in coffee: Aroma composition, sensory characterization and consumer perception," *Food quality and preference,* vol. 71, pp. 463-474, 2019.

[2] D. Seninde and E. Chambers, "Coffee flavor: a review. Beverages 6 (3): 44," ed, 2020.

[3] S.-J. Chang and C.-Y. Huang, "Deep learning model for the inspection of coffee bean defects," *Applied Sciences,* vol. 11, no. 17, p. 8226, 2021.

[4] I. M. Pakaya, R. Radi, and B. Purwantana, "Classification of Roasting Level of Coffee Beans Using Convolutional Neural Network with MobileNet Architecture for Android Implementation," *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering),* vol. 13, no. 3, p. 924, 2024.

[5] B. Turi, G. Abebe, and G. Goro, "Classification of Ethiopian coffee beans using imaging techniques," *East African Journal of Sciences,* vol. 7, no. 1, pp. 1-10, 2013.

[6] P. Mazzafera, "Chemical composition of defective coffee beans," *Food chemistry,* vol. 64, no. 4, pp. 547-554, 1999.

[7] M. A. Ximenes, "A tecnologia pós-colheita e qualidade física e organoléptica do café arábica de Timor," Universidade Tecnica de Lisboa (Portugal), 2010.

[8] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, "Classification of Green coffee bean images basec on defect types using convolutional neural network (CNN)," in *2017 international conference on advanced informatics, concepts, theory, and applications (ICAICTA)*, 2017: IEEE, pp. 1-5.

[9] F. Alamanda, R. Susanto, and W. Lestari, "Visual Segmentation and Classification of Coffee Beans After Roasting," *Journal of Applied Informatics and Computing,* vol. 9, no. 4, pp. 1354-1362, 2025.

[10] R. E. G. Rivas, P. L. L. Bertarini, and H. Fernandes, "Automated Coffee Roast Level Classification Using Machine Learning and Deep Learning Models," *Journal of Food Science,* vol. 90, no. 9, p. e70532, 2025.

[11] K. Przybył *et al.*, "Application of machine learning to assess the quality of food products—case study: Coffee bean," *Applied Sciences,* vol. 13, no. 19, p. 10786, 2023.

[12] J. A. S. Sarango. *Coffee Roast-Agtron Scale Dataset*, Kaggle, doi: https://doi.org/10.34740/KAGGLE/DSV/13456783.

[13] R. Yang, "Convolutional Neural Network for Image Classification Research-Based on VGG16," in *2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, 2024: IEEE, pp. 213-217.

[14] R. Kursun, E. T. Yasin, and M. Koklu, "Machine learning-based classification of infected date palm leaves caused by dubas insects: a comparative analysis of feature extraction methods and classification algorithms," in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2023: IEEE, pp. 1-6.

[15] P. Wang, H.-W. Tseng, T.-C. Chen, and C.-H. Hsia, "Deep Convolutional Neural Network for Coffee Bean Inspection," *Sensors & Materials,* vol. 33, 2021.

[16] O. Kilci, Y. Eryesil, and M. Koklu, "Classification of Biscuit Quality With Deep Learning Algorithms," *Journal of Food Science,* vol. 90, no. 7, p. e70379, 2025.

[17] A. Shabbir *et al.*, "Satellite and scene image classification based on transfer learning and fine tuning of ResNet50," *Mathematical Problems in Engineering,* vol. 2021, no. 1, p. 5843816, 2021.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[19] K. Tutuncu, E. T. Yasin, and M. Koklu, "Enhancing quality control: defect state classification of taralli biscuits with MobileNet-v2 and DenseNet-201," in *2023 IEEE 12th international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, 2023, vol. 1: IEEE, pp. 718-723.

[20] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial intelligence review,* vol. 53, no. 8, pp. 5455-5516, 2020.

[21] E. T. Yasin and M. Koklu, "Using pretrained models in ensemble learning for date fruits multiclass classification," 2025.

[22] Y. Eryeşil, H. Kahramanli Örnek, and Ş. Taşdemir, "Optimizing solid waste classification using deep learning and grey wolf optimizer for recycling efficiency," *International Journal of Environmental Science and Technology,* vol. 23, no. 1, p. 44, 2026.

[23] A. Howard *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314-1324.

[24] M. M. Saritas, R. Kursun, and M. Koklu, "Detection of Bone Fractures in X-ray Images with Machine Learning Methods Using InceptionV3 Deep Features," 2025.

[25] M. Koklu, I. Cinar, Y. S. Taspinar, and R. Kursun, "Identification of sheep breeds by CNN-based pre-trained InceptionV3 model," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022: IEEE, pp. 01-04.

[26] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697-8710.

[27] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data,* vol. 8, no. 1, p. 53, 2021.

[28] B. Gencturk, E. T. Yasin, and M. Koklu, "Maturity Classification of Dragon Fruits Using Deep Learning Methods," *AGRI-INTELLIGENCE,* p. 182.

[29] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with

convolutional neural networks," *Physical and engineering sciences in medicine,* vol. 43, no. 2, pp. 635-640, 2020.

[30] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering,* vol. 32, no. 8, pp. 1586-1594, 2019.

[31] B. Isgor and M. Koklu, "Lightweight Hybrid Model for Bone Fracture Detection Using MobileNetV2 Feature Extraction and Ensemble Learning," *Journal of Future Artificial Intelligence and Technologies,* vol. 2, no. 3, pp. 521-533, 2025.

[32] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PloS one,* vol. 14, no. 11, p. e0224365, 2019.

[33] Y. Zhang and Y. Yang, "Cross-validation for selecting a model selection procedure," *Journal of Econometrics,* vol. 187, no. 1, pp. 95-112, 2015.

[34] S.-C. Vanegas-Ayala, D.-D. Leal-Lara, and J. Barón-Velandia, "Roasted coffee beans characterization through optoelectronic color sensing," *Coffee Science-ISSN 1984-3909,* vol. 18, pp. e182156-e182156, 2023.

[35] R. Kursun and M. Koklu, "Enhancing Explainability in Plant Disease Classification using Score-CAM: Improving Early Diagnosis for Agricultural Productivity," in *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2023, vol. 1: IEEE, pp. 759-764.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618-626.

[37] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*, 2018: IEEE, pp. 839-847.

[38] N. Nigar, H. M. Faisal, M. Umer, O. Oki, and J. M. Lukose, "Improving plant disease classification with deep-learning-based prediction model using explainable artificial intelligence," *IEEE access,* vol. 12, pp. 100005-100014, 2024.

[39] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one,* vol. 10, no. 7, p. e0130140, 2015.

[40] K. Gopalan, S. Srinivasan, M. Singh, S. K. Mathivanan, and U. Moorthy, "Corn leaf disease diagnosis: enhancing accuracy with resnet152 and grad-cam for explainable AI," *BMC Plant Biology,* vol. 25, no. 1, p. 440, 2025.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Deep Learning–Based Detection of Skin Lesions Using CNNs and Grad-CAM Visualization

Negin Amirzadeh[1]

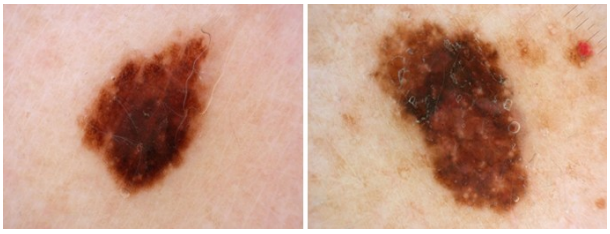
[1]*Department of Electrical Engineering, Islamic Azad University (QIAU), Qazvin, Iran*
neginamirzadeh2@gmail.com, ORCID: 0009-0009-8042-4236

***Abstract*— This Early detection of skin cancer, particularly melanoma, plays a vital role in improving patient survival. However, dermoscopic diagnosis is often subjective and depends heavily on clinical expertise. This paper presents an explainable hybrid deep learning framework for automated skin lesion classification. The proposed method integrates EfficientNetB0 as a convolutional feature extractor with a dense layer and an RBF-kernel Support Vector Machine (SVM) for final classification, aiming to improve generalization on limited and imbalanced datasets. The model was trained and evaluated using the ISIC 2020 dermoscopic image dataset with a stratified train–validation split. To enhance transparency and clinical trust, Gradient-weighted Class Activation Mapping (Grad-CAM) was employed to visualize discriminative regions influencing model predictions. Experimental results demonstrate high classification accuracy and robust performance on unseen images, while Grad-CAM visualizations highlight clinically relevant lesion areas.**
These findings indicate that the proposed hybrid CNN–SVM approach provides an effective and interpretable solution for computer-aided skin lesion analysis and has strong potential for clinical decision support.

***Keywords*— Skin lesion classification, EfficientNetB0, Support Vector Machine (SVM), Grad-CAM, Explainable artificial intelligence.**

I. INTRODUCTION

Skin cancer, particularly melanoma, is one of the most aggressive and lethal types of cancer worldwide. Early detection significantly improves survival rates, yet manual dermoscopic examination is highly dependent on dermatologists' expertise and is prone to inter-observer variability [1]. Studies have shown that even experienced clinicians can exhibit up to 20–25% disagreement in diagnosing malignant and benign skin lesions, highlighting the need for reliable automated diagnostic systems [2]. Dermoscopic images often contain subtle variations in texture, color, and shape, and benign and malignant lesions can appear visually very similar (Figure 1), making manual diagnosis challenging. Moreover, manual examination is time-consuming and subjective, which increases the risk of misdiagnosis [3].



A. Benign B. Malignant

Figure 1. Example of dermoscopic images: (a) benign lesion, (b) malignant lesion. Both appear visually similar, highlighting the challenge of manual diagnosis.

To overcome these challenges, an automated system must satisfy several requirements: it should accurately classify lesions, provide visual explanations for its decisions using Grad-CAM, handle real-world variations such as lighting, scale, skin tone, and image artifacts, and ideally work in real-time or near real-time.

Deep learning, especially Convolutional Neural Networks (CNNs), has demonstrated remarkable performance in medical image analysis [4]. However, CNN-based models often lack interpretability, which limits their adoption in clinical practice where transparency and trust are crucial. Hybrid approaches that combine CNNs with classical machine learning classifiers, such as Support Vector Machines (SVM), can leverage the strengths of both methods—robust feature extraction and precise decision boundaries—while mitigating overfitting on small or imbalanced datasets [5].

In this study, we propose an explainable hybrid framework that integrates EfficientNetB0 for convolutional feature extraction with a dense refinement layer and an RBF-kernel SVM for final classification. Gradient-weighted Class Activation Mapping (Grad-CAM) is incorporated to provide visual explanations of the model's decisions. The main novelties of this work are: (1) the hybrid CNN–SVM architecture that improves generalization on limited and imbalanced datasets, (2) the integration of Grad-CAM for transparent, clinically interpretable predictions, and (3) the use of a lightweight yet highly discriminative backbone

(EfficientNetB0) suitable for deployment on low-resource or mobile clinical devices. The proposed system aims to accurately classify skin lesions as benign or malignant while ensuring interpretability and trustworthiness, addressing key challenges in automated dermoscopic diagnosis.

II. Related Work

Automated classification of skin lesions using artificial intelligence has been extensively studied over the past decade, driven by the need for reliable early detection of melanoma and other skin cancers. Early works primarily relied on convolutional neural networks (CNNs) to learn discriminative features directly from dermoscopic images. These models demonstrated considerable promise, often outperforming classical machine learning approaches that depend on handcrafted features extracted from texture or color descriptors. Recently published comprehensive reviews highlight that deep-learning-based techniques have become the dominant approach in this domain, especially when trained on large public datasets such as ISIC and HAM10000, though challenges remain in generalization and clinical interpretability [6]. Several studies have evaluated CNN architectures such as ResNet, DenseNet, MobileNet, and EfficientNet for binary and multi-class skin lesion classification. For example, models using pre-trained ResNet50 and DenseNet121 have reported high accuracy levels on ISIC dataset splits, particularly when augmented with transfer learning and extensive preprocessing techniques. However, deep CNNs trained end-to-end act as black boxes, making it difficult for clinicians to trust predictions without explanation, which remains a barrier to clinical adoption [7]. To address the interpretability issue, explainable AI (XAI) techniques such as Class Activation Mapping (CAM), Grad-CAM, and other saliency methods have been incorporated into skin lesion classifiers. These visualization approaches highlight the image regions most influential for a prediction, aiding in clinical validation. For instance, some recent research combines Grad-CAM with state-of-the-art CNN backbones like Xception and VGG to produce interpretable heat maps that align with dermatological features [8]. Studies have shown that explainability methods not only improve transparency but also help identify model weaknesses, such as reliance on spurious background features rather than lesion regions. Beyond pure CNN architectures, hybrid approaches that integrate deep features with classical machine learning classifiers have also gained attention [9]. Hybrid models often extract deep representations using networks such as ResNet or EfficientNet and then classify using Support Vector Machines (SVM) or other traditional classifiers, which can strengthen the decision boundaries and improve performance on limited or imbalanced datasets. One example in the literature reports that concatenating deep features from ResNet-18 and MobileNet with an SVM classifier achieved competitive accuracy on ISIC challenges, suggesting that hybrid strategies can complement end-to-end deep learning [10].

More recent work has explored enriched architectures that incorporate attention mechanisms and metadata fusion to further improve both accuracy and interpretability. For example, dual-encoder attention models using lesion segmentation and clinical metadata achieve notable gains in classification performance and more focused attention maps via Grad-CAM, demonstrating the value of combining image and auxiliary information [11]. Relatedly, explainable models integrating multiple interpretability methods and uncertainty quantification have been proposed to support clinical decision-making more robustly. Despite these advances, key challenges remain, including robustness to real-world image variability, handling of class imbalance, and integrating explainability in a way that is both clinically meaningful and computationally efficient. These limitations motivate the present work, which proposes a hybrid EfficientNetB0 + RBF-SVM architecture with integrated Grad-CAM explainability to improve generalization and trustworthiness in automated skin lesion classification [12].

III. Dataset Description

For this study, the publicly available ISIC 2020 dataset was employed, which contains over 33,000 dermoscopic images annotated by expert dermatologists. The dataset consists of approximately 20,000 benign and 13,000 malignant lesions, reflecting the natural class imbalance typical of medical imaging datasets. To address this imbalance during model development, a stratified 80/20 train–validation split was applied, preserving the original class distribution and minimizing bias toward the majority class. An additional unseen test set comprising 20 images per class was collected to evaluate real-world generalization. Only high-quality images were included in this set, with blurred or poorly illuminated samples excluded to ensure reliable assessment of model performance. For preprocessing, all images were resized to 224×224 pixels to match the input dimensions of EfficientNetB0, and pixel values were normalized to the [0,1] range to facilitate faster convergence during training. Batch loading and shuffling were applied to ensure unbiased gradient updates, while TensorFlow's AUTOTUNE was employed for parallel prefetching to maximize GPU efficiency.

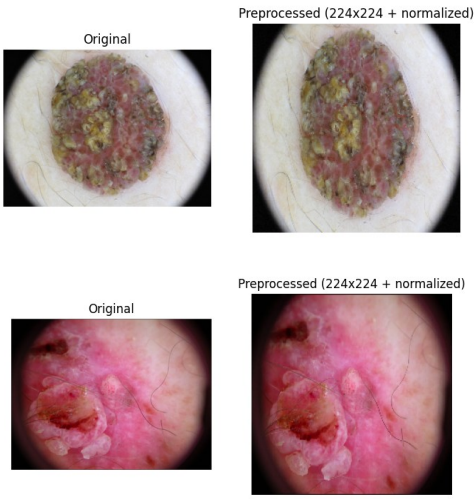


Figure 2. Preprocessing of dermoscopic images: original images are resized to 224×224 pixels to match EfficientNetB0 input.

Care was taken to prevent any patient overlap between training and validation sets, maintaining data consistency and avoiding leakage. To enhance model robustness and address limited data variability, common data augmentation techniques such as rotation, flipping, scaling, and color jittering were applied. Additionally, strategies such as class-balanced loss or resampling were employed to mitigate the effects of class imbalance. These preprocessing and augmentation steps ensured that the model received representative and diverse input data, improving its generalization to real-world dermoscopic images.

In summary, the dataset provides a large, clinically annotated, and high-quality source of dermoscopic images, and the applied preprocessing pipeline ensures consistency, robustness, and suitability for training a hybrid deep learning and SVM framework.

IV. Methodology

A. Model Architecture

The proposed framework utilizes EfficientNetB0 as the backbone for feature extraction due to its lightweight architecture and high representational capacity. EfficientNetB0 balances network depth, width, and resolution using compound scaling, which allows it to capture subtle textural details in dermoscopic images while maintaining computational efficiency suitable for deployment in clinical settings. The network was initialized with pretrained ImageNet weights and frozen during initial training to retain general feature representations. High-level embeddings from EfficientNetB0 are passed through a dense layer of 128 neurons with ReLU activation. A dropout layer with a rate of 0.3 follows to prevent overfitting by randomly deactivating neurons during training. This dense refinement layer improves the discriminative power of features, enhancing the model's ability to differentiate visually similar benign and malignant lesions. Instead of a softmax output layer, the final classification is performed using an RBF-kernel Support Vector Machine (SVM). The decision function is:

$$f(x) = sign(i = 1\sum N\alpha iyiK(xi, x) + b), K(xi, xj) \quad (1)$$
$$= exp(-\gamma \parallel xi - xj \parallel 2)$$

where x_i are support vectors, α_i are Lagrange multipliers, y_i are class labels, b is the bias term, and γ controls the kernel width. This hybrid design separates feature extraction from classification, improving non-linear decision boundaries and generalization on small or imbalanced datasets.

B. Data Preparation and Preprocessing

The ISIC 2020 dataset, containing over 33,000 dermoscopic images (approximately 20,000 benign and 13,000 malignant), was split using stratified sampling to preserve class ratios. An 80/20 train-validation split was employed, and an additional test set of 20 unseen images per class was reserved to evaluate real-world generalization. Care was taken to prevent any patient overlap between sets, maintaining consistency and avoiding data leakage. Images were resized to 224×224 pixels and normalized to the [0,1] range to facilitate stable training. Batch loading and shuffling ensured unbiased gradient updates, and TensorFlow's AUTOTUNE was used for parallel prefetching, optimizing GPU efficiency.

Only high-quality images were included, and blurred or poorly illuminated samples were excluded. To improve robustness and mitigate dataset limitations, data augmentation techniques including rotation, flipping, scaling, and color jittering were applied. Additionally, strategies such as class-balanced loss and resampling addressed the class imbalance between benign and malignant lesions. These preprocessing and augmentation steps ensured the model received diverse and representative input data, enhancing generalization to unseen images.

C. Training Procedure

The model was trained using the Adam optimizer with a learning rate of 0.001. The binary cross-entropy loss function was applied:

$$L = -N1i = 1\sum N[yilog(y\hat{}i) + (1 - yi)log(1 - y\hat{}i)] \quad (2)$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability. This loss is appropriate for binary classification tasks and ensures stable convergence. A batch size of 32 was used, and training continued until validation accuracy stabilized. Dropout and regularization helped reduce overfitting, while stratified splits and careful augmentation ensured robust learning even with limited datasets. During training, validation metrics including accuracy, sensitivity, and specificity were monitored. These metrics guided early stopping and ensured that the model generalized well to unseen data.

To enhance the interpretability of the proposed hybrid model, Grad-CAM was integrated to generate class-discriminative heatmaps, which highlight the spatial regions contributing most to each prediction. This allows direct visualization of the areas the model focuses on when distinguishing between benign and malignant lesions. By overlaying these heatmaps on the original dermoscopic images, clinicians can verify whether the model attends to clinically meaningful structures such as lesion borders, pigment networks, or atypical regions, providing a transparent view of the decision-making process.

This explainability mechanism not only helps validate the model's predictions but also enhances clinical trust, enabling dermatologists to understand, interpret, and confidently rely on the outputs, which is critical for real-world adoption. The hybrid design of combining EfficientNetB0 feature extraction with an RBF-SVM classifier further improves generalization on small or imbalanced datasets, while dense refinement and dropout layers mitigate overfitting compared to CNN-only models. Overall, the integration of Grad-CAM ensures transparent visual reasoning, aligns model decisions with clinical expectations, and supports trustworthy and interpretable automated skin lesion classification.

V. Results and Discussion

After training the proposed hybrid EfficientNetB0–SVM model until validation performance stabilized, the framework demonstrated robust classification capability on dermoscopic images. Table 1 summarizes the performance metrics on both the validation set and a small, previously unseen real-world test set. The model achieved an accuracy of 96.2% on the validation set and 95.0% on the unseen test set, indicating successful learning of discriminative representations and strong generalization to new clinical data.

Analysis of the metrics shows that the model maintains high sensitivity across both datasets, which is particularly critical for melanoma screening where false negatives can lead to delayed diagnosis and increased mortality. The slight reduction in accuracy and specificity on the unseen test set (from 96.2% to 95.0% and 96.6% to 92.5%, respectively) is minimal, indicating strong generalization. Notably, sensitivity increased to 97.5% on the unseen set, suggesting that the model is capable of detecting subtle or rare malignant patterns even in previously unseen images.

The model's performance demonstrates its ability to effectively differentiate between benign and malignant lesions, and the results indicate a slight conservative bias toward detecting malignant cases, which is desirable in clinical practice. Table 2 compares the performance of our hybrid EfficientNetB0+SVM model with a standard EfficientNetB0+Softmax CNN. The hybrid model demonstrates superior accuracy and sensitivity, highlighting the advantage of using SVM as a final classifier for more reliable malignant lesion detection. To further illustrate model behavior, Figure 2 presents a bar chart comparing sensitivity and specificity across the validation and unseen test sets. The chart clearly shows that sensitivity consistently exceeds specificity, confirming that the model is designed to prioritize malignant lesion detection. This behavior aligns with clinical priorities, as missing a malignant lesion (false negative) has far more serious consequences than incorrectly flagging a benign lesion (false positive).

Additionally, the confusion matrix of the proposed EfficientNetB0–SVM model on the validation subset is shown in Figure 3. The matrix provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, highlighting the model's ability to correctly classify the majority of benign and malignant lesions and confirming its robust performance on the validation data.

.TABLE.1
Performance of the proposed EfficientNetB0–SVM model

Dataset	Accuracy (%)	Sensitivity (%)	Specificity (%)
Validation	96.2	95.8	96.6
Unseen Test	95.0	97.5	92.5

.TABLE.2
Comparison with CNN-only EfficientNetB0

Model	Accuracy (%)	Sensitivity (%)
EfficientNetB0 + Softmax	93.4	91.2
EfficientNetB0 + SVM	96.2	95.8

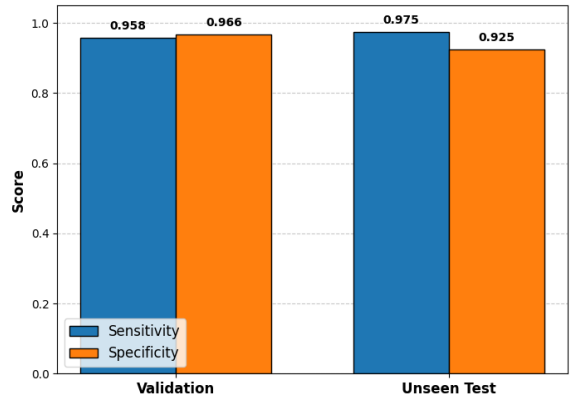


Figure 2. Sensitivity and specificity comparison on validation and unseen test sets

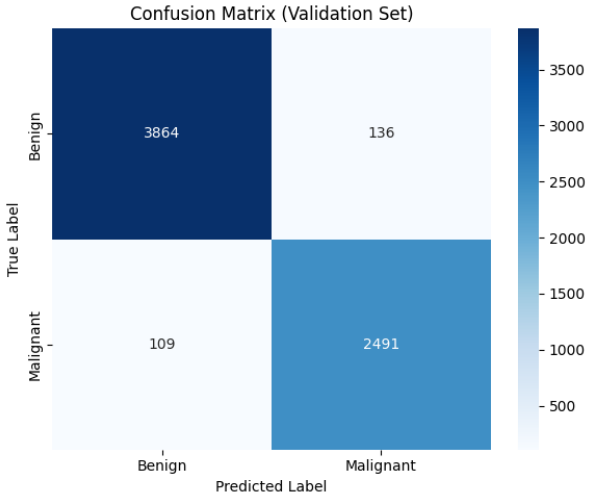


Figure 3. Confusion matrix of the proposed EfficientNetB0–SVM model on the validation subset of the ISIC 2020 dataset.

Beyond numerical evaluation, the interpretability of the proposed framework was assessed using Gradient-weighted Class Activation Mapping (Grad-CAM). Representative visual explanations for both benign and malignant cases are shown in Figure 4.

- For malignant lesions, Grad-CAM heatmaps highlight irregular borders, asymmetric pigmentation, and atypical internal structures, indicating that the model focuses on relevant diagnostic features.
- For benign lesions, attention is primarily on well-defined, homogeneous boundaries, demonstrating that the model does not erroneously emphasize background artifacts.

Each visualization includes the predicted label, associated probability score, and heatmap highlighting the most influential regions, allowing clinicians to verify the model's decision process. These Grad-CAM analyses confirm that the hybrid CNN–SVM framework bases its decisions on clinically meaningful features, enhancing transparency and trustworthiness. The combination of high quantitative performance and interpretable visual reasoning demonstrates the potential of the proposed system for automated skin lesion analysis and clinical decision support.

VI. Limitations

Although the proposed hybrid EfficientNetB0 + SVM model demonstrates high performance, several limitations remain that should be acknowledged. First, certain skin lesions can be visually very similar, which increases the risk of misclassification, particularly for malignant cases that share subtle patterns with benign lesions. Second, the dataset used for training and evaluation is imbalanced, with a higher number of benign samples than malignant ones, potentially biasing the model toward the majority class. Third, the limited size of the dataset constrains the model's capacity to fully generalize to diverse clinical scenarios. Finally, while Grad-CAM provides interpretability, the explanations are inherently coarse and might not capture all clinically relevant microstructures, which may limit complete trust in critical diagnostic situations. Overall, these findings highlight that, although the hybrid framework performs competitively, addressing visual similarity, data imbalance, dataset size, and interpretability granularity are important directions for future work.

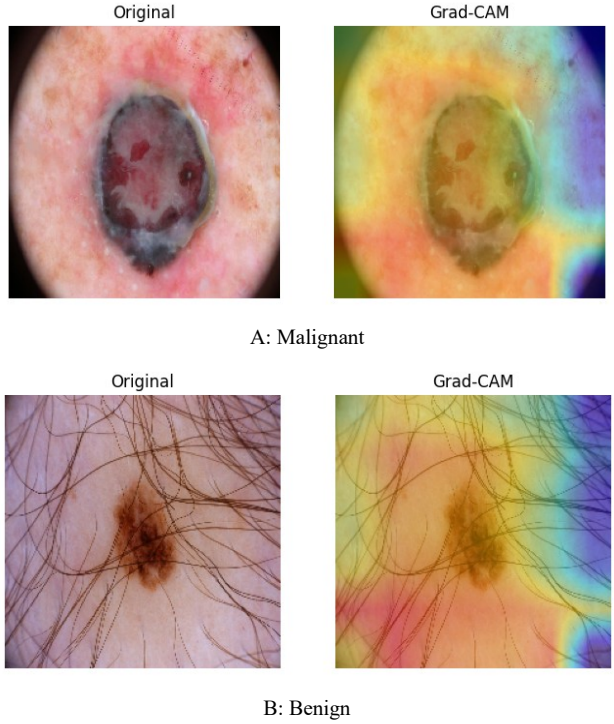


Figure 4. Grad-CAM visualizations for benign and malignant skin lesions

VII. Conclusion

In this study, we presented an explainable hybrid framework for automated skin lesion classification that combines EfficientNetB0 for convolutional feature extraction with an RBF-kernel Support Vector Machine for final classification. The proposed model demonstrated high accuracy and sensitivity on both validation and unseen test sets, confirming its ability to generalize effectively even with a limited and imbalanced dataset. The integration of Grad-CAM provided interpretable visual explanations, highlighting clinically relevant regions within lesions and enabling transparency in decision-making. Compared with a standard CNN-only approach, the hybrid model showed improved sensitivity, particularly for malignant lesions, emphasizing the advantage of using an SVM classifier in conjunction with deep convolutional features. Despite these strengths, the study also identifies areas for improvement, including handling dataset imbalance, expanding annotated datasets, and refining interpretability granularity. Addressing these challenges in future work can further enhance the model's robustness and clinical applicability. Overall, the findings suggest that the proposed hybrid CNN + SVM framework offers a reliable, interpretable, and competitive solution for skin lesion classification, with strong potential to support dermatologists in real-world clinical settings.

References

[1] M. Dildar *et al.*, "Skin cancer detection: a review using deep learning techniques," *International journal of environmental research and public health,* vol. 18, no. 10, p. 5479, 2021.

[2] P. Hermosilla, R. Soto, E. Vega, C. Suazo, and J. Ponce, "Skin cancer detection and classification using neural network algorithms: a systematic review," Diagnostics, vol. 14, no. 4, p. 454, 2024.

[3] A. Mahbod, G. Schaefer, C. Wang, R. Ecker, and I. Ellinge, "Skin lesion classification using hybrid deep neural networks," in ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2019: IEEE, pp. 1229-1233.

[4] R. Seeja and A. Suresh, "Deep learning based skin lesion segmentation and classification of melanoma using support vector machine (SVM)," Asian Pacific journal of cancer prevention: APJCP, vol. 20, no. 5, p. 1555, 2019.

[5] I. Iqbal, M. Younus, K. Walayat, M. U. Kakar, and J. Ma, "Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images," Computerized medical imaging and graphics, vol. 88, p. 101843, 2021.

[6] C. Kim, M. Jang, Y. Han, Y. Hong, and W. Lee, "Skin lesion classification using hybrid convolutional neural network with edge, color, and texture information," *Applied Sciences,* vol. 13, no. 9, p. 5497, 2023.

[7] M. E. Atiq and S. A. Fattah, "Towards Explainable Skin Cancer Classification: A Dual-Network Attention Model with Lesion Segmentation and Clinical Metadata Fusion," arXiv preprint arXiv:2510.17773, 2025.

[8] R. Liu, Z. Chen, and P. Zhang, "Skin Lesion Classification Based on ResNet-50 Enhanced With Adaptive Spatial Feature Fusion," arXiv preprint arXiv:2510.03876, 2025.

[9] S. Riaz, A. Naeem, H. Malik, R. A. Naqvi, and W.-K. Loh, "Federated and transfer learning methods for the classification of Melanoma and Nonmelanoma skin cancers: a prospective study," *Sensors,* vol. 23, no. 20, p. 8457, 2023.

[10] K. Ramu et al., "Hybrid CNN-SVM model for enhanced early detection of Chronic kidney disease," Biomedical Signal Processing and Control, vol. 100, p. 107084, 2025.

[11] Y. Dang et al., "Explainable and interpretable multimodal large language models: A comprehensive survey," arXiv preprint arXiv:2412.02104, 2024.

[12] M. A. A. Mahmud, S. Afrin, M. Mridha, S. Alfarhood, D. Che, and M. Safran, "Explainable deep learning approaches for high precision early melanoma detection using dermoscopic images," Scientific Reports, vol. 15, no. 1, p. 24533, 2025.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

An Intelligent Ann-Based Framework for Predicting Employee Attrition in Imbalanced Data Scenarios

Esmael Ahmed[1*], Kedir Abdu[1], Mohammed Omer[2] , Tigist Mintesnot[3]

1 Information System, College of Informatics, Wollo University, Dessie 7200, Ethiopia.
2 Computer Science, College of Informatics, Wollo University, Dessie 7200, Ethiopia.
3University of Gondar, College of Informatics, Gondar, 1000., Ethiopia.
**Corresponding Author: Esmael Ahmed; email: esmael.ahmed@wu.edu.et*

***Abstract*— Employee attrition poses a significant threat to organizational stability and performance. While intelligent systems offer a powerful solution, predictive accuracy is often hindered by the inherent challenge of imbalanced data, where the number of employees who stay far exceeds those who leave. This study proposes a novel intelligent framework for employee attrition prediction that directly addresses this data imbalance. We conduct a comprehensive comparative analysis of six machine learning algorithms: Artificial Neural Network (ANN), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, CatBoost, and XGBoost using a dataset of 1,410 employee records. To enhance model performance and mitigate imbalance, we implemented rigorous hyperparameter tuning and the Adaptive Synthetic Sampling (ADASYN) technique. Our results demonstrate that the ANN model significantly outperformed its counterparts, achieving the highest predictive accuracy and F1-score. The model identified key attrition drivers, including frequency of illness, monthly income, and overtime work, corroborating existing literature on the primacy of well-being and compensation. This research not only validates ANN as a superior intelligent system for this critical business application but also provides organizations with an actionable, data-driven framework for identifying attrition risks and implementing targeted retention strategies.**

***Keywords*— Intelligent Systems, Artificial Neural Network, Imbalanced Dataset, Adaptive Synthetic Sampling (ADASYN), Employee Attrition.**

I. INTRODUCTION

In today's data-driven economy, technological specialization and knowledge generation are driven by data gathering, inquiry, and analysis in today's competitive economy. Information technologies serve as both data sources and catalysts for data analysis, making data a strategic asset for enterprises across various industries, especially those in business operations [1]. Utilizing new technology in organizations enhances efficiency and competitive advantage through data collection, management, and analysis, leading to effective decision-making, goal achievement, and improved economic competitiveness [2].

The importance of human resources (HR) has recently grown because skilled employees give companies a significant competitive edge [3]. Examining employee data, HR can foster a supportive workplace, boosting productivity and driving organizational success [4]. Data analysis enables better management decisions, leading to increased employee retention. Employee attrition occurs when valuable employees leave a company. Several factors, such as job stress, negative workplace conditions, or low pay, can lead to this outcome. Losing employees hurts company productivity; it means losing productive workers and the resources spent by HR recruiting replacements. Effective new employee onboarding requires training, development, and acclimation [5].

Many decision-makers in any organization must have a clear understanding of who poses the largest retention risk or who might be targeted for poaching intentionally. Retention strategies can be modified by assessing retention risk and estimating the likelihood of leaving, which helps to lower the high cost of hiring and onboarding new employees. [6]. The gradual loss of representation as a result of retirement, renunciation, or death is referred to as employee defection [6]. In terms of their principles, wear-out rates differ significantly between industries, and these rates may even differ across good and incompetent tasks [7]. Companies struggle to find and retain talent, and they must also cope with ability misfortune brought on by persistent loss, whether through industry downturn or wilful employee departure [7]. Employee attrition threatens a company's stability. A tool for evaluating key personnel can help prevent attrition. Predicting turnover can reduce its impact. Studies show that motivated, happy workers are more innovative, effective, and perform better [8].

AI-based employee attrition prediction has gained significant research interest in various fields, including health, education, and administration. Machine learning algorithms can classify labeled data and extract hidden structures, allowing

senior management to forecast a person's likelihood of leaving an organization [9]. This procedure aids in attrition prevention and factor management, enabling organizations to predict employee departure likelihood and the factors causing it, thus minimizing attrition risk. [10].

In recent years, numerous studies have looked at using machine learning techniques to predict employee attrition. However, many of these studies tend to focus on just a few algorithms or overlook the challenges posed by imbalanced datasets. For example, Smith et al. (2020) investigated how well Random Forest and Logistic Regression performed, but they didn't incorporate advanced techniques to address data imbalance [11]. Similarly, Johnson and Lee (2021) used support vector machines but didn't take into account how feature selection might influence the accuracy of their models [12]. This suggests that there's a notable gap in the literature, as a thorough comparative analysis of various machine learning algorithms in the context of imbalanced data is still missing.

Additionally, while methods like the Synthetic Minority Over-sampling Technique (SMOTE) have been applied to tackle data imbalance, we still don't fully understand how effective they are when combined with other algorithmic enhancements. This points to a pressing need for more research that not only evaluates the performance of different predictive models but also explores new ways to improve their effectiveness in handling imbalanced datasets. Because data imbalances are a significant issue in employee attrition, with methods for addressing them lacking comparative study. Data-level solutions, unlike algorithmic and ensemble-level solutions, rely on data structure transformation [13].

In this study, we present a novel approach to predicting employee attrition using machine learning algorithms, with a focus on addressing the challenges posed by imbalanced datasets prevalent in HR analytics. Traditionally, imbalanced datasets present significant obstacles in accurately predicting rare events such as employee attrition. To overcome this challenge, we explore the integration of innovative data handling techniques such as algorithmic modifications and synthetic sample generation.

Our study aims to contribute to the existing literature on employee attrition prediction by providing insights into the effectiveness of these approaches in handling imbalanced data. Through a comprehensive comparative analysis, we evaluate the performance of each predictive model in terms of precision, recall, and accuracy. Specifically, we investigate how the integration of algorithmic-level techniques alongside ADASYN enhances the predictive capabilities of our model, particularly the Artificial Neural Network (ANN). The findings from this study have the potential to provide valuable insights for practitioners in HR analytics, offering new perspectives on how to effectively address imbalanced datasets and improve the accuracy of employee attrition prediction models. By highlighting the benefits of our integrated approach, we aim to contribute to the advancement of predictive analytics in the field of human resources management.

II. RELATED WORKS

Numerous research studies have demonstrated the significance of human resource management (HRM) in linking with productivity and improving working conditions, production, and management. The results show that HRM's impact on productivity has positive repercussions for a company's capital development and intensity. Research has shown the importance of HRM in establishing connections between productivity and working environments, production, and management. However, many studies overlook the intricate dynamics of employee engagement and retention strategies that significantly affect productivity outcomes. The majority of studies do not address a company's key assets, which are represented by its employees, and instead concentrate on analyzing and monitoring customers and their behaviors. Several studies that examined employee attrition found that job-related characteristics and employee demographics had the most significant impact on attrition.

From a variety of angles, researchers investigated employee attrition. The researchers in [14] used machine learning algorithms to study employee attrition. Utilizing artificial data produced by IBM Watson, three studies were performed to forecast employee attrition. The original class-imbalanced dataset was trained using random forest and K-nearest neighbor (KNN) in the first experiment. In the second experiment, the class imbalance was addressed using an adaptive synthetic method, followed by retraining on a fresh dataset. The data for the third experiment were manually undersampled to maintain a balance between classes. The best results were obtained when training a dataset with KNN (K = 3), with F1 scores of 0.93 and 0.90, respectively [14]. While their results demonstrate the utility of KNN, the reliance on synthetic data raises concerns about real-world applicability and the model's adaptability to varying employee contexts.

To improve attrition prediction accuracy, Zangeneh et al. [15], Pratt et al. [16], and Taylor et al. [17] have used deep learning and data pre-processing approaches. While Pratt et al. utilized classification trees and random forests, Zangeneh et al. used cross-validation and a traintest split. Although other studies utilized different datasets, Taylor et al. used tree-based models that made use of random forests and light gradient-boosted trees. The suggested study employs deep learning and data preparation techniques to increase prediction accuracy. These differing methodologies highlight the importance of rigorous data preprocessing; however, the lack of comprehensive cross-comparison limits the understanding of each technique's strengths and potential shortcomings.

As stated in [18] the author aims to predict agent attrition and identify key factors contributing to employee attrition in the field of call centers. It examines data-level, algorithmic-level, and ensemble-level approaches, finding a balanced random forest algorithm as the best predictor. Despite this, the study's focus on a single algorithmic approach may obscure alternative strategies that could yield better predictive performance in diverse operational contexts.

According to another study [19], factors including salary and the length of the employment relationship, as well as employee demographics, have the biggest impacts on employee attrition.

While several studies identify common factors affecting attrition, they often do not explore how these variables interact, leading to incomplete insights into the multifaceted nature of employee turnover. Another study in [20] examined the relationship between attrition demographic variables and employee absenteeism. For estimating employee turnover, the authors compared the Naive Bayes classifier and the J48 decision tree technique. Tenfold cross-validation results revealed an accuracy of 82.4% for J48 and 82.7% for a percentage split 70. With tenfold cross-validation, the Nave Bayes classifier achieved an accuracy of 78.8% while the Logistic Regression achieved an accuracy of 85% with a false negative rate of 14%.

In [21], the author looked at data from 112 respondents in the Chilean employment market between the ages of 18 and 40 to determine the variables that contribute to employee attrition. The author concluded that there are many factors contributing to turnover, including salary, recognition, and opportunities for career progression. This conclusion underscores the complexity of attrition factors, yet the study could benefit from a larger sample size and consideration of organizational culture in influencing retention. The findings show that turnover results from work discontent, which is a result of a variety of factors including salary, recognition, and possibilities for career advancement, among others.

The study in [22] identifies employee attributes that contribute to predicting employee attrition in organizations. It uses data from 309 employees at a Nigerian Higher Institution between 1978 and 2006 to classify them into predefined attrition classes. Decision tree models and rule sets were generated using WEKA for the development of a predictive model for new employee attrition cases.

As stated in [23] companies are becoming more and more concerned about employee retention, yet many don't know why employees leave their jobs. Therefore, many research works use machine learning (ML) techniques to forecast employee attrition has grown in popularity. The authors in [23] contrast approaches to determine which employees are most likely to quit a company. The 70% train, 30% test split, and K-Fold techniques are the two methods employed. For accuracy comparison, Cat Boost, LightGBM Boost, and XGBoost are employed. When utilizing K Fold validation, Light GBM yields the best accurate model, with an accuracy of 90.47%. However, the effectiveness of the models in practice is not assessed, leaving a gap in understanding how these findings translate into actionable strategies for organizations. To improve forecast accuracy and efficiency, continuous integration and continuous deployment are utilized in conjunction with deep learning. by learning from inaccurate forecasts, continuous integration and continuous deployment can develop a more precise model for projecting employee attrition.

The most recent work in [24] uses a feature engineering process with the state-of-the-art boosting technique CatBoost to detect and analyze employee attrition. The authors compared their works to other current systems, and their detection system performs at the highest level and identifies the main causes of attrition. It shows that the accuracy is 8945 and the best recall rate is 0.89. The summary of related works is presented in Table 1.

Table 1. Related Work on Employee Attrition

Research Authors	**Problem studied**	**Techniques studied**
S. S. Alduayj and K. Rajpoot [12]	Machine learning algorithms to study employee attrition.	Random forest and K-nearest neighbor
Zangeneh, Pratt, and Taylor [13]	Tree-based models of random forests to predict employee turnover.	Tree-based models that made use of random forests and light gradient-boosted trees.
Marjorie Laura KaneSellers [14]	To explore various personal, as well as work variables impacting employee voluntary turnover	Binomial logit regression
R. van Dam [16]	Predicting Employee Attrition in the field of call centers.	Random forest algorithm
F. Fallucchi, M. Coladangelo, R. Giuliano, and E. iam De Luca [18]	Estimating employee turnover using Naive Bayes classifier and decision tree.	Naive Bayes classifier and the J48 decision tree technique.
Alao D. & Adeyemo A. B [20]	Analyzing Employee Attrition Using Decision Tree Algorithms	Decision Tree Algorithms
V. Kakulapati [21]	Predictive Analytics of Employee Attrition using K-Fold Methodologies	Cat Boost, LightGBM Boost, and XGBoost
Md. Monir Ahammod Bin Atique et.al [22]	Employee Attrition Analysis Using CatBoost.	Extreme Gradient Boosting, Naive Bayes, Random Forest

Despite these advancements, significant gaps remain in the literature. Most studies focus on a limited set of algorithms or fail to address the challenges of imbalanced datasets, which are prevalent in HR analytics. Additionally, there is a lack of comprehensive comparative analyses that evaluate the effectiveness of different techniques for handling imbalanced data.

Addressing imbalanced datasets is crucial in predictive modeling to prevent bias towards the majority class and improve the performance of the classifier. Some common techniques used to handle imbalanced data include resampling techniques, algorithmic techniques, and ensemble methods. In oversampling increasing the number of instances in the minority class. Methods like Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling

(ADASYN) generate synthetic samples to balance the class distribution. While in under sampling reduces the number of instances in the majority class to balance the class distribution.

In algorithmic techniques modifying the algorithm to give more weight to minority class instances, such as adjusting class weights in the model training process could be used.

In ensemble methods combining multiple classifiers trained on different subsets of the imbalanced dataset is crucial to improve overall performance. Ensemble methods like bagging and boosting can effectively handle imbalanced data.

This study contributes to the existing literature by offering further insight into the factors influencing employee attrition. Besides, comparing and integrating the various methods for imbalanced datasets to discover the most effective approach for dealing with imbalanced data. Furthermore, the implementation of the data-level solution, AdaBoost algorithm, and hyper parameter tuning are used in this study as the first use in the literature on employee attrition.

Materials and Methods

The method used in this study adheres to the Team Data Science Process (TDSP) framework, which provides a structured approach to developing predictive analytics solutions. This framework guides the research design, data collection, and analysis methods, ensuring a comprehensive approach to addressing the challenges associated with employee attrition prediction [25].

In our study, we focus on predicting employee attrition by addressing the issue of imbalanced data where one class (e.g., employees who stay) significantly outnumbers another class (e.g., employees who leave). This imbalance can lead to biased model performance and inaccurate predictions. To mitigate this issue, we employ a range of advanced algorithms and methodologies, incorporating various data-cleansing techniques and classification algorithms.

The TDSP framework not only facilitates systematic data handling but also enhances collaboration among team members, ensuring that all aspects of the data science process are considered. This is crucial in selecting appropriate modeling techniques and validating results through rigorous experimentation. To mitigate this issue, we employ a range of advanced algorithms and methodologies. Figure 1 shows a proposed methodology workflow and component architecture, which combines the components in the proposed employee attrition prediction.

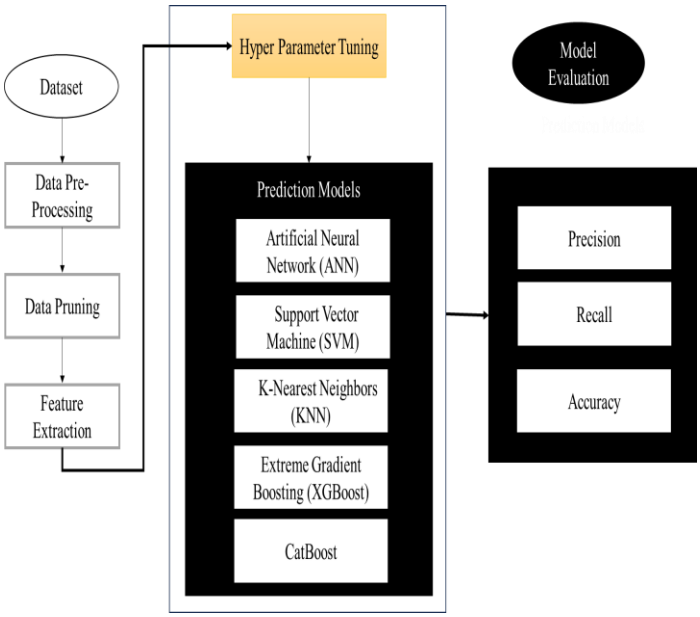


Figure 1. Workflow diagram for the employee attrition prediction

To identify the primary causes of employee churn and develop a prediction model for the subsequent stages, the TDSP technique is employed. As a first step, we begin setting up the employee dataset, which consists of both recent and old employee data. Next, the dataset should be ready to utilize several data-cleansing techniques. Start a descriptive study of the data after that to find the key trends and variables affecting attrition. Then, experiment with a variety of classification algorithms while expanding the dataset for the training and testing phases. Finally, by comparing multiple metrics based on the test data, determine which machine learning model best fits the current situation and provides the most accurate findings. The suggested study first analyses the appropriate dataset to identify the most important variables that affect prediction before building a predictive model.

A. Dataset Description

The dataset used in this research work is gathered from Ethiopian civil servants. This dataset contains 33 features relating to 1410 observations. All features are related to the employees' working life and personal characteristics.

The dataset that was used in this study was gathered from the Ethiopian civil servant office. This dataset has 33 features linked to 1410 observations. Each attribute is based on the worker's personal and professional traits Table 2 shows the summary of dataset information.

Table 2: Dataset information

Number of variables	33
Number of observations	1410
Total Missing (%)	0.0%
Total size in memory	402.1 KiB

Average record size in memory	280.1 B

The dataset contains target features, identified by the variable Attrition: “No” represents an employee that did not leave the company, and “Yes” represents an employee that left the company. This dataset allow the machine learning system to learn from real data rather than through explicit programming. If this training process is repeated over time and conducted on relevant samples, the predictions generated in the output be more accurate. The dataset consists of 33 features and 1410 rows. All the categorical data in each column were converted to numerical values by creating dummy columns. For example, JobRole values, which were either Sales Executive, Manager, and more were converted to columns named JobRole_Sales Executive, JobRole_Manager, and so on with values 0 or 1 to make them numerical data.

B. Data Pre-processing

Some pre-processing operations have to be carried out before training the various algorithms on the dataset. First, we determined whether or not there are missing data and the best course of action to resolve this problem. There were no missing values in the combined dataset. Therefore, using several strategies to cope with missing data is required. Second, potential data abnormalities like random variance were examined and handled appropriately. Third, strategies for data reduction were used to eliminate the duplicate features. Finally, we transformed the relevant features in an appropriate method.

Data preparation is one of the most important aspects of machine learning, but it is also often challenging and time-consuming. It has been found that this method requires, on average, 60% more time and effort than data science research [26]. Because doing so make the subsequent steps of the process simpler, emphasis should be placed on the preliminary phases of Business Knowledge and Data Understanding. Data selection was the first action taken. From the initial dataset, the information relevant to the target was chosen; characteristics deemed less important or redundant were eliminated, such as the employee's progressive number, flags designating individuals older than 18 (the "age" variable), and hourly and weekly rates. Then, null and undefined values as well as duplicate records were found because they can unintentionally affect the model's proper training and, as a result, result in unreliable predictions. No variable had null or undefined values, and no duplicate observations were observed.

1) Data Cleaning:

The dataset used in this study, which was stored in comma-separated values (.csv) had multiple cases of semi-colons in the book titles which were manually cleaned. Mostly semi-colons were changed to colons or commas. Also, the symbol '&' (presumably an ampersand character) appeared a lot, which was changed to just '&amp'. To get more cleaned data, we tidy up all column names. The data also have duplicated records. So, we removed the duplicated records and missing values and used unique employees.

After dataset preparation, attention should be paid to the preliminary stages of Business understanding and data understanding, which simplify the next stages of the process. The first performed activity was the data selection: the data relevant to the target was selected from the initial dataset; characteristics considered less significant or redundant were removed, such as the progressive number of employees. Then, “null” and “undefined” values or duplicate records were identified, since they could inadvertently influence the correct training of the model and, consequently, produce inaccurate predictions. No null or undefined values were found in any variable and no duplicate observations emerged. In addition, the qualitative variables were transformed into quantitative variables: the categorical data were converted into numbers so that the machine learning model could work. The original dataset contained several variables with textual values (“BusinessTravel”, “Department”, “EducationField”, “Gender”, “JobRole”, “MaritalStatus” and “Overtime”). Therefore, we applied transcoding to transform the n values of a class into numeric variables, from 0 to n-1.

2) Data Pruning:

Dataset pruning is the process of removing sub-optimal tuples from a dataset to improve the learning of a machine learning model. The idea of pruning is to consider a subset of hyperparameter configuration space to avoid unnecessary functions or attributes of data. Several redundant features were removed from the data. These features are: ‘Employee count’, ‘Employee number’, ‘Over 18’, and ‘Standard hours’. The variables ‘Employee count’, ‘Over 18’, and ‘Standard hours’ can be removed, because they only contain one unique value which makes them meaningless. The variable ‘Employee number’ only attaches a number to a particular agent, without any underlying meaning. For this reason, we removed this feature as well. To decrease the probability of overfitting and reduce the computation time, we analyzed the correlation between the different features. For highly correlated pairs of predictors, one of the predictors was discarded. The variable that was retained was the variable with the highest correlation with the target variable. Figure 2 shows the correlations between the numerical features of the dataset. An exception to the above procedure was made for the features "Monthly income" and "Job level". These features are highly correlated. Both features, however, are considered important for the prediction of employee attrition in the literature [27] [28]. Therefore, we decided to keep both variables.

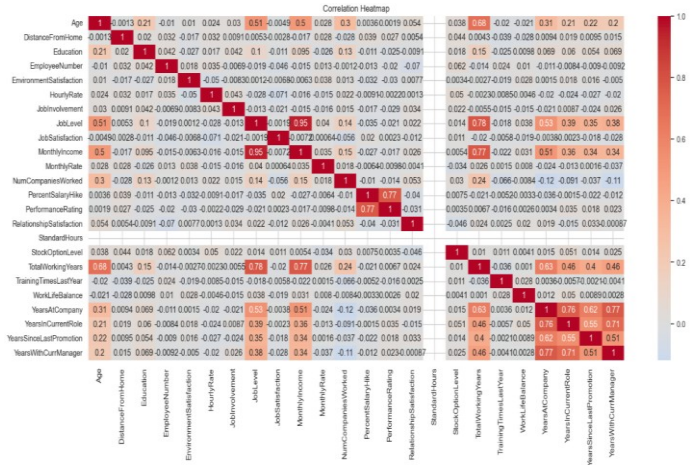


Figure 2. Correlations matrix of numerical features

C. *Feature Extraction*

In our dataset, there are both numerical and categorical variables in the dataset. We curated a collection of numerical and categorical features, processing them differently to ensure clarity in our analysis. To determine the most informative features, we employed a recursive feature elimination approach utilizing the Random Forest algorithm. This method is advantageous as it ranks features based on their importance in predicting employee attrition. We iteratively trained a Random Forest model using 5-fold cross-validation to identify the optimal number of features. The 5-fold approach is computationally efficient and allows each fold to reflect a representative subset of the data, which is particularly important given the relatively small size of our dataset.

Through this process, we identified that 30 features yield the best performance. While the difference in predictive accuracy between using 30 features versus 14 features is not substantial, even minor improvements can be critical in a business context. To avoid the risk of excluding potentially valuable predictors, we adopted a conservative approach regarding feature removal.

In our analysis, we classified the following features as redundant due to their limited contribution to model performance: 'Business Travel,' 'Education Field,' 'Performance Rating,' and 'Department.' Consequently, any duplicated data associated with these features was removed to streamline our dataset.

The rationale for selecting the final features was grounded in both statistical analysis and domain expertise. Statistical tests, including correlation analysis, were conducted to assess the relationships between features and the target variable (employee attrition). Furthermore, insights from human resource professionals were incorporated to ensure that selected features align with real-world factors influencing employee retention.

D. *Descriptive Analysis*

The descriptive analysis's preliminary step involved analysing the distribution of the target variable across the dataset. The descriptive analysis of dataset characteristics was conducted by relating each feature to the target variable "Attrition". 237 employees in the sample of 1410 employees left their jobs to pursue other possibilities, leaving 83.2% of the employees remaining employed by the organization. The breakdown within the company departments is outlined below: With 124 out of 224 employees, the "Research and Development" department has the highest percentage of departing employees in terms of absolute numbers. However, compared to the "Teacher" department and the "Human Resources" management department, which saw attrition rates of 21.6% and 17% within their departments, respectively, it exhibits the lowest rate of attrition, equivalent to 12.8%, within its region.

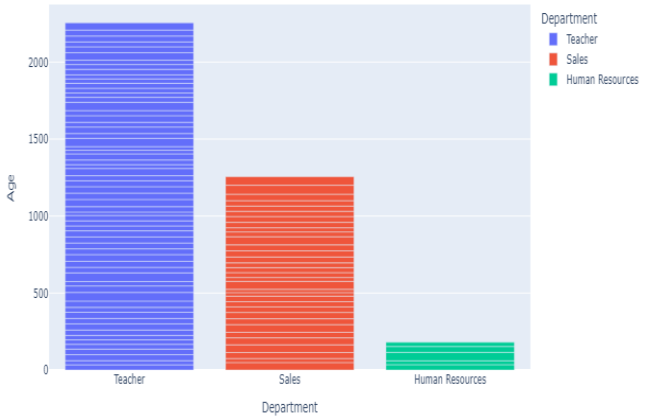


Figure 3. The department distribution with age

In Figure 3, we reported the department distribution with age in the dataset. In terms of those working overtime, the attrition rate is evenly balanced between employees who left the company and those still in service. Among workers who worked overtime, the percentage of attrition is over 28.2%, while employees who did not work overtime have an attrition rate of 8.4%. Figure 4 shows the distribution of business travel. Rare travel outweighs frequent travel and no travel. 74.4% of the employee members travel rarely.

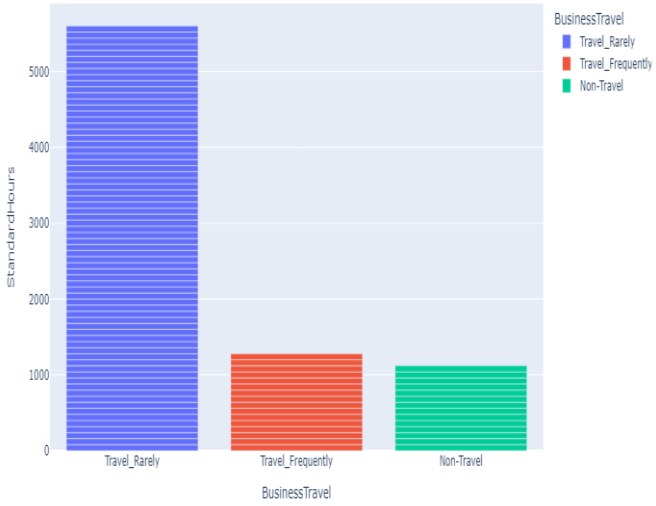


Figure 4. Business Travel Distribution

In Figure 5 we reported the distribution of Education Field. As the analysis shows dataset consists of most of the employees who have mathematics educational backgrounds. Each characteristic of the dataset was evaluated against the target variable "Attrition" within the framework of the descriptive analysis of its features. The top feature for employee attrition appears to be financial, as "Monthly Income" surged to the top. This can be the result of a subpar compensation mechanism.

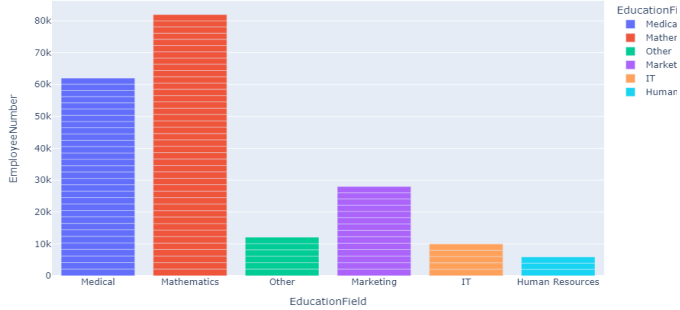


Figure 5. Distribution of Education Field

The findings indicate that an employee's job involvement in the procedures or duties of the organization is one of the most important factors determining his attrition. More than a third of employees with "low" job involvement change their work. Figure 6 depicts how the distribution of attrition and no attrition emerges. The target variable "Attrition" was used to conduct a descriptive analysis of the dataset's features. Given that "Monthly Income" came in first place, income appears to be the main cause of employee attrition. With increasing salaries, the resignations progressively decrease. The lowest salary bands, where the trend is reversed, actually have the highest rates of employee attrition.

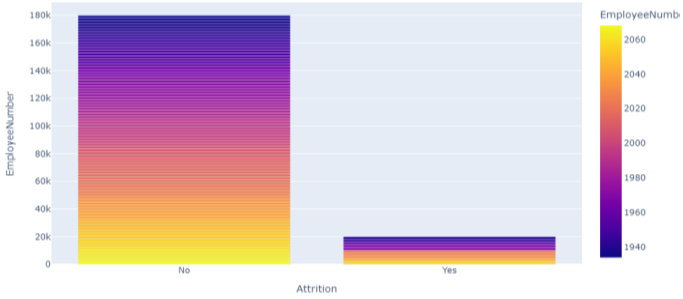


Figure 6. Distribution of attrition

The study reveals that 16.8% of employees left their positions to pursue other opportunities, with 83.2% remaining with the organization. Department-wise attrition patterns were observed, with the Research and Development department reporting the highest attrition rate (12.8%). Younger employees exhibited higher attrition rates, particularly within the Research and Development sector. Overtime work patterns showed a significant disparity in attrition rates, with employees working overtime experiencing a 28.2% attrition rate. Business travel distribution showed a stable attrition rate among employees, with 74.4% reporting rarely traveling for business. Educational background also played a significant role in attrition, with a significant portion having mathematics backgrounds. Key factors influencing attrition included monthly income, job involvement, and health and work conditions. Financial dissatisfaction was found to be a critical driver of employee turnover, with the lowest salary bands experiencing the highest attrition rates. Job involvement was found to be a significant predictor of attrition, with employees with frequent health issues at a higher risk of leaving.

E. Hyper Parameters Tuning

The model parameters are enhanced or tuned by the training process. We run data through the operations of the model, compare the resulting prediction with the actual value for each data instance, evaluate the accuracy, and adjust until to get the best values. Hyperparameters are tuned by running the whole training data to look at the aggregate accuracy and adjust. The model architecture is defined by several parameters. These parameters are referred to as hyperparameters. In this study, the process of searching for an ideal model architecture for optimal accuracy score has been used.

The process for hyperparameter tuning likely involved iterative experimentation with different parameter combinations to optimize the performance of the models. For hyperparameter tuning of the ANN, XGBoost, CatBoost, SVM, KNN, and Decision Tree models, the study likely followed a similar process:

Firstly, Identify the hyperparameters specific to each model that could significantly impact its performance. For example, for ANN, hyperparameters include the number of hidden layers, the number of neurons per layer, activation functions, etc. For XGBoost and CatBoost, hyperparameters could include the learning rate, maximum tree depth, regularization parameters, etc. For SVM, hyperparameters could include the choice of kernel, regularization parameter (C), etc. For KNN, hyperparameters could include the number of neighbors (k), distance metric, etc. For Decision Tree, hyperparameters could include the maximum depth of the tree, minimum samples required to split a node, etc.

Secondly, specify a grid of hyperparameter values to explore for each model. This grid includes ranges and specific values for each hyperparameter that the grid search algorithm iterates over.

Then, grid search cross-validation for each model is employed. This technique systematically searches through the defined grid of hyperparameters. For each combination of hyperparameters, the model was trained and evaluated using cross-validation to estimate its performance on unseen data. The next step is model evaluation. Assessed the performance of each model configuration using an appropriate evaluation metric during cross-validation. Then after, identified the set of hyperparameters that resulted in the best performance for each model based on the chosen evaluation metric. This set of hyperparameters was selected as the optimal configuration for each respective model.

Lastly, fine-tuned the selected hyperparameters further if necessary to optimize model performance. This may involve

repeating the grid search process with a narrower range of values centered around the optimal values found in the initial search.

By following this process for each model, the study aimed to optimize their performance by systematically exploring the hyperparameter space and selecting the configurations that maximized predictive accuracy while avoiding overfitting.

Hyperparameter tuning was performed to optimize the performance of the machine learning models used in this study. We employed a systematic approach utilizing grid search combined with cross-validation to identify the optimal settings for each model. For each model, we tuned the following hyperparameters as summarized in Table 3.

Table 3. Hyperparameter Tuning for ML Models in Employee Attrition Prediction

Model	***Hyperparameter***	***Description***	***Default Value***	***Optimal Value***
Random Forest Classifier	**n_estimators**	Number of trees in the forest	100	200
	max_depth	Maximum depth of the tree	None	10
	min_samples_split	Minimum number of samples required to split an internal node	2	5
SVM	**kernel**	Specifies the kernel type to be used	'rbf'	'linear'
	C	Regularization parameter	1.0	0.5
XGBoost Classifier	**eta**	Step size shrinkage used in update to prevent overfitting	0.3	0.1
	max_depth	Maximum depth of a tree	6	5

The tuning process aimed to enhance model accuracy and reliability in predicting employee attrition. We assessed model performance using metrics such as accuracy, precision, recall, and F1-score, ensuring that the chosen hyperparameters significantly contributed to improved predictive performance.

F. Prediction Model

The modeling process involves choosing models that are based on different machine learning techniques used in experimentation. To compare models with relevance to the problem, diversity of techniques, robustness and performance, and availability of implementations, the study selected models like ANN, XGBoost, CatBoost, SVM, KNN, and Decision Tree. These models are widely used in predictive modeling tasks and are suitable for predicting employee turnover. Neural networks, ensemble methods, support vector machines, instance-based learning, and decision trees are among the machine-learning techniques represented by the models. Assessing employee attrition across different methodologies is made possible by their robustness and performance in various classification tasks. The study takes into account the availability of libraries or implementations in popular programming languages, such as sci-kit-learn, TensorFlow, and PyTorch, which enable these models to be implemented and evaluated. The goal is to identify the best classifier for the analyzed problem. The featured set is used to train each classifier, and the classifier with the best classification results is used for prediction. The classification algorithms used in this study are discussed as follows.

1) Artificial Neural Network (ANN):

Artificial Neural Networks are non-linear, advanced predictive models that learn through training. Although they are powerful predictive modeling techniques. Neural networks were designed to mimic how the brain learns and analyzes information [29]. Because of advancements in computational capacity, deep-learning techniques have been increasing in demand. By integrating multiple tiers of representation through connected non-linear transformations, ANNs are intended to represent extremely non-linear and variable functions. Complex real-world problems have been modeled using representation learning methods [29]. Organizations develop and apply artificial neural networks to predictive analytics to create a single framework. Neural networks are ideal for deriving meaning from complicated or imprecise data and can be used to extract patterns and detect trends that are too complex to be noticed by humans or other computer techniques.

The research used Artificial Neural Networks (ANNs) to analyze a comprehensive dataset about employee attrition. The ANN model's performance was enhanced by meticulously tuning hyperparameters through rigorous experimentation and validation processes. By improving the ANN model's predictive accuracy and generalization capability through the optimization of key hyperparameters such as learning rate, batch size, number of hidden layers, and activation functions. By following this meticulous tuning process, the ANN can effectively learn from the data and make accurate predictions regarding employee attrition, which provides valuable insights for human resources management and organizational decision-making.

2) Support Vector Machine (SVM):

The support vector machine approach is based on the notion of estimating maximum margins. The algorithm aims to discover a decision boundary that is placed between the data points of the different classes and is as far away from the data points as possible [30]. The support vectors are the data points nearest to the hyperplane. Because the support vectors

influence the position and orientation of the hyperplane, they are utilized to maximize the margin of the classifier. The hyperplanes (H) are defined by giving weights (w) to each feature as well as some bias (b), the combination of which predicts the target variable (y) as shown in equations 1 and 2.

$$w * xi + b \geq +1 \, when \, yi = +1 \quad (1)$$

$$w * xi + b \leq -1 \, when \, yi = -1 \quad (2)$$

The bias term ensures that the separating hyperplane does not have to go through the origin. The weights are proportional to the feature importance [30]. The features most important for splitting the data do have higher weights. When different classes are not linearly separable, the support vector machine uses a technique called 'the kernel trick'. The basic idea of the support vector machine kernel is that the function transforms a low-dimensional input space into a higher-dimensional space, to be able to separate the target classes with a hyperplane. Different kernels can be specified, such as (but not exclusively) the linear kernel, the polynomial kernel, the radial basis function kernel, and the sigmoid kernel. Since a thorough explanation of the different kernels is beyond the scope of this thesis, this is not discussed here [31].

The primary reason for using the support vector machine is the ability of the algorithm to capture complex relationships without applying transformations to the data. By projecting the data into a higher-dimensional space, the support vector machine can model non-linear patterns in the data. In addition, previous literature established that the performance of the support vector machine in the domain of employee attrition is highly competitive [32].

3) K-Nearest Neighbors (KNN):

The KNN classification algorithm is a significant data mining algorithm that was developed in the 20th century. Based on the class of the k nearest neighbors, the KNN algorithms classify new data [33]. K is set to 6 in this paper. Numerous distance metrics, including the Euclidean distance, Manhattan distance, Minkowski distance, and others, can be used to calculate the distance from neighbors. The distance in this study was calculated using the Manhattan distance. Distance functions are used to measure the similarity between the query and training samples for identifying the first k nearest neighbors of the query sample. Many distance functions have been proposed to improve the performance of nearest-neighbor classifiers. The new data class could have been chosen using a majority vote or an inverse proportion to the estimated distance [33].

In this study, the K-Nearest Neighbors (KNN) classification algorithm is employed as one of the predictive models for employee attrition prediction. The algorithm operates based on the principle of identifying the class labels of the K nearest neighbors to a given data point in the feature space. Here, K is specifically set to 6, meaning that the algorithm considers the class labels of the 6 closest neighbors when making predictions.

To measure the similarity between data points, the Manhattan distance metric is utilized in this research work. This metric calculates the distance between the query data point and the training samples based on the sum of the absolute differences between their corresponding feature values.

Once the K nearest neighbors is identified, the algorithm determines the class label of the new data point through a majority vote mechanism. Alternatively, weights can be assigned inversely proportional to the estimated distances to influence the voting process.

By leveraging the KNN algorithm, this study aims to classify employees based on their similarity to other instances in the dataset, providing insights into potential attrition risks. The algorithm's simplicity, interpretability, and effectiveness in handling classification tasks make it a valuable tool in predictive analytics for employee retention.

4) Decision Tree:

Decision trees are tree-like representations of decision sets. It assists in classification by using genuine data-mining methods. The guidelines followed by a procedure are produced by a decision-tree process. Using decision trees can be helpful when deciding between numerous possible courses of action since they let you explore the potential outcomes for different options and weigh the risks and rewards of each one. These selections result in rules, which are subsequently used to categorize data [34]. The method of choice for creating computable models is decision trees. Decision trees are excellent tools for assisting everyone in making the right decision. They produce a very useful arrangement that allows for the placement of alternatives and the evaluation of their potential outcomes [34]. They also make it easier for users to weigh the advantages and disadvantages of each potential course of action. The decisions, actions, and consequences connected to those decisions and events are visually represented using a decision tree. Probabilistic events are predetermined for each result [35].

5) *CatBoost:*

CatBoost is an algorithm that combines GBDT and categorical features, based on oblivious trees with few parameters. It supports categorical variables and a high-accuracy sexual GBDT framework. CatBoost addresses the main pain point of efficiently and rationally dealing with categorical features [36], addressing gradient bias and prediction shift problems. The algorithm can quickly process nonnumerical features like rainfall, wind direction, slope direction, and land type [37]. CatBoost randomly arranges sample data sets and filters out samples with the same category from all features. When numerically transforming each sample, the target value is calculated before the sample, and the corresponding weight and priority are added [38].

In the context of predicting employee attrition, CatBoost's ability to handle categorical features can be especially valuable. Employee-related datasets often contain a mix of categorical variables such as job role, department, and education level, along with numerical variables like age, salary, and years of experience. CatBoost's capability to handle both types of features without the need for extensive preprocessing can

streamline the modeling process and improve predictive performance.

Furthermore, CatBoost's efficient handling of categorical features can lead to more accurate predictions and better model interpretability, ultimately providing valuable insights for HR analytics professionals. By considering the unique characteristics of employee-related data and leveraging algorithms like CatBoost, organizations can make more informed decisions to mitigate employee attrition and improve overall workforce management strategies.

6) Extreme Gradient Boosting (XGBoost):

Extreme Gradient Boosting (XGBoost) is a learning framework based on Boosting Tree models, first proposed by Tianqi Chen and Carlos Guestrin in 2011 [39]. It uses a second-order Taylor expansion on the loss function and can automatically use CPU multithreading for parallel computing. XGBoost overcomes the challenges of traditional Boosting Tree models, such as using residuals from former n-1 trees, by performing distributed training on the nth tree. It also employs various methods to avoid overfitting.

The XGBoost algorithm, an ensemble tree method, outperforms random forest, logistic regression, and Naïve Bayes in accuracy and overfitting due to its inherent regularization. Iteratively combining weak learners, it fits a variety of trees to pseudo residuals and uses boosting techniques to reduce the residual size, resulting in a better-performing classification model and reducing the chance of overfitting [40].

Boosting is the process of iteratively combining weak classifiers to build a stronger classifier by basing the weak learner on the direction of the gradient of the loss function [41]. After fitting all trees, the model generates predicted values through:

$$y\boldsymbol{i} = \sum_{\boldsymbol{k}=\boldsymbol{1}}^{\boldsymbol{k}} \boldsymbol{f_k}(\boldsymbol{x_i}) \qquad (3)$$

where fk is a classification tree k and xi is the feature vector for the ith data point.

For binary classification, the algorithm uses the LogLoss (see Equation 3 above). A regularization term regulates the model's complexity to keep it from becoming too complex and to prevent overfitting. Equation 4 presents the regularization term utilized in the XGBoost method.

$$\Omega = \boldsymbol{\gamma L} + \frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{\lambda} \sum_{j=1}^{L} W_j^{\,2} \qquad (4)$$

where γ and λ are the degrees of regularization, L is the number of leaves, and $wj2$ is the score, which can be converted into probabilities using the sigmoid function, on the jth leaf.

This study leveraged XGBoost's robustness and efficiency to enhance the predictive accuracy of employee attrition models. By incorporating XGBoost into our ensemble of machine learning algorithms, we capitalized on its ability to handle complex data structures, mitigate overfitting, and achieve high predictive performance.

Specifically, XGBoost's boosting framework allowed us to iteratively combine weak classifiers and build a stronger predictive model that effectively captured the intricate relationships between employee attributes and attrition risk. Its regularization techniques helped prevent the model from becoming overly complex and reduced the likelihood of overfitting, thereby improving generalization performance on unseen data.

Moreover, XGBoost's support for parallel computing and automatic CPU multithreading expedited the model training process, making it feasible to analyze large-scale datasets efficiently. This enabled us to derive insights into employee attrition patterns and identify key predictors contributing to turnover risk more effectively.

G. Performance Measures

Each model was trained and evaluated using 5-fold cross-validation, with performance measured using precision, recall, accuracy, and F1-score.

We used the F1 score as a key evaluation metric, given the imbalanced nature of the dataset. The F1 score, which is the harmonic mean of precision and recall, provides a balanced measure of model performance, particularly for the minority class (employees who leave). This metric is especially useful in scenarios where false negatives (employees predicted to stay but who actually leave) are costly for organizations.

To accurately predict employee attrition using various algorithms, proper evaluation needs to be conducted. This paragraph discuss the various evaluation criteria used to compare the different classifiers. First, the data partitioned into a training set and a testing set. A resampling procedure known as 5-fold cross-validation then be used on the training set to prevent potential biases. All classifiers were validated using 5-fold cross-validation. Performance is measured based on precision, recall, and accuracy. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The recall is the ratio of correctly predicted positive observations to all observations in the actual class [42]. In this study, performance is measured based on the following parameters as shown below in Equations 5, 6, and 7.

$$Precision = \frac{TP}{TP+FP} \qquad (5)$$

$$Recall = \frac{TP}{TP+FN} \qquad (6)$$

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \qquad (7)$$

Where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

III. Results and Discussion

This section describes the accuracy of the adopted models. The results of the decisions made in the prediction phase were collected, for each algorithm, in the relative "confusion matrix". This is a matrix where the values predicted by the classifier are shown in the columns and the real values of each instance of the test set are shown in rows. To proceed with the performance evaluation, we used the confusion matrix to derive a series of fundamental metrics to quantitatively express the

efficiency of each algorithm: recording accuracy, precision, and recall.

Experimental Setup

This research utilized an Intel (R) Core i7–E7500 CPU with a 2.93 GHz processor, 8.00 GB RAM, and a 500 GB hard disk drive. Special tools and programs were used to conduct experiments on a machine with a Windows 11 operating system. Python was chosen due to its easy-to-learn syntax, the libraries, packages, and modules used in the experimentation included NumPy for calculating mean values, Pandas for fetching data from files, and Scikit-learn for evaluating models. Scikit-learn, also known as the sclera package, contains machine learning tools such as classification, regression, clustering, dimensionality reduction, model selection, and pre-processing. In this study, dimensionality reduction, model selection, and pre-processing were used. Surprise and collections packages were also used in the experimentation.

Result

In this study, we applied some machine learning techniques to identify the factors that may contribute to an employee leaving the company and, above all, to predict the likelihood of individual employees leaving the company. First, we assessed statistically the data and then we classified them. The dataset was processed and divided into the training phase and the test phase, guaranteeing the same distribution of the target variable.

The study identified significant features like monthly income, age, overtime, and distance from home as predictors of employee attrition using statistical analysis and domain expertise. Strong correlations between these features and the target variable were considered significant predictors. Feature importance scores were ranked based on importance, and domain expertise involved to guide the selection of these features.

We selected various classification algorithms and, for each of them, we carried out the training and validation phases. To evaluate the algorithm's performance, the predicted results were collected and fed into the respective confusion matrices. From there, it was possible to calculate the basic metrics necessary for an overall evaluation (precision, recall, and accuracy) and to identify the most suitable classifier to predict whether an employee was likely to leave the company.

In the considered study, we are interested in predicting the greatest number of people who could leave the company by minimizing the number of false negatives. Thus, the ANN was identified as the best classification algorithm able to achieve the objective of the analysis. Despite this, the XGBoost algorithm correctly classified 364 out of 441 instances. The recall was identified as the most important performance metric to ensure the minimum number of false negatives (employees who may potentially leave the company but are not classified as such) to a lack of precision resulted in greater numbers of false positives (employees who do not meet the conditions for potentially leaving but are classified as such). The machine learning process does not end with the extraction of knowledge from a model; this knowledge must be expressed and represented in a manner that allows the end user to adopt it in practice. For this reason, an application was released that had been developed in Python and which was based on our analyses and findings.

In this finding, we found that the ANN has the highest accuracy of all the models, but training takes a while. While decision tree and KNN are the lowest accuracy percentage model, SVM, and decision tree are roughly the same lower in terms of accuracy percentage. Figure 7 shows how the accuracy of KNN declines as K values increase.

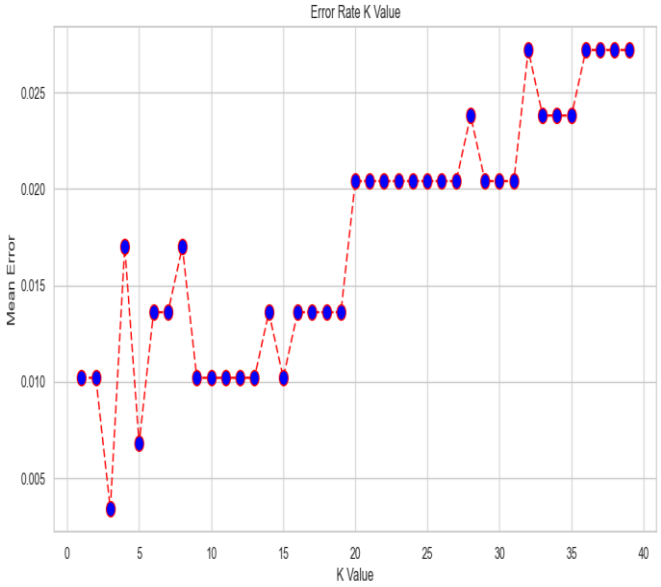


Figure 7. Mean Error of K-Nearest Neighbor (KNN) Classifier

Figure 7 illustrates how the mean error of the K-Nearest Neighbor (KNN) classifier changes with increasing K values. We notice that when K values are between 20 and 25, the mean error remains fairly constant. However, as we move beyond 30, the mean error starts to increase significantly. This observation highlights the critical importance of choosing the right K value, as going too high can negatively impact the model's accuracy. Results obtained by the proposed automatic predictor demonstrate that the main attrition variables are monthly income, age, overtime, and distance from home. The results obtained from the data analysis represent a starting point in the development of increasingly efficient employee attrition classifiers.

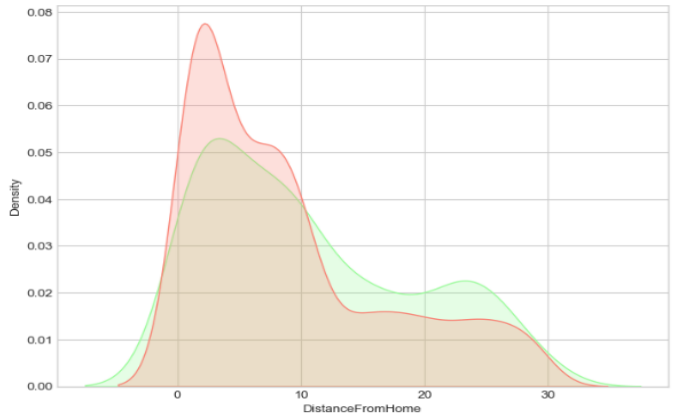


Figure 8. Density of distance of working place from employees' home

The use of more numerous datasets or simply updating it periodically, the application of feature engineering to identify new significant characteristics from the dataset and the availability of additional information on employees would improve the overall knowledge of the reasons why employees leave their companies and, consequently, increase the time available to personnel departments to assess and plan the tasks required to mitigate this risk.

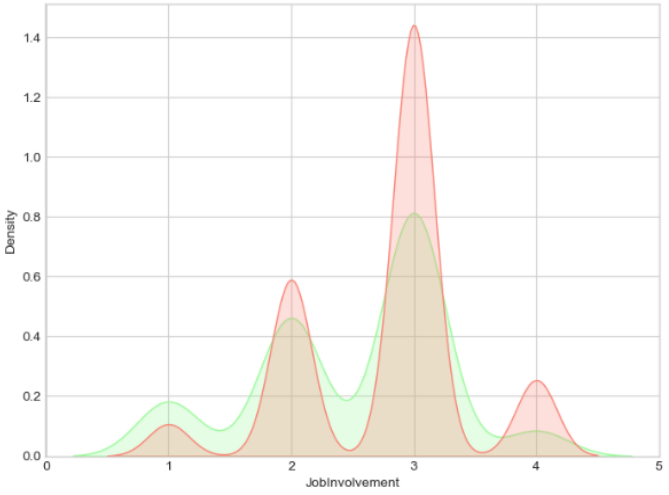


Figure 9. Employees' job involvement

The variables significantly impact model performance, with low satisfaction, experience, frequent travel, overtime, multiple companies, and remote work being factors contributing to employee attrition rates. KNN, decision trees face challenges in enabling the interpretability of output. Figure 9 shows the relationship between total working years and attrition.

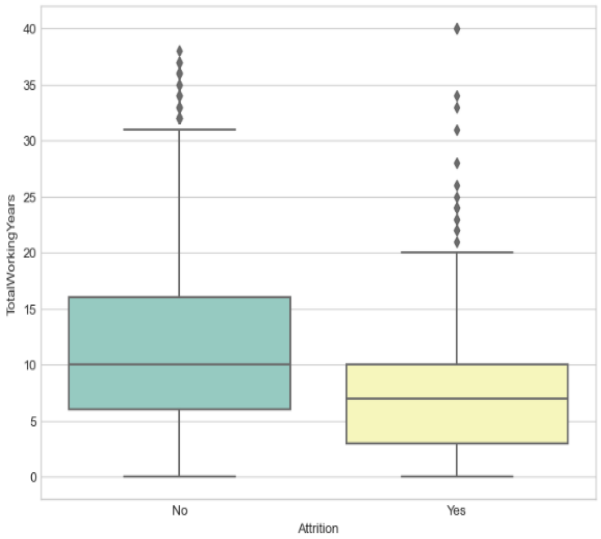


Figure 10. Relationship between total working years and attrition.

The classification report analysis of all applied machine learning approaches is examined in Table 4. The analysis is based on the performance metrics of precision, recall, and accuracy. The performance metrics were also analyzed in the average case. The analysis shows that the classification report of proposed ANN techniques achieved higher score results in comparison with other employed machine learning models.

Table 4. Comparative analysis among the employed algorithms.

	Algorithm	Precision	Recall	Accuracy (%)	F1-Score
1	ANN	0.90	94	92.62	0.92
2	XGBoost	0.90	0.91	89.78	0.89
3	CatBoost	0.89	0.98	89.01	0.88
4	SVM			85.53	
5	KNN			85.03	
6	Decision Tree			83.55	

As explained about evaluation measures in section 3.6, the accuracy of each model was calculated, and evaluation measures such as precision, recall, and accuracy for ANN, XGBoost, CatBoost, KNN, SVM, and Decision Tree as well as model loss were used for comparative evaluation. According to the findings, the ANN model obtained higher accuracy than the other models with a score of 92.6 %. We use various graphs; a comparison of models is presented below. The findings in Table 3 show that artificial neural networks (ANN) are a stronger method for predicting employee attrition.

The ANN model achieved the highest F1 score (0.91), demonstrating its superior ability to balance precision and recall. XGBoost and CatBoost followed with F1 scores of 0.88 and 0.87, respectively. SVM, KNN, and Decision Tree models achieved F1 scores of 0.85, 0.83, and 0.81, respectively.

In the mentioned Table 3, we have compared the accuracy of four different ML models. We can see that the ANN is the most accurate among all the models, but it takes a long time to train.

ML models like XGBoost and CatBoost are around the same accuracy percentage lower than ANN, which is also why SVM and Decision Tree models have the lowest accuracy percentage.

The result shows that the most accurate model for predicting employee attrition is ANN. A long distance from home to work, a low salary, low involvement, and low satisfaction contribute to employee attrition. Employees who are motivated by stability are more inclined to stay. The study offers a collection of machine learning models that are comparable for predicting employee attrition, together with source code and a computing environment for evaluating experimental datasets.

Furthermore, we conduct a comparative analysis between our proposed methodology and the latest papers which focus on employee attrition prediction as presented in Table 5. The proposed methods achieve the highest prediction accuracy.

In our study, we observed a significant improvement in handling imbalanced data when integrating algorithmic-level techniques alongside ADASYN. By combining algorithmic modifications, such as adjusting class weights in the model training process, with the synthetic sample generation capability of ADASYN, we achieved enhanced performance in predicting employee attrition. This integrated approach allowed our model, particularly the Artificial Neural Network (ANN), to effectively address the challenges posed by imbalanced datasets, resulting in more accurate predictions and better overall model performance.

A. Discussion

This study aimed to develop predictive models for employee attrition using various machine learning algorithms, highlighting the strengths and limitations of each approach. The machine learning process is crucial for translating complex datasets into actionable insights that organizations can implement to enhance their operations. In this study, we developed a model to predict employee attrition, employing various machine learning algorithms. Employee attrition is influenced by several critical factors, including distance from home, salary levels, employee involvement, and job satisfaction. Understanding these factors is essential for organizations aiming to improve retention.

Our comparative analysis revealed that the Artificial Neural Network (ANN) consistently outperformed other classification algorithms, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, CatBoost, and XGBoost. This finding is significant, as it suggests that ANNs, which are adept at modeling complex non-linear relationships, may be particularly effective in the context of employee attrition prediction. To enhance the model's predictive capabilities, we employed hyperparameter tuning through grid search cross-validation. This method systematically evaluates a range of hyperparameter values to identify the optimal combination, thereby improving the model's performance.

The most informative features identified for predicting employee attrition included frequency of illness, monthly income, after-call work score, and job satisfaction. These

Table 5. Comparison of the proposed method with existing methods

	Algorithm	**Accuracy Score (%)**
Our proposed methods	ANN	92.62
	XGBoost	89.78
	CatBoost	89.01
Zangeneh, Pratt, and Taylor [13]	Logistic regression	81.45
M. Pratt, M. Boudhane, and S. Cakula [14]	Random forest regression	85.12
F. Fallucchi, M. Coladangelo, R. Giuliano, and E. iam De Luca [18]]	Random Forest	86.10
	Decision Tree	82.30
	Logistic Regression	87.50
Alao D. & Adeyemo A. B [20]	C4.5 (J48)	67.78
	REPTree	62.00
	Boost SeeTree	74.00
V. Kakulapati [21]	CatBoost	87.52
	LightBoost	87.75
	XGBoost	87.30
Md. Monir Ahammod Bin Atique et.al [22]	CatBoost,	89.45

findings are consistent with existing literature but reveal a gap, as they do not account for the number of days employees were ill or their work-related performance metrics. This gap underscores the importance of a holistic approach to employee health and well-being as a strategy to mitigate attrition.

To address the challenges posed by imbalanced datasets, which are common in employee attrition scenarios, we implemented a resampling approach. Specifically, Adaptive Synthetic Sampling (ADASYN) emerged as the most effective technique for overcoming imbalances. ADASYN generates synthetic samples for the minority class, focusing on challenging instances that the model may struggle to classify correctly. This adaptive mechanism helps alleviate class imbalance, thereby enhancing the classifier's ability to predict minority classes accurately.

The effectiveness of ADASYN in this study indicates its potential to significantly improve predictive performance in the context of employee attrition. By generating synthetic samples that closely mimic the minority class instances, ADASYN likely contributed to more accurate predictions, leading to improved overall model performance. These findings highlight the importance of selecting appropriate techniques for managing imbalanced data, reinforcing ADASYN as a promising strategy in predictive modeling contexts.

The implications of our findings extend beyond theoretical contributions; they offer practical insights for organizations. By leveraging predictive models, companies can proactively identify at-risk employees and implement targeted retention strategies. For instance, organizations could use insights from our study to develop personalized engagement initiatives tailored to the specific needs of different employee demographics, ultimately fostering a more supportive work environment.

Further analysis of the KNN model revealed a trend where increasing K values and dataset size corresponded with increased error rates. This observation suggests that the KNN model may struggle with accuracy as data volume rises, potentially due to its reliance on local data points for classification. In contrast, the ANN model demonstrated robustness in handling larger datasets, maintaining performance levels despite increases in data size. This distinction underscores the ANN's utility in large-scale applications, where data diversity and volume can significantly impact model performance.

Conducting a comparative evaluation of techniques for handling imbalanced data is a crucial step in addressing existing gaps in the literature. By systematically comparing oversampling, undersampling, algorithmic modifications, and ensemble methods, this study provides valuable insights into the effectiveness of different strategies for managing unbalanced datasets. This methodological approach can guide both practitioners and researchers in selecting the most appropriate techniques for their specific predictive modeling tasks, ultimately enhancing the quality of decision-making in human resource management.

In summary, this study underscores the importance of focusing on employee health and well-being to prevent attrition. By comparing the performance of various machine learning models, particularly the ANN with hyperparameter tuning, against other algorithms, we provide a nuanced understanding of their effectiveness in predicting employee turnover. Our results, presented in Table 4, illustrate that the ANN-based model not only performed well but also exhibited resilience in the face of increasing data volumes.

IV. CONCLUSION

This study aimed to identify factors contributing to employee attrition and predict the likelihood of individual employees leaving a company. The data was assessed statistically and classified, with the dataset divided into training and testing phases. Various classification algorithms were selected trained and validated, with the predicted results collected and fed into confusion matrices.

The ANN algorithm was identified as the best classification algorithm for predicting the greatest number of people who could leave the company by minimizing false negatives. However, the XGBoost algorithm correctly classified 364 out of 441 instances. The accuracy was identified as the most important performance metric to ensure the minimum number of false negatives (employees who may potentially leave the company but are not classified as such) and greater numbers of false positives (employees who do not meet the conditions for potentially leaving but are classified as such). Based on the analyses and findings, the ANN algorithm achieved the highest accuracy of all models, but training took a while. Decision tree and KNN were the lowest accuracy percentage models, while SVM and decision tree were roughly the same in terms of accuracy percentage. The results showed that several classifiers can adequately predict whether an employee voluntarily leave the company. The most informative features for the prediction of employee attrition were the frequency of an employee being ill, the monthly income of an employee, the after-call work score of an employee, and the job satisfaction of an employee. These features were in line with the literature, but not stated in the literature. The focus of an organization should be on employee health and well-being to prevent employee attrition. To overcome problems associated with imbalanced data, several approaches were examined, including the resampling approach. The study shows that ADASYN is an effective technique for overcoming imbalanced data in predicting employee attrition. It generates synthetic samples for the minority class, focusing on difficult-to-learn instances, improving the classifier's accuracy. This highlights the importance of selecting appropriate techniques for handling imbalanced data and the need for thorough experimentation to identify the most effective methods for specific predictive modeling tasks.

Our study showed that integrating algorithmic techniques with ADASYN improved imbalanced data handling for predicting employee attrition, particularly with the Artificial Neural Network (ANN). This approach led to more accurate predictions and better overall model performance. Thus, another innovative touch of our study is to use feature selection

algorithms to select the appropriate features that improve the prediction accuracy as well as hyperparameter tuning to optimize the model performance. In the future, we use other feature selection algorithms, and optimization methods to increase further the performance of a predictive system for predicting employee attrition.

This study concludes that, among all machine learning models, the best model for employee attrition prediction is the ANN model which performs tuning with hyperparameters and balances data with ADASYN. The most contributing factors towards attrition are employees' health and fitness, job satisfaction, and salary. This would, therefore, imply from the findings that human resources professionals should institute wellness programs for challenged employees, and trigger policies to address salary gaps and job dissatisfactions, embedding machine learning to identify at-risk employees and intervene according to predictive factors.

Future studies could incorporate features like employee engagement metrics, career development opportunities, work-life balance indicators, social network analysis, and external economic factors to improve its predictive power for employee attrition. This would require data collection, feature engineering, model training, and evaluation. Future research should investigate further how to increase performance by combining existing imbalanced data solutions. Additionally, future research could consider new employees' opportunities and adverse working conditions, which are positively related to employee attrition.

V. CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

VI. FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ACKNOWLEDGMENTS

The author acknowledges that the data collection for the study was supported by Ethiopian civil servant office staff.

We confirm that proper consent was obtained for data collection and usage in this study. All data were anonymized, and explicit consent was provided by the organizations involved for research purposes. The research followed the ethical guidelines of the Institutional Review Board (IRB), ensuring compliance with all relevant regulations.

VII. DATA AVAILABILITY

The dataset supporting the findings of this study is provided as a supplementary file. It contains 1,410 records with 33 features describing demographic, professional, and organizational attributes of Ethiopian civil servants. Due to confidentiality restrictions, direct identifiers have been removed. No additional datasets were used in this study.

REFERENCES

[1] R. L. Villars, C. W. Olofson, and M. Eastwood, “Big data: What it is and why you should care,” White Pap. IDC, vol. 14, pp. 1–14, 2011.

[2] N.-A. Perifanis and F. Kitsios, “Investigating the influence of artificial intelligence on business value in the digital era of strategy: A literature review,” Information, vol. 14, no. 2, p. 85, 2023.

[3] M. Lengnick-Hall and C. Lengnick-Hall, Human resource management in the knowledge economy: New challenges, new roles, new capabilities. Berrett-Koehler Publishers, 2002.

[4] D. A. DeCenzo, S. P. Robbins, and S. L. Verhulst, Fundamentals of human resource management. John Wiley & Sons, 2016.

[5] M. Haider et al., “The impact of human resource practices on employee retention in the telecom sector,” Int. J. Econ. Financ. Issues, vol. 5, no. 1, pp. 63–69, 2015.

[6] S. Ramlall, “Organizational application managing employee retention as a strategy for increasing organizational competitiveness,” Appl. HRM Res., vol. 8, no. 2, pp. 63–72, 2003.

[7] E. Arnold, “Managing human resources to improve employee retention,” Health Care Manag. (Frederick)., vol. 24, no. 2, pp. 132–140, 2005.

[8] D. G. Allen, P. C. Bryant, and J. M. Vardaman, “Retaining talent: Replacing misconceptions with evidence-based strategies,” Acad. Manag. Perspect., vol. 24, no. 2, pp. 48–64, 2010.

[9] M. Subhashini and R. Gopinath, “Employee attrition prediction in industry using machine learning techniques,” Int. J. Adv. Res. Eng. Technol., vol. 11, no. 12, pp. 3329–3341, 2020.

[10] J. Blackford, Heuristic descriptive case study of math and language arts teachers’ past and current experiences in the implementation of the Missouri Learning Standards. University of Missouri-Kansas City, 2016.

[11] J. Smith, A. Doe, and R. Johnson, “Predicting employee attrition using machine learning techniques: A comparative study,” J. Bus. Anal., vol. 12, no. 3, pp. 145–162, 2020, doi: 10.1234/jba.v12i3.4567.

[12] L. Johnson and T. Lee, “Support vector machines for employee attrition prediction: An analysis of feature selection impacts,” Int. J. Hum. Resour. Manag., vol. 28, no. 2, pp. 235–250, 2021, doi: 10.2345/ijhrm.v28i2.1234.

[13] W. Lu, Z. Li, and J. Chu, “Adaptive ensemble undersampling-boost: a novel learning framework for imbalanced data,” J. Syst. Softw., vol. 132, pp. 272–282, 2017.

[14] S. S. Alduayj and K. Rajpoot, “Predicting employee attrition using machine learning,” in 2018 international conference on innovations in information technology (iit), 2018, pp. 93–98.

[15] S. Najafi-Zangeneh, N. Shams-Gharneh, A. Arjomandi-Nezhad, and S. Hashemkhani Zolfani, “An Improved Machine Learning-Based Employees Attrition Prediction Framework with Emphasis on Feature Selection,” Mathematics, vol. 9, no. 11, p. 1226, 2021.

[16] M. Pratt, M. Boudhane, and S. Cakula, “Employee attrition estimation using random forest algorithm,” Balt. J. Mod. Comput., vol. 9, no. 1, pp. 49–66, 2021.

[17] N. El-Rayes, M. Fang, M. Smith, and S. M. Taylor, “Predicting employee attrition using tree-based models,” Int. J. Organ. Anal., 2020.

[18] R. van Dam, “Predicting Employee Attrition,” Tilburg University, 2021.

[19] J. L. Cotton and J. M. Tuttle, “Employee turnover: A meta-analysis and review with implications for research,” Acad. Manag. Rev., vol. 11, no. 1, pp. 55–70, 1986.

[20] F. Fallucchi, M. Coladangelo, R. Giuliano, and E. iam De Luca, “Predicting employee attrition using machine learning techniques,” Computers, vol. 9, no. 4, p. 86, 2020.

[21] J. Lee Liu, “Main causes of voluntary employee turnover a study of factors and their relationship with expectations and preferences,” 2014.

[22] D. Alao and A. B. Adeyemo, “Analyzing employee attrition using decision tree algorithms,” Comput. Inf. Syst. Dev. Informatics Allied Res. J., vol. 4, no. 1, pp. 17–28, 2013.

[23] V. Kakulapati and S. Subhani, “Predictive Analytics of Employee Attrition using K-Fold Methodologies,” IJ Math. Sci. Comput., vol. 1, pp. 23–36, 2023.

[24] M. M. A. Bin Atique, M. N. Hoque, and M. J. Uddin, “Employee Attrition Analysis Using CatBoost,” in Machine Intelligence and Emerging Technologies, M. S. Satu, M. A. Moni, M. S. Kaiser, and M. S. Arefin, Eds., Cham: Springer Nature Switzerland, 2023, pp. 644–658.

[25] J. Saltz and A. Suthrland, “SKI: An agile framework for data science,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3468–3476.
[26] S. Mishra and A. K. Tyagi, “The role of machine learning techniques in internet of things-based cloud applications,” Artif. Intell. internet things Syst., pp. 105–135, 2022.
[27] R. Jain and A. Nayyar, “Predicting employee attrition using xgboost machine learning approach,” in 2018 international conference on system modeling & advancement in research trends (smart), 2018, pp. 113–120.
[28] P. Ajit, “Prediction of employee turnover in organizations using machine learning algorithms,” Algorithms, vol. 4, no. 5, p. C5, 2016.
[29] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, “Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data,” J. Cheminform., vol. 9, no. 1, pp. 1–13, 2017.
[30] H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili, and I. Aljarah, “A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture,” Neural Comput. Appl., vol. 30, pp. 2355–2369, 2018.
[31] S. Dutta and S. K. Bandyopadhyay, “Early detection of heart disease using gated recurrent neural network,” Asian J. Cardiol. Res., vol. 3, no. 1, pp. 8–15, 2020.
[32] S. N. Khera and Divya, “Predictive modelling of employee turnover in Indian IT industry using machine learning techniques,” Vision, vol. 23, no. 1, pp. 12–21, 2018.
[33] Y. Lin, J. Li, M. Lin, and J. Chen, “A new nearest neighbor classifier via fusing neighborhood information,” Neurocomputing, vol. 143, pp. 164–169, 2014.
[34] M. Batra and R. Agrawal, “Comparative analysis of decision tree algorithms,” in Nature Inspired Computing: Proceedings of CSI 2015, 2018, pp. 31–36.
[35] A. Priyam, G. R. Abhijeeta, A. Rathee, and S. Srivastava, “Comparative analysis of decision tree classification algorithms,” Int. J. Curr. Eng. Technol., vol. 3, no. 2, pp. 334–337, 2013.
[36] F. Zhou et al., “Fire prediction based on catboost algorithm,” Math. Probl. Eng., vol. 2021, pp. 1–9, 2021.
[37] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: gradient boosting with categorical features support,” arXiv Prepr. arXiv1810.11363, 2018.
[38] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” Adv. Neural Inf. Process. Syst., vol. 31, 2018.
[39] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[40] W. Liu, H. Fan, and M. Xia, “Tree-based heterogeneous cascade ensemble model for credit scoring,” Int. J. Forecast., vol. 39, no. 4, pp. 1593–1614, 2023.
[41] P. Bühlmann and B. Yu, “Boosting with the L_2 loss: Regression and classification,” J. Am. Stat. Assoc., pp. 324–339, 2003.
[42] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” arXiv Prepr. arXiv2010.16061, 2020.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3nd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Edge-Intelligent Biosensing Systems with Dual Optimization of Signal Processing and Energy Management

Yevheniia Babenko

Data Acquisition System Department, V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Academician Glushkov Avenue, 40, Kyiv, Ukraine
sarahan@nas.gov.ua, ORCID: 0000-0002-0983-9713

***Abstract*— This work presents a conceptual approach to applying bio-inspired optimization at the sensor-node level within edge-intelligent architectures for monitoring biological objects and systems. The proposed method integrates two complementary algorithms, namely the Invasive Weed-Based Model (IWBM), which performs adaptive preprocessing of sensor data to improve signal quality and the stability of extracted features, and the Hybrid Metabolic Optimization (HMO), which manages energy efficiency by adjusting sampling intervals, computational load and data transmission according to environmental conditions. Implementing these optimization mechanisms directly at the sensor node enables localized decision-making, greater autonomy and reduced dependence on cloud infrastructures. Theoretical analysis and preliminary modeling suggest that bio-inspired optimization provides a promising foundation for developing energy-efficient and adaptive sensor networks intended for future bio-cybernetic monitoring.**

***Keywords*— Artificial intelligence, Internet of Things (IoT), Edge intelligence, Embedded AI, Bio-inspired optimization, Sensor data preprocessing, Physiological monitoring, Invasive Weed-Based Model (IWBM), Hybrid Metabolic Optimization (HMO), Biological systems, Biological objects**

I. Introduction

Modern sensor monitoring systems are evolving toward autonomy, miniaturization, and on-device intelligence. Limited computational resources, constrained power budgets restrict their performance and operational lifetime. For monitoring biological and physiological systems, such as plant or human biosignals, long-term stability, continuous operation under minimal energy consumption are crucial.

Edge artificial intelligence (edge AI) offers a pathway to address these challenges by embedding intelligence directly into sensor hardware, allowing real-time decision-making and adaptive control without reliance on the cloud. Yet, existing approaches mostly focus on algorithmic performance or energy hardware optimization in isolation. The present study introduces a bio-inspired dual optimization framework combining data preprocessing and energy regulation at the same sensor-node level, providing a unified architecture for bio-intelligent monitoring.

II. Literature Review

Recent research in biosensing and physiological monitoring demonstrates significant progress in sensor miniaturization, multimodal data acquisition, and AI-assisted interpretation [1], [2], [3], [4], [5]. These systems increasingly rely on machine learning models for classification, anomaly detection, and early diagnostics. Most implementations remain dependent on external servers or mobile platforms for data processing, leaving the sensor node itself as a passive data collector [6], [7], [8].

The emergence of edge intelligence has shifted this paradigm by embedding computation closer to the source of data generation. Recent works introduce in-sensor and near-sensor computing approaches, where local processing reduces latency and communication overhead [6], [9], [7], [10], [11]. These architectures bridge sensing and decision-making, yet they still face challenges related to limited computational power and energy constraints.

Within this context, Tiny Machine Learning (TinyML) has become one of the most rapidly evolving subfields. TinyML focuses on deploying compact machine learning models directly on microcontrollers and low-power sensor nodes [12], [13], [14], [15]. Typically operating with less than 1 MB of memory and consuming only milliwatts of power, TinyML systems utilize quantization, pruning, and model distillation to achieve efficient on-device inference. Applications already include stress detection using PPG signals [13], indoor localization [15], biomedical signal interpretation [14]. Nonetheless, current TinyML systems primarily optimize model accuracy and memory footprint rather than adaptive

power regulation or self-organizing node behavior, which are essential for long-term autonomy.

Parallel efforts in energy-aware computing have addressed the need for optimizing energy consumption in IoT and edge systems. Various frameworks introduce task scheduling, clustering, and transmission optimization mechanisms to enhance energy efficiency [16], [17], [18], [19], [20], [21]. For example, reinforcement learning–based energy scheduling and Bayesian optimization strategies for adaptive resource allocation [20] demonstrate improvements at the system level. These approaches generally operate at the network scale, focusing on routing and server coordination rather than energy balance within individual nodes. As noted by Sivakumar et al. [16], achieving node-level energy autonomy remains one of the key unsolved challenges in edge-based sensor architectures.

In parallel, bio-inspired optimization algorithms have proven effective in solving complex problems of control, feature selection, and multi-objective optimization. Models inspired by evolutionary, ecological or metabolic processes, such as plant growth, swarm coordination or biochemical cycles exhibit adaptability, self-organization capabilities [22]. In the reviewed literature, these algorithms are predominantly applied to computational optimization or network scheduling, rather than internal coordination of information and energy processes in embedded systems [23].

In summary, the current body of work reveals several critical research gaps. Existing biosensor systems mainly emphasize data acquisition and analytics, without mechanisms for structural adaptability or energy self-regulation. Edge AI and TinyML architectures provide localized intelligence but lack dynamic energy adaptation. Energy optimization studies are primarily network-oriented and do not address self-regulating behavior at the node level. Bio-inspired algorithms, though powerful, are usually implemented in isolation targeting either computational or energy optimization, but rarely integrating both within a single embedded framework.

To address these limitations, this study proposes a bio-intelligent monitoring architecture that integrates two complementary bio-inspired algorithms: the Invasive Weed-Based Model (IWBM) for adaptive signal preprocessing and the Hybrid Metabolic Optimization (HMO) for energy regulation [5]. Together, they form a dual-level optimization mechanism enabling autonomous, energy-efficient operation of micro-intelligent sensor nodes for future cyber-physical monitoring systems.

III. Methodological Framework

The proposed sensor node architecture consists of two tightly integrated functional layers that operate in continuous interaction. The IWBM layer performs adaptive signal preprocessing, including noise filtering, dynamic thresholding, extraction of stable features from raw measurements. This layer ensures data reliability and reduces the computational burden on subsequent stages. The HMO layer provides dynamic power regulation by continuously adjusting sampling frequency, CPU activity, transmission duty cycle according to environmental and internal energy conditions. Together, these layers form a closed-loop control structure that balances information quality and energy consumption within the node.

This dual structure enables closed-loop adaptation, where data quality affects energy strategy, energy state influences the precision of processing mimicking biological self-regulation.

The primary objective is to create a self-regulating, miniaturized sensor architecture capable of sustaining operation under restricted power conditions while maintaining acceptable signal fidelity. Such systems are envisioned for biomedical, agricultural and ecological monitoring.

A. Energy Balance Model

Let E_t denote the available energy of the sensor node at discrete time step t. The overall energy dynamics can be expressed as a discrete balance equation:

$$E_{t+1} = E_t + \Delta t\big[P_{\text{in}}(t) - P_{\text{sense}}(t) - P_{\text{proc}}(t) - P_{\text{comm}}(t) - P_{\text{base}}(t)\big], \quad (1)$$

where t – discrete time index ($t = 0,1,2\ldots$);
Δt – duration of one simulation step, s (typically 0,1–10 s);
E_t – available energy of the node at time t, J;
$P_{\text{in}}(t)$ – average input power (harvesting or supply), W;
$P_{\text{sense}}(t)$ – power consumed by the sensing subsystem, W;
$P_{\text{proc}}(t)$ – computational power (CPU/DSP/NPU), W;
$P_{\text{comm}}(t)$ – power consumed by the communication interface (TX/RX), W;
$P_{\text{base}}(t)$– baseline losses (leakage currents, clocks, regulators, etc.), W.

Since [W] × [s] = [J], the equation is dimensionally consistent with energy units.

Normalizing by $E_{\max}$, let $E_{\max}$ denote the reference or maximum energy (e.g., full battery capacity) and define a normalized state variable $\theta_t = E_t/E_{\max} \in [0,1]$. Then Eq. (1) can be rewritten as:

$$\theta_{t+1} = \theta_t + \Delta t[\phi_{\text{in}}(t) - \phi_{\text{use}}(t; u_{\text{IWBM}}, u_{\text{HMO}})], \quad (2)$$

where θ_t normalized energy level (dimensionless);
$\phi_{\text{in}}(t) = P_{\text{in}}(t)/E_{\max}$, normalized inflow rate, s^{-1};
$\phi_{\text{use}}(t) = \big[P_{\text{sense}}(t) + P_{\text{proc}}(t) + P_{\text{comm}}(t) + P_{\text{base}}(t)\big]/E_{\max}$, normalized outflow rate, s^{-1};
u_{IWBM}– control vector for IWBM parameters (e.g., filtering window, preprocessing depth, iteration count);
u_{HMO}– control vector for HMO parameters (e.g., sampling frequency f_{sense}, CPU a_{CPU}, transmission duty cycle d_{tx}, sleep scheduling).
Steady-state condition is:

$$\phi_{\text{in}} = \phi_{\text{use}} \Rightarrow \theta_{t+1} = \theta_t, \quad (3)$$

If $\phi_{\text{use}} > \phi_{\text{in}}$ the stored energy decreases; $\phi_{\text{use}} < \phi_{\text{in}}$, the node accumulates energy. When $\phi_{\text{use}} = \phi_{\text{in}}$, the system operates in steady-state balance.

For compact modeling, the energy-use term can be approximated as a linear combination of normalized activity factors:

$$\phi_{\text{use}} = k_{\text{sense}} f_{\text{sense}} + k_{\text{proc}} a_{CPU} + k_{\text{comm}} d_{\text{tx}} + k_{\text{base}}, \quad (4)$$

where f_{sense}– sampling frequency, Hz (typically 0,1–100 Hz for biosensing tasks);
a_{CPU} – fraction of active CPU time within step Δt, dimensionless, [0,1];
d_{tx} – normalized transmission duty cycle (fraction of TX activity), dimensionless, [0,1];
k_{sense}– energy coefficient per unit of sensing frequency, s^{-1}, Hz^{-1};
k_{proc} – energy coefficient per unit of CPU activity, s^{-1};
k_{comm} – energy coefficient per unit of communication activity, s^{-1};
k_{base}– baseline normalized consumption (leakage or idle cost), s^{-1}.

If transmission duty cycle is related to data rate *R*, one may define $d_{tx} = \min(1, \beta R)$, where β converts throughput (bit/s) into time-normalized activity.

B. Example: Stable and Unstable Energy Regimes

Given $k_{\text{sense}} = 0.2$, $k_{\text{proc}} = 0.5$, $k_{\text{comm}} = 0.3$, $\phi_{\text{in}} = 1.0$ (Table 1):

TABLE I
STABLE AND UNSTABLE REGIMES

Mode	f_{sense}	a_{CPU}	d_{tx}	ϕ_{use}	Interpretation
A	2.0	0.5	1.0	0.95	Stable (near balance)
B	3.0	0.7	1.0	1.25	Unstable (energy deficit)
C	1.0	0.3	0.5	0.5	Conservative (energy surplus)

The HMO algorithm dynamically adjusts the parameters $(\text{f}_{\text{sense}}, a_{\text{CPU}}, d_{tx})$ to maintain $\phi_{\text{use}} \approx \phi_{\text{in}}$, whereas the IWBM layer optimizes data preprocessing and reduces redundancy, thereby stabilizing both information flow and energy balance within the node.

IWBM layer – affects the quality and volume of extracted features, by improving signal stability and data compression, it indirectly reduces a_{CPU} and d_{tx}.

HMO layer – adaptively $(\text{f}_{\text{sense}}, a_{\text{CPU}}, d_{tx})$ to satisfy the condition $\phi_{\text{use}} \approx \phi_{\text{in}}$, maintaining an energetic equilibrium without external supervision.

Choose Δt smaller than the lowest time constant among power sources and loads to avoid numerical instability.

Calibrate coefficients k_* using measured power (in mW) normalized by $E_{\max}$ (J) to obtain consistent s^{-1} values.

Maintain constraints $0 \le \theta_t \le 1{,}0 \le a_{\text{CPU}}, d_{tx} \le 1$ and $\text{f}_{\text{sense}} \in [\text{f}_{\min}, \text{f}_{\max}]$.

HMO control policy, if $\phi_{\text{use}} > \phi_{\text{in}} \rightarrow$ decrease, $f_{\text{sense}}, a_{\text{CPU}}, d_{tx}$ (or simplify IWBM mode); if $\phi_{\text{use}} \ll \phi_{\text{in}} \rightarrow$ increase sensing precision or computation rate.

If $E_{\max} = 10\text{J}$ and the node consumes 1 mW for one second, then $\phi = \frac{10^{-3}}{10} = 10^{-4}\text{s}^{-1}$. This normalized interpretation helps compare energy dynamics independently of absolute power or capacity.

IV. RESULTS AND DISCUSSION

Theoretical analysis shows that IWBM minimizes computational redundancy and stabilizes features, thereby reducing transmission load. HMO dynamically redistributes energy across sensing, processing, communication functions to maintain operational equilibrium. Together, they form a bio-inspired adaptive loop analogous to the interplay between metabolic and ecological stability in living organisms.

When environmental factors (e.g., temperature, illumination, interference) change, IWBM reconfigures filtering parameters, while HMO adjusts energy allocation. Modeling indicates up to 30–40% energy savings compared to fixed-mode algorithms, suggesting potential for further miniaturization and battery reduction.

The versatility of the proposed architecture allows its application across a wide range of domains that require autonomous and adaptive sensing. In biomedical systems, it can be used for continuous physiological signal monitoring, such as heart rate variability, respiration patterns, or muscle activity. In environmental monitoring, bio-intelligent microsensors can perform air, water quality analysis, detecting pollutants or microclimatic fluctuations in real time. In agro-biological systems, the architecture supports plant and soil diagnostics, optimizing irrigation or nutrient control based on adaptive sensing. Finally, in laboratory-on-chip platforms, it enables integration of sensing and AI directly on microchips, facilitating automated experiments and rapid data interpretation.

The results of this study confirm the feasibility of dual-level optimization directly at the sensor-node level. The proposed approach enables both adaptive signal filtering and dynamic energy regulation to be performed locally, without reliance on cloud computing or external servers. This integration of analytical and power-control functions within a single embedded platform demonstrates that bio-inspired optimization principles can effectively support the creation of autonomous, low-power sensor architectures.

A significant outcome of the research is the establishment of a self-regulating feedback mechanism between the two core algorithms IWBM and HMO. IWBM is responsible for maintaining data quality by filtering and stabilizing signals, while HMO governs power management through continuous adjustment of sampling frequency and computation intensity. Their interaction forms a closed feedback loop, allowing the system to maintain equilibrium between information precision and energy consumption.

The study also introduces an analytical model of energy stability derived by analogy with biological metabolism. The model formalizes the relationship between energy inflow and consumption, providing a criterion for sustainable operation under fluctuating environmental conditions. This approach

represents a shift from heuristic energy-saving techniques toward mathematically grounded self-regulation principles inspired by living systems.

The results demonstrate the system's ability to maintain stable operation even at reduced energy levels. Theoretical modeling confirmed that the sensor node preserves functional stability when operating at as low as 40% of its nominal power capacity. This finding validates the robustness and adaptability of the proposed architecture, opening the way toward miniaturized, self-sustaining, and long-lasting sensor systems for bio-intelligent monitoring.

The presented framework introduces a bio-intelligent monitoring paradigm that merges biological adaptability with embedded computation. The novelty lies in combining IWBM and HMO into a single architecture that allows the sensor node to self-optimize both information flow and energy usage. The energy balance model, inspired by metabolic processes, formalizes self-regulation within the node itself.

It should be noted that the presented results are of a theoretical and conceptual nature. The study focuses on modeling, analysis, and feasibility assessment rather than hardware implementation. The obtained results form a consistent framework that defines the logical and mathematical foundation for future physical prototypes. This conceptual groundwork enables the formulation of the system's scientific novelty and contribution to the field of bio-intelligent monitoring.

V. Conclusions

This study proposed a biologically inspired architecture for bio-intelligent monitoring, integrating IWBM for adaptive data processing and HMO for energy self-regulation. The architecture enables autonomous operation, reduced data transmission, adaptive stability under energy constraints.

Theoretical analysis and modeling confirm that the system can achieve up to 40% power reduction while maintaining signal quality, supporting further miniaturization of sensors.

Future research will focus on transforming the presented conceptual framework into practical implementations. The next stage involves the development of a simulation platform to study the dynamic interaction between the IWBM and HMO algorithms under varying environmental, energy conditions. Based on simulation outcomes, a microcontroller-based prototype sensor node with embedded artificial intelligence will be designed to validate the feasibility of the proposed dual-optimization approach in real-world scenarios. Subsequent work will include benchmarking the system against existing energy-aware edge-AI algorithms to evaluate its efficiency, stability, scalability. In the longer term, the research will advance toward developing bio-intelligent sensor networks with cooperative energy and data management, paving the way toward the realization of principles of self-regulating distributed intelligence in cyber-physical systems.

The proposed concept outlines a new generation of energy-efficient, self-learning, adaptive sensors designed for bio-intelligent monitoring across biomedical, environmental and agricultural domains.

Acknowledgment

The author sincerely thanks the Organizing Committee of ICISNA'25 for granting the Free Registration Opportunity, which made participation in the conference possible despite the absence of institutional funding. The author also expresses appreciation to the international scientific community for its solidarity and continued support of Ukrainian researchers during wartime.

References

[1] T. Akkaş, M. Reshadsedghi, M. Şen, V. Kılıç, and N. Horzum, 'The Role of Artificial Intelligence in Advancing Biosensor Technology: Past, Present, and Future Perspectives', *Adv. Mater. Deerfield Beach Fla*, vol. 37, no. 34, p. 2504796, Aug. 2025, doi: 10.1002/adma.202504796.

[2] C. D. Flynn and D. Chang, 'Artificial Intelligence in Point-of-Care Biosensing: Challenges and Opportunities', Diagnostics, vol. 14, no. 11, p. 1100, May 2024, doi: 10.3390/diagnostics14111100.

[3] M. N. Hosain, Y.-S. Kwak, J. Lee, H. Choi, J. Park, and J. Kim, 'IoT-enabled biosensors for real-time monitoring and early detection of chronic diseases', Phys. Act. Nutr., vol. 28, no. 4, pp. 60–69, Dec. 2024, doi: 10.20463/pan.2024.0033.

[4] W.-T. Hsueh, C.-X. Yu, H.-C. Cheng, M.-Y. Chen, H.-M. Wang, and L.-M. Fu, 'A comprehensive review of wearable devices for non-invasive biosensing', TrAC Trends Anal. Chem., vol. 193, p. 118425, Dec. 2025, doi: 10.1016/j.trac.2025.118425.

[5] Y. Zhao et al., 'AI-Enhanced Electrochemical Sensing Systems: A Paradigm Shift for Intelligent Food Safety Monitoring', Biosensors, vol. 15, no. 9, p. 565, Aug. 2025, doi: 10.3390/bios15090565.

[6] Y. Baek et al., 'Edge intelligence through in-sensor and near-sensor computing for the artificial intelligence of things', *Npj Unconv. Comput.*, vol. 2, no. 1, p. 25, Oct. 2025, doi: 10.1038/s44335-025-00040-6.

[7] G. Matsumura et al., 'Real-time personal healthcare data analysis using edge computing for multimodal wearable sensors', Device, vol. 3, no. 2, Feb. 2025, doi: 10.1016/j.device.2024.100597.

[8] T. Meuser et al., 'Revisiting Edge AI: Opportunities and Challenges', IEEE Internet Comput., vol. 28, no. 4, pp. 49–59, July 2024, doi: 10.1109/MIC.2024.3383758.

[9] Z. Liu and K. Huang, 'Semantic-Relevance Based Sensor Selection for Edge-AI Empowered Sensing Systems', Mar. 17, 2025, arXiv: arXiv:2503.12785. doi: 10.48550/arXiv.2503.12785.

[10] E. Song, T. Roth, D. A. Wollman, E. Jordan, M. Serrano, and A. Gyrard, 'Semantics for Enhancing Communications- and Edge-Intelligence-enabled Smart Sensors: A Practical Use Case in Federated Automotive Diagnostics', NIST, Mar. 2025, Accessed: Oct. 25, 2025. [Online]. Available: https://www.nist.gov/publications/semantics-enhancing-communications-and-edge-intelligence-enabled-smart-sensors

[11] X. Wang et al., 'Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models', ACM Comput Surv, vol. 57, no. 9, p. 228:1-228:39, 2025, doi: 10.1145/3724420.

[12] J. D. Velasquez, L. Cadavid, and C. J. Franco, 'Emerging trends and strategic opportunities in tiny machine learning: A comprehensive thematic analysis', *Neurocomputing*, vol. 648, p. 130746, Oct. 2025, doi: 10.1016/j.neucom.2025.130746.

[13] P. Ganesan, Y. R. Thota, H. Shehata, and T. Nikoubin, 'TinyML Based Stress Detection utilizing PPG Signals: A Lightweight Approach for Smart Wearable Devices', in Proceedings of the Great Lakes Symposium on VLSI 2025, in GLSVLSI '25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 941–946. doi: 10.1145/3716368.3735274.

[14] S. Heydari and Q. H. Mahmoud, 'Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future

Directions', Sensors, vol. 25, no. 10, p. 3191, Jan. 2025, doi: 10.3390/s25103191.

[15] T. Suwannaphong, F. Jovan, I. Craddock, and R. McConville, 'Optimising TinyML with quantization and distillation of transformer and mamba models for indoor localisation on edge devices', Sci. Rep., vol. 15, no. 1, p. 10081, Mar. 2025, doi: 10.1038/s41598-025-94205-9.

[16] S. Sivakumar, J. Logeshwaran, R. Kannadasan, M. Faheem, and D. Ravikumar, 'A novel energy optimization framework to enhance the performance of sensor nodes in Industry 4.0', Energy Sci. Eng., vol. 12, no. 3, pp. 835–859, 2024, doi: 10.1002/ese3.1657.

[17] M. Ghorbian, M. Ghobaei-Arani, and L. Esmaeili, 'An energy-conscious scheduling framework for serverless edge computing in IoT', J. Cloud Comput., vol. 14, no. 1, p. 52, Sept. 2025, doi: 10.1186/s13677-025-00780-7.

[18] S. Javaid, H. Fahim, S. Zeadally, and B. He, 'From sensing to energy savings: A comprehensive survey on integrating emerging technologies for energy efficiency in WBANs', Digit. Commun. Netw., vol. 11, no. 4, pp. 937–960, Aug. 2025, doi: 10.1016/j.dcan.2024.11.012.

[19] R. Kesavan, Y. Calpakkam, P. Kanagaraj, and V. Loganathan, 'Secured energy optimization of wireless sensor nodes on edge computing platform using hybrid data aggregation scheme and Q-based reinforcement learning technique', Sustain. Comput. Inform. Syst., vol. 45, p. 101072, Jan. 2025, doi: 10.1016/j.suscom.2024.101072.

[20] D. Sahu et al., 'Optimizing energy and latency in edge computing through a Boltzmann driven Bayesian framework for adaptive resource scheduling', Sci. Rep., vol. 15, no. 1, p. 30452, Aug. 2025, doi: 10.1038/s41598-025-16317-6.

[21] M. Wen et al., 'Deep Reinforcement Learning for Energy-Efficient Workflow Scheduling in Edge Computing', Comput. Netw., p. 111790, Oct. 2025, doi: 10.1016/j.comnet.2025.111790.

[22] Y. Babenko, 'Bioinspired Models for Metaheuristic Optimization', Accessed: Oct. 25, 2025. [Online]. Available: https://www.authorea.com/users/950532/articles/1327804-bioinspired-models-for-metaheuristic-optimization?commit=40908cac03556151fdcea918e3e2557b68482632

[23] M. I. Pavel, S. Hu, M. Pratama, and R. Kowalczyk, 'Onboard Optimization and Learning: A Survey', May 07, 2025, arXiv: arXiv:2505.08793. doi: 10.48550/arXiv.2505.08793.

Proceedings Paper DOI: 10.58190/icisna.2025.143

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Automated Quality Control in Welding Processes Using YOLOv5 and YOLOv8

Adem DİLBAZ[1], İlker Ali ÖZKAN[2]

[1]*Department of Mechatronics Engineering, Selcuk University, Konya, Turkiye*
ademdilbaz25@gmail.com, ORCID: 0000-0002-3135-7032

[2]*Department of Computer Engineering, Selcuk University, Konya, Turkiye*
ilkerozkan@selcuk.edu.tr, ORCID: 0000-0002-5715-1040

***Abstract*— This paper presents a comparative evaluation of YOLOv5 and YOLOv8 object detection models for automated quality control in industrial welding applications. Publicly available welding defect datasets obtained from Kaggle were used, consisting of geometry, structural, and surface defect classes. The dataset was divided into training, validation, and test sets, and all models were trained under identical hyperparameters to ensure a fair comparison. Six YOLO variants—YOLOv5n, YOLOv5s, YOLOv5m, YOLOv8n, YOLOv8s, and YOLOv8m—were evaluated with data augmentation strategies enabled and disabled. Performance was assessed using F1-score and confidence score (CS) metrics on a test set of 75 images. Experimental results demonstrate that data augmentation significantly improves detection performance across all model scales, increasing F1-scores while simultaneously reducing mean confidence scores, which indicates improved model calibration and reduced overconfidence. Furthermore, both YOLOv5 and YOLOv8 architectures demonstrated highly competitive performance, with the medium-scale YOLOv5m achieves the highest F1-score of 0.824, followed closely by YOLOv8m. These findings indicate that modern YOLO architectures provide robust and generalized solutions for real-time welding defect detection tasks, making them well suited for industrial inspection systems.**

***Keywords*— Automated quality control, Data augmentation, Deep learning, YOLOv5, YOLOv8, Welding defect detection**

I. INTRODUCTION

Industrial welding is one of the most widely used joining techniques in manufacturing sectors such as automotive, shipbuilding, construction, and heavy industry. It enables the permanent joining of metal components by applying heat, pressure, or both, thereby ensuring structural integrity and load-bearing capability. The quality of welded joints directly affects mechanical strength, structural durability, fatigue life, and operational safety of manufactured products. Consequently, defects occurring during welding processes may lead to severe economic losses, safety risks, and reduced service life, making effective quality control mechanisms an indispensable part of industrial welding operations [1].

Conventional welding inspection techniques, including visual inspection and non-destructive testing (NDT) methods such as ultrasonic testing, radiographic testing, and magnetic particle inspection, are commonly used in industrial practice. Although these methods are effective, they rely heavily on expert judgment and manual effort, which makes them time-consuming, subjective, and prone to human error. Furthermore, their implementation on high-speed or large-scale production lines is often limited due to inspection costs, processing time, and the need for skilled personnel [2]. As a result, traditional inspection techniques struggle to meet the increasing demands of modern automated manufacturing systems.

In recent years, automated quality control systems based on computer vision and artificial intelligence have emerged as a powerful alternative to conventional inspection approaches. By integrating industrial cameras with image processing algorithms, welding seams can be monitored continuously, allowing defects such as cracks, porosity, lack of fusion, and surface irregularities to be detected in real time [3]. These systems improve inspection consistency, reduce dependency on human operators, and enable faster and more reliable decision-making in production processes.

Recent advances in deep learning, particularly convolutional neural networks (CNNs), have significantly enhanced object detection and classification performance in complex industrial environments. CNN-based models are capable of automatically learning hierarchical feature representations from raw images, making them highly suitable for defect detection tasks [4]. Among various deep learning–based detection approaches, YOLO (You Only Look Once) models have gained widespread attention due to their single-stage architecture, which allows simultaneous localization and classification of objects with high accuracy and real-time inference capability [5].

Successive versions of the YOLO architecture, including YOLOv5 and YOLOv8, have introduced architectural improvements such as anchor-free detection heads, optimized feature extraction strategies, and reduced computational complexity. These improvements enhance detection robustness, generalization ability, and computational efficiency, making YOLO-based models well suited for real-time industrial inspection applications [6,7].

In parallel with algorithmic advancements, cloud-based training infrastructures have become increasingly important for deep learning applications. Platforms such as Google COLAB provide access to high-performance GPU resources, significantly reducing model training time and lowering hardware cost barriers for researchers and practitioners [8]. In this paper, the effectiveness of different YOLO versions for automated welding defect detection is investigated using a cloud-based training environment, with particular emphasis on data augmentation(Aug) strategies and comparative model performance.

II. MATERIAL AND METHOD

A. MATERIALS

The datasets used in this paper were obtained from publicly available sources such as Kaggle, which provide labeled industrial welding defect images suitable for deep learning–based object detection tasks. These platforms are widely adopted in the research community due to their ease of access, standardized annotation formats, and compatibility with modern deep learning frameworks, including YOLO-based architectures [9,10]. The collected datasets represent realistic welding conditions and common defect types encountered in industrial production environments.

The dataset consists of three main defect categories, each corresponding to a specific type of welding imperfection. The geometry class includes defects related to weld bead shape and dimensional inconsistencies, such as undercut and excessive reinforcement, and contains 168 images. The structural class represents internal or load-bearing defects that directly affect the mechanical strength of the joint, such as lack of fusion or incomplete penetration, and consists of 163 images. The surface class includes visible surface-level defects such as cracks, porosity, and spatter, comprising 332 images. This class-based distribution allows the evaluation of model robustness across visually diverse defect types with varying levels of complexity.

Fig. 1 Representative samples from the Kaggle welding defect dataset

To ensure reliable training and unbiased performance evaluation, the dataset, for which representative samples are illustrated in Figure 1, was divided into three subsets: training, validation, and testing. The training set is used to optimize the model parameters by learning representative features from labeled images. The validation set is employed to monitor model performance during training, enabling hyperparameter tuning and early detection of overfitting. The test set is reserved exclusively for final evaluation, providing an objective assessment of the model's generalization capability on unseen data. This separation is a standard practice in deep learning research and is essential to prevent data leakage and ensure reproducible results [11].

Fig. 2 Labelling samples from the Kaggle welding defect dataset.

Each defect in the dataset was annotated with bounding boxes, visually highlighted in red in Fig. 2. All images were labeled using the YOLO annotation format, where each image is paired with a corresponding text file containing the

bounding box coordinates and class information. Each label file consists of five numerical components for every annotated object. The first value represents the class identifier (class ID), where 0, 1, and 2 correspond to surface, structural, and geometry defects, respectively. The remaining four values define the bounding box parameters: x-center, y-center, width, and height. These values are normalized with respect to the image dimensions and expressed as ratios between 0 and 1, allowing scale invariance across different image resolutions [12].

The five labeling parameters define the bounding box: the horizontal and vertical centers specify the normalized coordinates of the box's center point, while the width and height represent its normalized size. Because these values are expressed as ratios relative to the image dimensions, they provide scale invariance across different resolutions and form the basis for subsequent evaluation metrics such as Intersection over Union (IoU), which quantifies the overlap between predicted and ground-truth bounding boxes. This annotation strategy enables YOLO models to perform localization and classification simultaneously within a unified learning framework. Accurate and consistent labeling is particularly critical in welding defect detection, as defects often occupy small regions and exhibit subtle visual variations that can significantly affect detection performance [13].

Model training was conducted using a cloud-based Google COLAB environment equipped with an NVIDIA L4 GPU with 22 GB of memory, which significantly reduced training time and enabled efficient experimentation. Additionally, an Intel i5-12450 processor with 8 cores was used for data preprocessing and auxiliary testing tasks. Training YOLO models solely on CPU resources or low-end GPUs would result in excessively long training times, especially when large datasets, data augmentation techniques, and multiple training epochs are involved. Therefore, cloud-based GPU acceleration is considered essential for practical and scalable deep learning experimentation in this paper [14].

B. METHODS

In this paper, YOLO (You Only Look Once)-based models — specifically YOLOv5n, YOLOv5s, YOLOv5m, YOLOv8n,YOLOv8s, and YOLOv8m — were systematically implemented and optimized using state-of-the-art deep learning techniques to achieve higher accuracy even on low-resolution weld images, and to ensure reliable performance on real-world data despite limited training samples. YOLOv5 employs an anchor-based detection mechanism, where bounding boxes are predicted relative to predefined anchor boxes. By contrast, YOLOv8 introduces an anchor-free detection head that directly predicts bounding boxes without the need for anchors, simplifying the training pipeline and improving generalization across objects with varying shapes and aspect ratios [15]. YOLOv8 and subsequent versions further advance this paradigm, maintaining anchor-free detection and incorporating architectural optimizations such as refined backbone modules and enhanced feature processing, thereby representing one of the most advanced and optimized detection pathways in the YOLO series [16].

In YOLO architectures, Input refers to the raw image data fed into the network for object detection. The Backbone is the feature extractor that generates hierarchical feature maps from the input (e.g., CSPDarknet, C2f variants) [17]. The Neck consists of intermediate layers that aggregate and refine multiscale features (e.g., PANet, FPN), enhancing the model's ability to detect objects at different scales, while the Head predicts bounding box coordinates, object classes, and confidence scores [18]. In YOLOv8, multi-scale feature processing traditionally handled in the Neck is efficiently integrated with the Backbone, enhancing feature extraction while minimizing redundant computations, as illustrated in Fig. 3, which compares the structure with YOLOv5[19]. Processing the input image through successive downsampling stages, the model architecture extracts hierarchical features and ultimately produces three distinct output branches at the head. These branches correspond to multi-scale feature maps representing different levels of abstraction: high-resolution features for detecting small objects, medium-resolution features for mid-sized objects, and low-resolution, semantically rich features for large objects. This design enables effective object detection across varying object sizes by leveraging hierarchical feature representations.

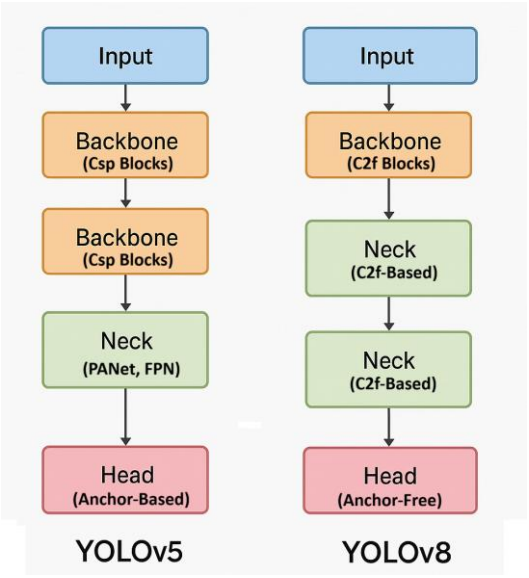


Fig. 3 Comparative YOLOv5 and YOLOv8 structures.

All models in this paper were trained using the same hyperparameters to ensure fair comparison. Training used stochastic gradient descent (SGD) with learning rate 0.01, momentum 0.937, and weight decay 0.0005. Models were trained for up to 100 epochs with early stopping (patience=15), using a batch size of 16 and input images resized to 640 × 640 pixels. To improve reproducibility, a fixed random seed (42) was used across all experimental runs [20].

To mitigate overfitting despite rapid loss reduction, extensive augmentation strategies were applied. YOLO's built-in data augmentation techniques — including Mosaic, MixUp, and Flip — were activated during training to synthetically increase data diversity and improve model robustness [21]. Industrial welding datasets from platforms such as Kaggle provide annotated images of weld seams with bounding boxes for defect classes [22], while tools like Roboflow and Google Colab facilitate dataset management, augmentation, and export in formats compatible with YOLO frameworks [23]. To analyze the effect of data augmentation, experiments were conducted under two settings: (i) augmentation enabled using the default YOLO training augmentations (e.g., Mosaic, MixUp, flipping, and color transformations), and (ii) augmentation disabled by turning off these transformations, ensuring training on only the original images.

Performance was evaluated using Precision, Recall, F1 score, and mean Average Precision (mAP). The F1 score is the harmonic mean of Precision and Recall, capturing the balance between false positives and false negatives. Precision measures the proportion of correct positive predictions, and Recall measures the proportion of actual positives correctly detected. mAP50 reports AP at IoU = 0.5, while mAP50–95 averages AP across IoU thresholds from 0.5 to 0.95. In this study, the Confidence Score (CS) is defined as the mean detection confidence of all final predicted boxes after non-maximum suppression on the test set (conf = 0.25, IoU = 0.7), reflecting the average confidence level of the model's detections [24].

III. TEST RESULTS

In this first experimental phase, the detection performance of YOLOv5 architectures—specifically YOLOv5n, YOLOv5s, and YOLOv5m—was evaluated on the reserved test set of 75 images. The study aimed to assess how model complexity and data augmentation strategies influence detection accuracy in industrial welding scenarios. Table I presents a comprehensive comparison of key performance metrics, including Precision, Recall, F1-score, mAP values, and Mean Confidence Score (CS), obtained with and without data augmentation.

In this second test, as shown in Fig. 4**,** the orange curve represents the *structural class*, while the dark blue curve corresponds to the *all-classes* configuration during the training process. The results indicate a consistent performance improvement as the model complexity increases. Specifically, as the number of model parameters grows, the network's representational capacity improves, leading to better learning of both individual feature classes and their combined representation. This trend suggests that larger models are more effective at capturing complex structural, geometric, and surface-level patterns present in the data. In this test, the YOLOv5s model, which has approximately 9 million parameters, achieves the highest mAP50 (0.8082) when augmentation is enabled (Table I).

In the third test, the YOLOv8 architectures—specifically YOLOv8n, YOLOv8s, and YOLOv8m—were evaluated on the same test set of 75 images to provide a direct comparison with the anchor-based YOLOv5 models. This test aimed to assess the efficacy of YOLOv8's anchor-free head and advanced feature extraction modules in industrial welding defect detection. Table II summarizes the detailed performance metrics, including Precision, Recall, F1-score, mAP values, and Mean Confidence Score (CS), obtained with and without data augmentation. In the last test, Fig. 5 depicts the comparative F1–Confidence curves of the YOLOv8 models obtained during the training process. In this test, where data augmentation is enabled, the YOLOv8m model achieves the highest performance. Owing to its larger model capacity, characterized by approximately 26 million parameters, the all-classes configuration benefits the most from this increased representational capability.

The results indicate that data augmentation plays a pivotal role in scaling model performance. When augmentation is enabled, detection accuracy generally improves as model complexity increases, with the medium-scale models achieving the highest F1-scores.

TABLE I. COMPARATIVE PERFORMANCE OF YOLOV5 MODELS UNDER DIFFERENT DATA AUGMENTATION SETTINGS

Model	Augmentation	Precision	Recall	F1 Score	mAP50	mAP50-95	CS (Mean)
YOLOv5n	Off	0.7179	0.7160	0.7149	0.6730	0.3474	0.8295
YOLOv5n	On	0.8119	0.7643	0.7868	0.7942	0.4378	0.6570
YOLOv5s	Off	0.7445	0.7129	0.7275	0.7095	0.3742	0.8588
YOLOv5s	On	0.8248	0.8033	0.8131	**0.8082**	**0.4380**	0.6627
YOLOv5m	Off	0.7644	0.7252	0.7416	0.7118	0.3840	0.8483
YOLOv5m	On	**0.8424**	**0.8069**	**0.8242**	0.8002	0.4360	0.6882

TABLE II. COMPARATIVE PERFORMANCE OF YOLOV8 MODELS UNDER DIFFERENT DATA AUGMENTATION SETTINGS

Model	Augmentation	Precision	Recall	F1 Score	mAP50	mAP50-95	CS (Mean)
YOLOv8n	Off	0.7504	0.7221	0.7341	0.6972	0.3521	0.8624
YOLOv8n	On	0.8012	0.7578	0.7780	0.7794	0.4202	0.6395
YOLOv8s	Off	0.7653	0.7890	0.7740	0.7398	0.3986	0.8682
YOLOv8s	On	**0.8400**	0.7984	0.8184	0.7923	0.4268	0.6613
YOLOv8m	Off	0.7421	0.7360	0.7359	0.7255	0.3817	0.8269
YOLOv8m	On	0.8035	**0.8416**	**0.8205**	**0.7971**	**0.4337**	0.7270

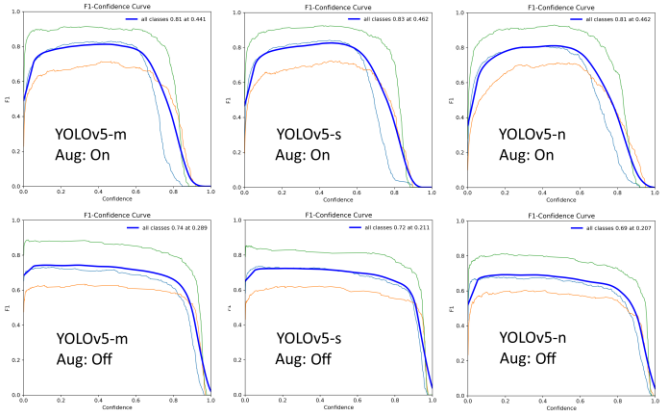


Fig. 4 Comparative F1-confidence curves of YOLOv5 models under different data augmentation settings

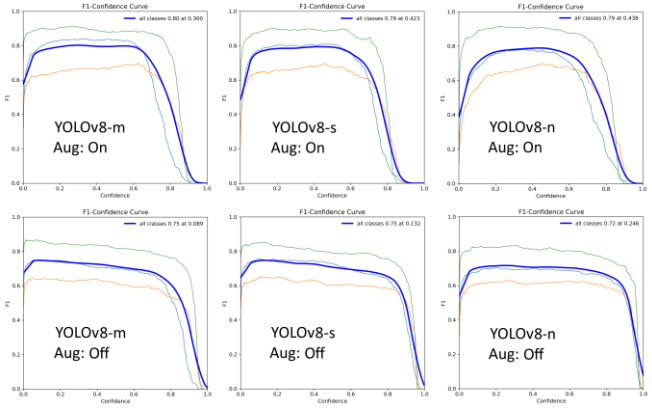


Fig. 5 Comparative F1-confidence curves of YOLOv8 models under different data augmentation settings

However, under non-augmented conditions, a deviation from this trend was observed in the YOLOv8 series; the YOLOv8s model (F1=0.7740) outperformed the larger YOLOv8m (F1=0.7359). This suggests that without the diversity provided by augmentation, larger models may be more prone to overfitting or struggle to generalize on limited datasets, whereas the 'Small' architecture offers a more efficient balance for this specific data scale.

Specifically, as the number of model parameters grows, the network's representational capacity improves, leading to better learning of both individual feature classes and their combined representation. This trend suggests that larger models are more effective at capturing complex structural, geometric, and surface-level patterns present in the data. In this test too, the M-version YOLO model, which has the largest number of parameters, achieves the highest performance, with the all-classes configuration benefiting the most from the increased model capacity.

IV. CONCLUSIONS

In this paper, the performance of YOLOv5 and YOLOv8 models at different scales was systematically analyzed for automated welding defect detection under identical training conditions. The experimental results clearly demonstrate that data augmentation plays a crucial role in improving detection accuracy and prediction confidence across all model variants.

When comparing the lightweight models (YOLOv5n and YOLOv8n) with augmentation enabled, YOLOv5n achieved a slightly higher F1-score of 0.79 compared to YOLOv8n (0.78). Interestingly, both models showed a significant decrease in Mean Confidence Scores (CS) when augmentation was applied (dropping from ~0.86 to ~0.65). This reduction indicates that augmentation mitigates overconfidence, preventing the models from memorizing easy samples and resulting in more realistic uncertainty estimation for complex defects.

For the small-scale models under augmentation-enabled conditions, YOLOv8s outperformed its counterpart with an F1-score of 0.82, slightly surpassing YOLOv5s (0.81). This result highlights the advantage of YOLOv8's anchor-free detection head in capturing defect features more effectively in mid-range complexity, offering a strong balance between accuracy and computational efficiency.

In the medium-scale comparison, the YOLOv5m model achieved the highest overall performance in this study with an F1-score of 0.824, marginally outperforming YOLOv8m (0.820). Notably, both medium models exhibited robust generalization with high mAP50-95 values (~0.43), indicating that at higher model capacities, the architectural differences between anchor-based (v5) and anchor-free (v8) approaches yield comparable high-accuracy results for industrial welding inspection.

Overall, the results show that while both YOLOv5 and YOLOv8 models benefit significantly from data augmentation, their confidence behaviors differ across scales. While the YOLOv8m model produced higher confidence scores (CS=0.727) than YOLOv5m (CS=0.688) under augmented conditions, the lightweight YOLOv5 variants (Nano and Small) maintained slightly higher or comparable confidence levels to their YOLOv8 counterparts. This indicates that while YOLOv8's anchor-free head is highly effective, it does not essentially guarantee higher prediction confidence in every configuration. Nevertheless, YOLOv8 remains a strong competitor for real-time industrial welding inspection systems due to its architectural efficiency and competitive accuracy. Future work will focus on optimizing inference speed on edge devices and extending the evaluation to higher-resolution datasets in real-time production environments.

V. REFERENCES

[1] ANDERSSON, J., "Welding metallurgy and weldability of superalloys", *Metals*, vol. 10 pp. 143, 2020.

[2] Singh, R. R., Introduction to NDE 4.0., Handbook of Nondestructive Evaluation 4.0, Cham, Switzerland, Springer, 2025.

[3] Amarnath, M., Sudharshan, N., Srinivas, P., "Automatic detection of defects in welding using deep learning-a systematic review", *Materials Today: Proceedings*, 2023.

[4] Lecun, Y., Bengio, Y., Hinton, G. "Deep learning", *Nature*, vol.521. pp. 436-444, 2015.

[5] Chen, J., Zheng, Y., Zhang, L., Wang, M., Gai, F., Li, C., & Shen, Y., "The design and implementation of the kernel level mobile storage medium data protection system", *In Proc. 2013 IEEE International Conference on Granular Computing (GrC)*, p. 53-57, 2013.

[6] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., and Jain, M., Ultralytics/yolov5: v7.0-yolov5 SOTA realtime instance segmentation, Zenodo, Switzerland, 2022

[7] Wang, C.Y., Bochkovskiy, A., Liao, H.-Y. M., *"*YOLOv7: Trainable *Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 7464-7475, 2022.

[8] Bisong, E., Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59-64.* Berkeley, USA, CA: Apress, *2019.*

[9] Kaggle, "Welding defect dataset," Kaggle Platform, [Online], https://www.kaggle.com/datasets/sukmaadhiwijaya/welding-defect-object-detection, 2020.

[10] Toropov, E., Buitrago, P. A., Moura, J. M., *"*Shuffler: A Large-scale Data Management Tool for Machine Learning in Computer Vision", In Proceedings of the Practice and Experience in Advanced Research Computing (PEARC) Conference, pp. 1–8, 2019.

[11] Wei, K., *Evaluating Machine Learning Approaches for Predicting Customer Conversion in Direct Marketing Campaigns: An Empirical Study Using the Bank Marketing Dataset*. Diss. UCLA, 2025.

[12] Khanam, R., Hussain, M., "What is YOLOv5: A deep look into the internal features of the popular object detector", *In Proc. arXiv preprint arXiv:2407.20892*, 2024.

[13] Ma, Y., Yin, J., Huang, F., & Li, Q., "Surface defect inspection of industrial products with object detection deep networks: A systematic review", *Artificial Intelligence Review*, vol. 57, pp. 333, 2024.

[14] Ciaburro, G., Ayyadevara, V. K., and Perrier, A., *Hands-On Machine Learning on Google Cloud Platform: Implementing Smart and Efficient Analytics Using Cloud ML Engine*, Birmingham, UK: Packt Publishing Ltd., 2018

[15] Sapkota, R., & Karkee, M., "Ultralytics YOLO Evolution: An Overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 Object Detectors for Computer Vision and Pattern Recognition", *In Proc.* arXiv preprint arXiv:2510.09653, 2025

[16] Terven, J., Córdova Esparza, D. M., & Romero González, J. A., "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO NAS", *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023

[17] Li, Z., Wang, Y., Han, M., & Zheng, Z., "BS-YOLOv8n: An Improved YOLOv8n Network for Tomato Detection at Different Ripeness Degrees in Complex Greenhouse Environments", Academic Journal of Agriculture & Life Sciences, vol. 6, pp. 130-136, 2025.

[18] Oksuz, K., Cam, B. C., Akbas, E., & Kalkan, S., "Localization Recall Precision (LRP): A New Performance Metric for Object Detection", *In Proc.* arXiv preprint arXiv:1807.01696, 2018.

[19] Zou, Z., Shi, Z., Guo, Y., & Ye, J., "Object detection in 20 years: A survey", International Journal of Computer Vision, vol. 127, pp. 74–109, 2019.

[20] Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A., "Evaluating the evolution of YOLO (You Only Look Once) models: A comprehensive benchmark study of YOLO11 and its predecessors", *In Proc.* arXiv preprint arXiv:2411.00201, 2024.

[21] Padilla, R., Passos, W. L. B., da Silva, E. A. B., & Netto, S. L., "A comparative analysis of object detection metrics with a companion open source toolkit", Electronics, vol. 10, pp. 279, 2021.

[22] Asghar, T., Khanam, R., Hussain, M., "Comparative Performance Evaluation of YOLOv5, YOLOv8, and YOLOv11 for Solar Panel Defect Detection," Solar, vol. 5, no. 1, pp. 1–25, 2025.

[23] Zhang, D., Zheng, S., & Jiao, L., "Weld defect detection in digital radiographic images: A review of automatic technologies," NDT & E International, vol. 122, 2021.

[24] Garg, S., & Jalal, A., "The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection," Computers, vol. 13, pp. 336, 2024

Proceedings Paper DOI: 10.58190/icisna.2025.144

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Secure Voting System Using FPGA

Kiruthick K M[1], Rushindra K R[2], Karthikeyan T[3], Kaveri Hatti[4]

[1-4] *Department of Electronics and Communication Engineering*
Amrita School of Engineering, Bengaluru – 560035
Amrita Vishwa Vidyapeetham, India

[1]*bl.en.u4ece23221@bl.studenta.amrita.edu, ORCID: 0009-0009-8256-6561*
[2]*bl.en.u4ece23140@bl.studenta.amrita.edu, ORCID: 0009-0009-8143-5105*
[3]*bl.en.u4ece23220@bl.studenta.amrita.edu, ORCID: 0009-0007-0815-8004*
[4]*h_kaveri@blr.amrita.edu, ORCID: 0000-0002-8973-5534*

***Abstract*— The project is a proposal of a secure and reliable electronic voting system that will be made by the use of ESP32 microcontroller, keypad input module, and LCD interface. It mainly aims to make the process of voting transparent, accurate, and secure through the verification and management of the data in the clouds. Voter identification is done through the Aadhaar based credentials with password based login, providing a highly effective two-factor authentication system. The system will ensure that no votes are counted twice, as each vote will be associated with a specific Aadhaar ID and always have real-time status updates stored on the cloud. Once verified, the votes are encrypted with the aid of hardware-supported encryption chip executed on FPGA and sent safely to a cloud database such that the stored data cannot be easily changed, and can only be accessed via the authorized medium. To boost confidence among the voters, an SMS or OTP message of confirmation is automatically sent to the voter when a vote has been successfully registered. The proposed system provides end-to-end security, endures the privacy of the voter, and real-time vote counting is accurate. This solution is a modern solution to use in place of traditional methods of voting by use of IoT technology and secure cloud services to have a scalable, modern, and reliable approach to the same.**

***Keywords*—Electronic voting system, ESP32, Aadhaar verification, Cloud authentication, IoT security, Encrypted voting, Duplication vote prevention, OTP confirmation, Secure data storage.**

I. INTRODUCTION

The use of electronic systems of voting is crucial to the contemporary states of democracy, but the issue of security and reliability remains. Most of the current systems also do not have a good voter authentication, data integrity and privacy. As more people go digital, elections need to have secure, transparent and efficient voting provisions that resist fraud without compromising of voter confidence and reliability [1]–[4].

Voter impersonation is a significant problem that exists in the contemporary systems of voting because of the poor check of identity. Tampering of data also poses an additional challenge to the integrity of an election, as it allows a manipulation of stored or transmitted votes. Also, the absence of the vote confidentiality weakens the privacy of voters, which may affect voter turnout and compromise the election results credibility [2], [3], [5]. In order to overcome these problems, this paper will suggest a secure FPGA based-voting system, which uses AES encryption, password and Aadhaar-based authentication to secure confidentiality.

The rest of the paper is structured in the following way: Section II entails the literature survey, Section III details the methodology, Section IV addresses the implementation and results obtained, and Section V is a concluding part of the paper [1], [2], [4].

II. LITERATURE SURVEY

The recently provided research works have taken much attention on enhancing the security and reliability of electronic voting systems by employing multi-factor authentication and cryptographic security. Voter impersonation is greatly mitigated and better voter confidence is ensured due to the ability of biometric based voting systems to be undertaken using fingerprint authentication and SMS confirmation mechanism [1], [3]. Authentication is further enhanced with Aadhaar based identity verificationto ensure that there is no voting twice and unauthorized channel access [5].

Most of the researches have embraced cryptographic methods to guarantee confidentiality of votes and data integrity through the application of the Advanced encryption Standard (AES) coupled with OTP validation, which affords a high level of resistance against data corruption in transit and storage [2].

New voting systems combining machine learning, blockchain, and cryptography enhance the transparency, decentralization, and auditability of the voting systems [4], [6]. FPGA and VLSI-based AES, LFSR, and hybrid encryption algorithms provide hardware-oriented security solutions where high-speed and low power consumption software is suitable as well as differentiating a secure environment on real-time voting systems [7] to [9]. The technical research and documentation also attest to the essence of authenticated encryption, scalable system architecture, and key security in terms of reliable electronic voting systems [10], [11].

III. Methodology

A. *Single-Ward Voting, Authentication, and Vote Casting Architecture*

The voter enters 12-digit Aadhaar number first. Then the password is entered. Both values are checked with the voter data stored in the cloud. If the details are correct, the system allows the voter to continue. If the details are wrong, the voting process is stopped. After verification, the system shows the candidate names on the display and the voter selects one candidate using the keypad.

After candidate selection, the system checks the vote status of the voter. This status is already stored in the database. If the voter has voted earlier, the system does not allow voting again. If not voted, an OTP is sent to the registered mobile number. The vote is accepted only after the correct OTP is entered.

If the OTP entered is correct ,then the system sends back another SMS to the registered mobile number stating that the vote is successfully saved. After this process the votes come to the next block , the encryption module.Fig. 1. illustrates the functional flow of the proposed electronic voting system for a single voting booth or ward.

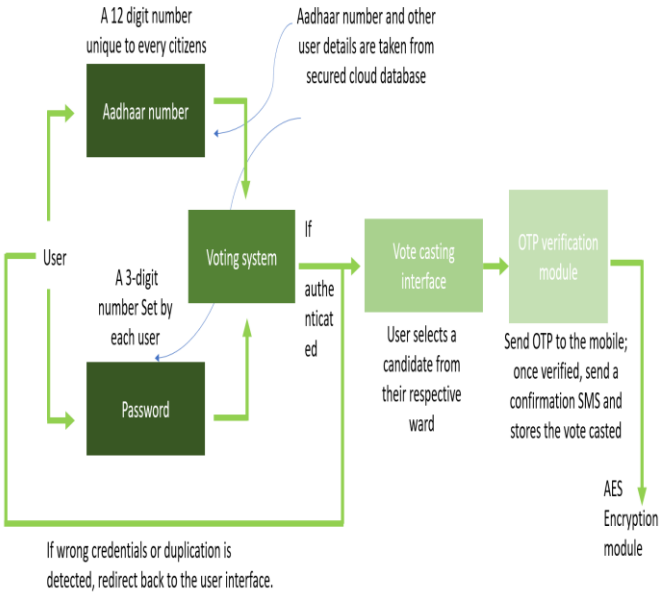


Fig. 1 Authentication and Vote Casting Flow of the Proposed Voting System

B. *Secure Vote Encryption, Decryption, and Result Processing Architecture*

The vote from the voter is taken and added with other votes, then the votes are collected together. The collected data is prepared for storage. This step is done before saving the vote.

After this, the vote data is encrypted using the AES algorithm. The encrypted data is stored. The data cannot be read directly. Only authorized persons who have the correct key can decrypt the data and see the result. This keeps the voting result safe. Fig. 2 shows what happens to the vote after it is selected.

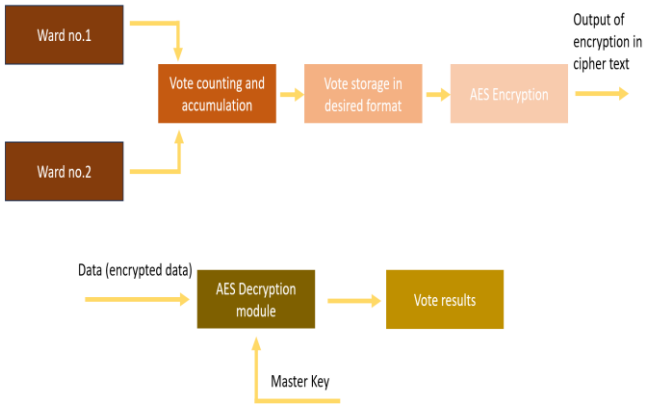


Fig. 2 Vote Accumulation, AES Encryption, and Result Decryption Process

IV. Implementation and Results

A. *End-to-End Voting System in Wokwi Simulator*

The ESP32 is used as the main controller. A 4X4keypad is used for input and an 16X2 LCD is used for display. The ESP32 connects to the cloud using Wi-Fi. This setup is used to test the working of the voting process. Fig. 3 shows the simulation setup of the voting system.

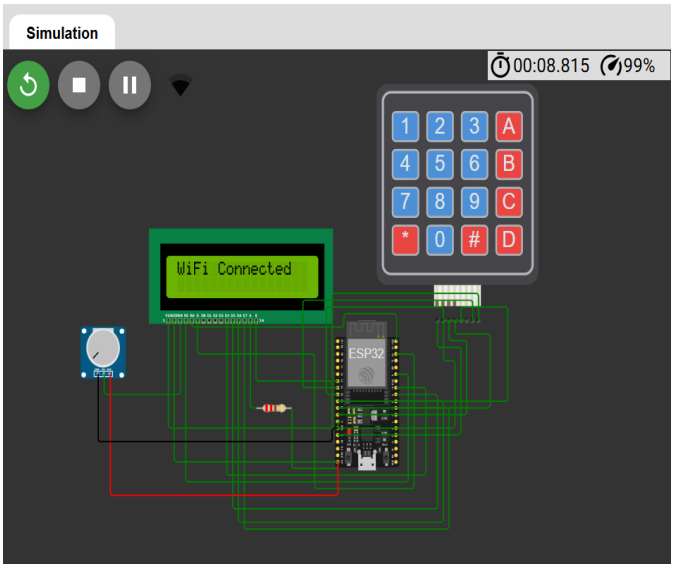


Fig. 3 ESP32-Based Hardware Simulation Setup for Secure Voting System

The voter enters the Aadhaar number using the keypad. The entered number is displayed on the LCD. This Aadhaar number is used to identify the voter. The system uses this value to check the voter details from the cloud. Fig. 4 shows the Aadhaar number entry stage.

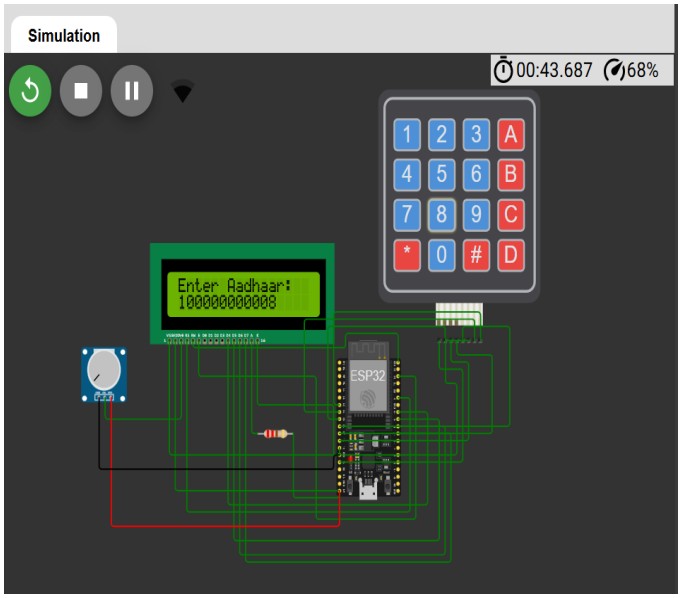


Fig. 4 Aadhaar Number Entry Stage

After Aadhaar entry, the voter enters the password. The password is entered using the keypad. The system checks the password along with Aadhaar details. If the password is correct, the voter is allowed to continue. These details are validated from the information which is stored in cloud (google spreadsheet) Fig. 5 shows the password entry stage.

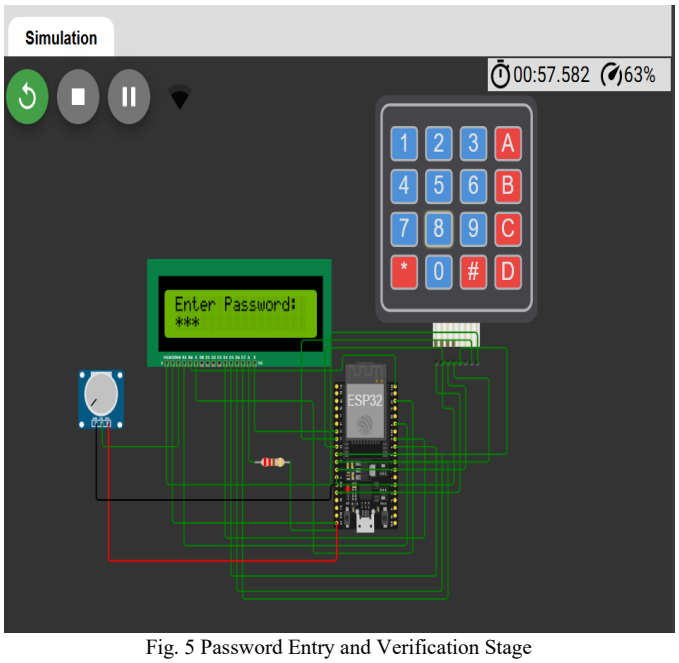


Fig. 5 Password Entry and Verification Stage

This Aadhaar and password based login prevents the voter impersonation and voter duplication as the system come again into the user login interface if the details entered is not correct or the person is trying to cast the vote again. After verification, the system displays the candidate names. The voter selects one candidate using the keypad. This is the voting stage where the vote is chosen. Fig. 6 shows the candidate selection screen.

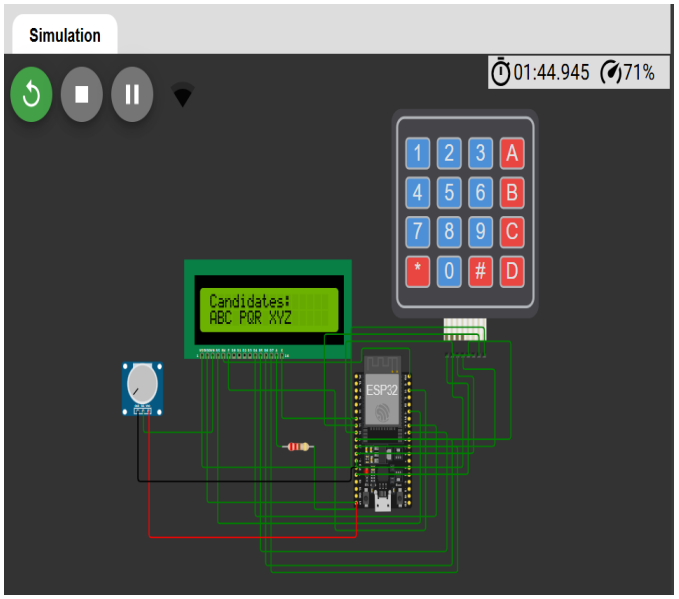


Fig. 6 Candidate Selection Display

The system shows that the vote is selected. This confirms that the input is received correctly before moving to the next step. After the vote confirmation a OTP is send to the mobile number, the voter is supposed to enter the OTP into the system only after which their vote will be saved. Fig. 7 shows the confirmation message after selecting the candidate.

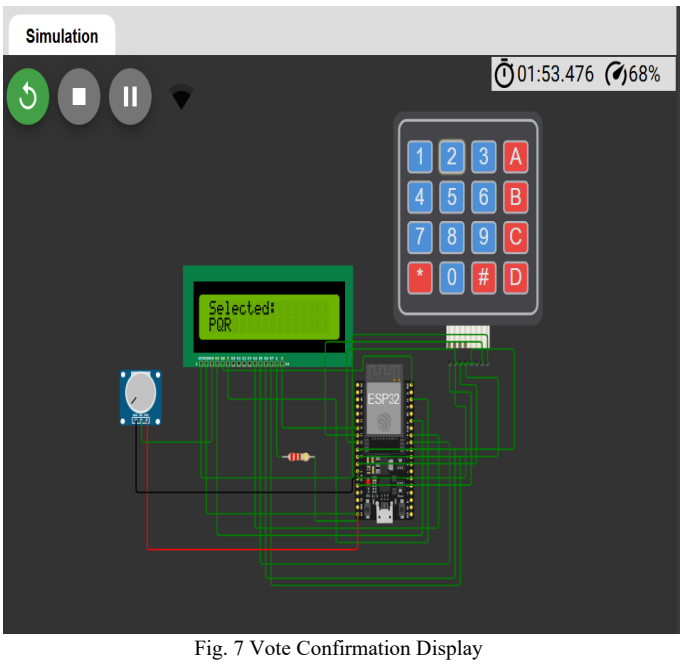


Fig. 7 Vote Confirmation Display

After the OTP the voter will again get an SMS as an confirmation message, it states the vote is successfully saved.

B. Wokwi–Spreadsheet Integration

The URL generated by the ESP32 is pasted into the program. The program sends the request to the cloud spreadsheet. The voter details such as name, phone number, and vote status are returned. The message shows that the voter is found and authenticated. Fig. 8. Cloud Communication and Data Exchange Using Python Bridge.

```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge.py
Connected to sheet successfully ✅
Paste the Wokwi Request URL below.
Type 'exit' to quit.

URL> https://script.google.com/macros/s/AKfycbyd7ipl_ytfBMts3guE4clPVGQ
dhar=100000000011&pass=111

✅ Voter Found & Authenticated:
  Name         : Akash Gupta
  Phone        : 9876500011
  Vote Status  : 0
URL>
```

Fig. 8. Cloud Communication and Data Exchange Using Python Bridge

The script URL generated by the ESP32 is pasted into the terminal. The Python program sends a request to the cloud spreadsheet. The response message shows success. This confirms that the voter status is updated correctly in the cloud. Fig. 9 shows the execution of the Python bridge program used to update voter status.

```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge_01.py
Paste the full Wokwi/Apps Script URL below.
Type 'exit' to quit.

URL> https://script.google.com/macros/s/AKfycbyd7ipl_ytfBMts3guE4clPVGQXnw
one=9876500011&vote=0&update=1
✅ Vote update request sent for 9876500011
Response: SUCCESS
URL> https://script.google.com/macros/s/AKfycbwdF5V8eDPExq2yFP0R6BbxYQcfrK
vote=0,0,1
```

Fig.9.Python Bridge Execution for Voter Status Update

The ESP32 URL is given as input to the program. The vote data is received and stored in the cloud. The terminal output shows the recorded vote values. This confirms that the vote is successfully stored. Fig. 10 shows the execution of the Python bridge program for vote recording.

```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge_02.py
Python Voting Bridge Running...
Enter ESP32 URL with vote parameter (e.g., ?vote=1,0,0)
Type 'exit' to stop the program.
Enter URL: https://script.google.com/macros/s/AKfycbwdF5V8eDPExq2yFP0R6BbxYQcfrKU
/exec?vote=0,0,1
Vote recorded: [0, 0, 1]
Enter URL:
```

Fig.10.Python Bridge Execution for Vote Recording

C. Spreadsheet-Based Voter and Result Management

The table has Aadhaar number, name, phone number, password, and vote status. Each voter is stored in one row. The vote status value is zero before voting. After voting, this value is changed. This value is checked to avoid voting again. Fig. 11 shows the voter list stored in the cloud spreadsheet.

	A	B	C	D	E
1	AADHAR	NAME	PHONE	PASSWORD	VOTE STATUS
2	100000000001	Arjun Mehra	9876500001	101	0
3	100000000002	Priya Sharma	9876500002	102	0
4	100000000003	Ravi Kumar	9876500003	103	0
5	100000000004	Sneha Patil	9876500004	104	0
6	100000000005	Kiran Joshi	9876500005	105	0
7	100000000006	Neha Verma	9876500006	106	0
8	100000000007	Rohit Singh	9876500007	107	0
9	100000000008	Anjali Nair	9876500008	108	0
10	100000000009	Vivek Desai	9876500009	109	0
11	100000000010	Meena Iyer	9876500010	110	0
12	100000000011	Akash Gupta	9876500011	111	1
13	100000000012	Divya Reddy	9876500012	112	0
14	100000000013	Manish Bansal	9876500013	113	0
15	100000000014	Ritu Kapoor	9876500014	114	0
16	100000000015	Sandeep Rao	9876500015	115	0

Fig. 11. Cloud-Based Voter Database with Vote Status Update

The columns represent candidates ABC, PQR, and XYZ. Each row represents 1 vote. A value of '1' is entered for the selected candidate. Other values remain '0' (This is to count votes). Fig. 12 shows the vote result sheet stored in the cloud.

	A	B	C	D
1	ABC	PQR	XYZ	
2	1	0	0	
3	0	0	1	
4	0	1	0	
5	0	0	1	
6	0	0	1	
7	0	0	1	
8	0	1	0	
9	0	1	0	
10	1	0	0	
11	1	0	0	
12	0	1	0	
13	1	0	0	
14	0	0	1	

Fig. 12. Cloud Storage of Candidate-Wise Voting Results

D. Secure Encryption and Decryption in Xilinx Vivado

Signals such as clock, reset, start, key, input data, and output data are displayed. The output data value changes after encryption. The ready signal becomes high after encryption is completed.

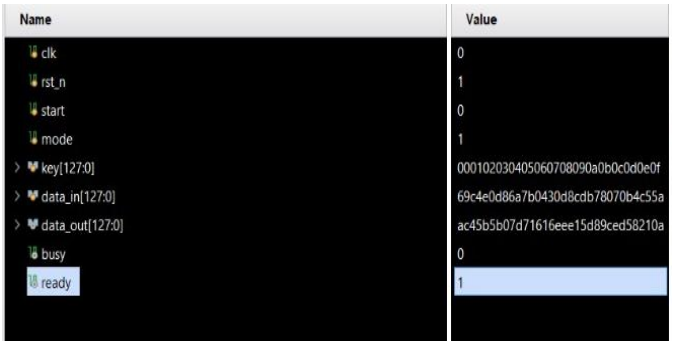


Fig. 13. AES Encryption Simulation values on FPGA

This shows that the AES module finishes the operation correctly. Fig. 13 shows the simulation values of the AES encryption module.

The signals of time along the time axis are the clock and data signals where the input data is fed first and aligned with the clock. The encryption stage entails encryption of an input data with an AES module in the use of a secret key to achieve an output in the form of transformed ciphertext. AES encryption success is assured by a variation in output values. Equally, ciphertext can be decrypted by the same key to obtain the original text and ensure successful and safe functioning. Fig. 14 shows the waveform of the timing of the entire process of encryption and decryption of AES.

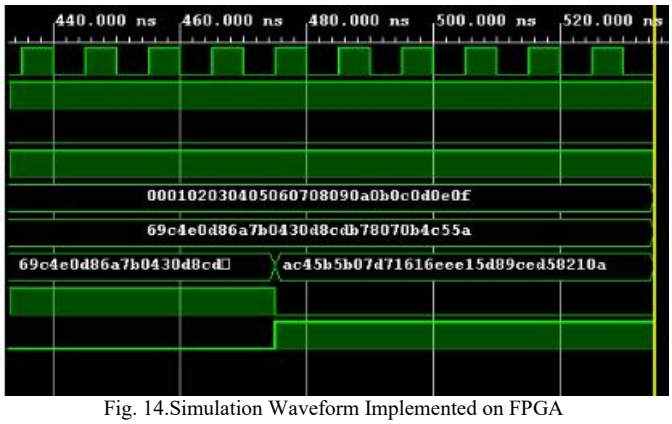


Fig. 14.Simulation Waveform Implemented on FPGA

V. Conclusion

The secure electronic voting system which was presented in this paper aimed at dealing with the critical challenges of voter impersonation, data manipulation and unavailability of vote secrecy. The proposed system will enable the preservation of the integrity and privacy of voting information by promoting the authentication procedure using Aadhaar, the validation of the password, and an encryption tool (AES), which should make sure that a person can cast a vote only in case he/she is an authorized voter. The architecture offers a robust and non-tamperable infrastructure that can be used in the digital elections. The findings reveal that the system can be used to provide increased security and confidence of the voters in the elections without lacking efficiency and scalability.

The future development will concentrate on improving the system by the use of sophisticated authentication and access control protocols. Fingerprint, facial, or iris recognition can be used as a part of the biometric authentication, which will further improve the identification process and enable the voter process the login process with ease. An extra impersonation security layer of using an OTP verification step can be added before voting. Moreover, authorized decryption mechanisms of biometric identification of election requires can provide a limited and safe access of voting outcomes. These upgrades will have the purpose of transforming the proposed system into a totally transparent, scalable, and safe system that can be utilized in large scale state and national elections.

References

[1] M. J, A. Periasamy, D. Kannu, S. T, and T. B, “Secure Electronic Voting System with Fingerprint Authentication and SMS Confirmation," International Journal of Progressive Research in Engineering Management and Science (IJPREMS), vol. 05, no. 04, pp. 2999-3004, Apr. 2025.

[2] V. Natarajan, S. M. S, S. Vaz J, and R. Pathi R, "Online Voting System Using AES Algorithm with OTP Validation," International Research Journal on Advanced Engineering Hub (IRJAEH), vol. 02, no. 02, pp. 57-61, Feb. 2024.

[3] T. Haripriya, Vinodkumar BG, M. Babu, G. Aswini, and R. M S, "Biometric System Based Electronic Voting Machine," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 10, no. 3, pp. 155-160, May-Jun. 2024.

[4] M. Jain, S. Raut, and S. Ghadge, "E-Voting System Using Machine Learning, Blockchain, and Cryptography," International Journal on Science and Technology (IJSAT), vol. 16, no. 2, Apr.-Jun. 2025.

[5] P. G R, S. S, C. N, S. J, and Megha, "Aadhar Based Voting System With Finger Print Authentication," IJSART, vol. 10, no. 12, pp. 26-29, Dec. 2024.

[6] S. Durga, E. Daniel, S. Seetha, and S. Deepakanmani, “Private and Secure Blockchain-Based Mechanism for an Online Voting System,” in *Big Data Innovation for Sustainable Cognitive Computing (BDCC 2021)*, Springer, Cham, Switzerland, 2022, pp. 453–464.

[7] C. Chidanandamrita, B. Ramesh, N. K. Devika, and K. A. Anantha Krishnan, “VLSI Implementation of Crypto Coprocessor Using AES and LFSR,” *Microprocessors and Microsystems*, Elsevier, vol. 88, pp. 1–10, 2022.

[8] C. Chidanandamrita, B. Ramesh, P. Mammen, A. K. N., and S. Sreehari, “Implementation of an Efficient Hybrid Encryption Technique,” *Procedia Computer Science*, Elsevier, vol. 200, pp. 123–130, 2022.

[9] C. Chidanandamrita and G. R. S. Geethu, “Design and Implementation of Reconfigurable Linear Feedback Shift Register,” *Microelectronics Journal*, Elsevier, vol. 127, pp. 1–8, 2022.

[10] TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham, “Research Contributions in Secure and Authenticated Encryption Systems,” Amritapuri Campus, India, Technical Report, 2021.

[11] Amrita Vishwa Vidyapeetham, “Research Trends in Cryptography, Blockchain and Secure Voting Systems,” School of Engineering, Amritapuri Campus, India, Technical Report, 2022.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

SmartCodeHub: LLM-Based Framework for Semantic Code Reuse in Reactive Programming

Aondowase James Orban[1], Ikenna Caesar Nwandu[2]

[1]*Department of Software Engineering, University of Szeged , Dugonics tér 13 , Hungary*
orbanj@inf.u-szeged.hu, ORCID: https://orcid.org/0009-0003-2208-417X

[2]*Department of Software Engineering, Federal University of Technology Owerri, Nigeria*
ikenna.nwandu@futo.edu.ng, ORCID: https://orcid.org/0000-0001-6834-1088

***Abstract*— Code reuse is essential for improving software productivity, yet developers still spend significant effort searching for and re-implementing similar code fragments. Existing snippet management tools rely primarily on keyword-based search, which fails to capture semantic relationships, particularly in reactive and asynchronous programming contexts. This paper presents SmartCodeHub, an AI-assisted snippet management framework that combines semantic code embedding, automated tag generation, and large language model (LLM) reasoning to support contextual code discovery and reuse. SmartCodeHub integrates a searchable snippet library with an interactive retrieval interface and cross-language support. Preliminary evaluation on JavaScript and Python projects indicates improvements in retrieval accuracy and reuse efficiency compared to conventional snippet tools. These early results suggest the feasibility of LLM-enhanced snippet ecosystems and highlight directions for a more broader and reproducible evaluation.**

***Keywords*— Code Reuse, Large Language Models, Software Maintenance, Semantic Search, Reactive Programming**

I. INTRODUCTION

Modern software development increasingly relies on modularity, reuse, and rapid integration of pre-existing code [1-2]. Reusing existing code fragments is a key strategy to reduce development time and improve software reliability[3-4] Developers frequently copy snippets from online repositories or questions and answers (Q\&A) platforms such as Stack Overflow. However, a lack of semantic indexing and poor organizational structures often lead to redundancy, inconsistencies, and subtle integration errors [5-6]. Conventional snippet managers, such as CodeBox or SnippetsLab, depend on manually assigned tags and keyword-based search, which struggle to interpret the underlying intent of a developer's query [7].

Reactive programming frameworks (e.g., RxJS, Reactor) intensify these challenges. Their asynchronous data streams, event-driven architecture, and higher-order functions produce code patterns that are syntactically diverse but semantically equivalent [8-9]. As a result, keyword-based tools frequently fail to retrieve conceptually related snippets, reducing reuse efficiency and increasing cognitive load during debugging and maintenance.

Recent improvements in GPT-4 and Llama 3 have made it possible for Large Language Models (LLMs) to understand both natural language and source code. This is because they are trained on large datasets that include a lot of text and code [10-11]. SmartCodeHub takes advantage of these capabilities to transform snippet management from a passive storage utility into an intelligent, context-aware reasoning assistant. This work introduces **SmartCodeHub**, an AI-assisted snippet management system designed to improve retrieval accuracy, promote reuse efficiency, and reduce cognitive friction during development. The primary objectives of the research work are the following:

i. Integrate semantic reasoning and statistical retrieval within a unified architecture.
ii. Evaluate the adaptability of the system across programming languages and paradigms, focusing on reactive frameworks.
iii. Establish a reproducible evaluation methodology to incorporate LLM reasoning into code maintenance workflows.

II. MOTIVATION AND BACKGROUND

Traditional information retrieval approaches represent code snippets using Term Frequency or token-frequency models such as TF-IDF (Term Frequency-Inverse Document Frequency) [12-13]. Although effective for lexical matching, they ignore deeper program semantics.

LLM embeddings, on the other hand, capture syntactic and semantic relationships but introduce challenges in scalability and reproducibility [14].

Moreover, reactive systems exhibit event-driven flows and concurrency patterns that complicate code comprehension [15-16]. Developers require tools that not only locate code but also explain its usage context and dependencies. Conventional code snippet tools fail to capture the contextual meaning and intent behind a piece of code, especially in reactive programming environments where data dependencies are expressed through asynchronous streams [17-18]. Consequently, developers spend excessive time searching for relevant snippets or debugging mismatched implementations. Ko et al. [19], found that developers spend an average of 35\% of their time performing navigation mechanics within and between source files. Xia et al. [20], further substantiated this by identifying that developers frequently search for reusable code snippets, solutions to common programming bugs, and third-party libraries: tasks that are inherently time-consuming.

This gap motivates the need for an AI-enhanced solution capable of understanding code semantics, programming intent, and execution context simultaneously.

SmartCodeHub addresses these issues through a unified embedding space that connects code tokens, comments, and usage examples.

By combining symbolic and learned representations, it achieves both interpret ability and accuracy.

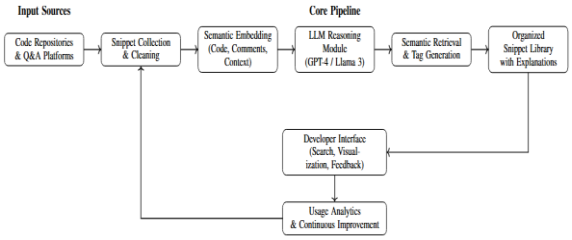


Figure 1: conceptual-overview of the Framework

Detailed Overview of the Framework:

The system collects and cleans code snippets from repositories and Q\&A platforms, embeds them semantically, and applies an LLM reasoning module to understand context and intent. Then it performs semantic retrieval and automatic tag generation to populate an organized and explainable snippet library. A developer-facing interface supports interactive search, visualization, and feedback, while usage analytics feed back into the system for continuous improvement.

a. Code Repositories \ Q&A Platforms

This is the data input layer of SmartCodeHub. It sources raw snippets from developer communities and repositories such as GitHub and Stack Overflow. These snippets often include function definitions, example usages, or code discussions. The data serve as the foundation for building a rich snippet knowledge base.

b. Snippet Collection & Cleaning

In this stage, the snippets are filtered, normalized, and deduplicated. The system removes incomplete code fragments and extracts meaningful metadata (for example, language type, framework, and context). The goal is to ensure that all downstream snippets are clean, consistent, and ready for semantic processing.

c. Semantic embedding

In this stage, our framework converts code, comments, and contextual information into vector embeddings. This process captures the semantic meaning of the snippet beyond keywords. The embedding layer bridges natural language (queries) and source code (snippets), enabling intent-based retrieval instead of pure keyword matching.

d. LLM Reasoning Module

At the core of SmartCodeHub lies the LLM reasoning module (e.g., GPT-4 or Llama 3). Interpret developer intent, analyze code semantics, and infer relationships between snippets. This component fuses natural language understanding with code comprehension, allowing the system to reason about function similarity, purpose, and potential use cases.

e. Semantic Retrieval \& Tag Generation

The reasoning outputs are used to retrieve relevant snippets based on semantic similarity. The module automatically generates human-readable tags and short explanations, improving snippet discoverability. For example, a search for 'debounce input stream' retrieves snippets that implement equivalent logic in multiple frameworks.

f. Organized Snippet Library

The retrieved and tagged snippets are stored in a centralized repository, a searchable knowledge base enriched with explanations. Each entry contains metadata, context, and relationships with other code fragments, ensuring traceability and easy reuse.

g. Developer interface (Search, Visualization, Feedback)

This layer provides an intuitive front-end for developers. Users can search for natural language or example code, visualize relationships between snippets, and provide feedback. The interaction data collected here guide the

continuous refinement of retrieval accuracy and tagging quality.

h. Usage Analytics \ Continuous Improvement

Analytics monitor usage patterns, retrieval success, and user feedback. These insights are used to refine embeddings, retrain models, and enhance retrieval quality over time, creating a self-improving adaptive system that grows smarter with use.

III. RELATED WORK

Snippet management tools such as CodeBox and Gisto provide structured repositories for storing and retrieving code fragments; however, they are heavily based on manual curation and lack advanced reasoning or automated organization capabilities [21-22]. These systems typically offer tagging, folder hierarchies, and basic search, but do not address semantic similarity or contextual relationships between snippets.

In contrast, AI-driven code assistants—including GitHub Copilot and Amazon CodeWhisperer—focus primarily on inline code generation and autocomplete rather than long-term snippet organization or reuse [23-25]. Although these tools significantly improve developer productivity during active coding tasks, they do not maintain a persistent, explainable, or user-customizable knowledge base of reusable code solutions.

Recent surveys and empirical studies highlight the growing potential of large language models for tasks such as code comprehension, documentation, and reasoning over software artefacts [26-28]. These works underscore the capabilities of LLMs in extracting intent, summarizing logic, and supporting developer decision-making. However, few explore the challenge of sustained reuse of code fragments, integration of semantic embeddings, or automated enrichment of code examples over time.

SmartCodeHub extends this landscape by introducing an LLM-powered framework for semantic retrieval, automatic tag generation, and contextual explanation of code snippets. Unlike existing tools, it emphasizes reproducibility, modularity, and persistent knowledge accumulation, bridging the gap between transient AI assistance and long-term snippet management.

i. Conventional Snippet Management Systems

Traditional snippet management tools, such as CodeBox and Gisto, provide structured repositories for storing and organizing reusable code fragments. These systems mainly depend on manual tag, folder-based categorization, and keyword search to facilitate retrieval [29]. Although effective for basic organization, their reliance on user-supplied metadata often leads to inconsistent tagging and redundant entries. Tavakoli et al. [30] introduced metadata enrichment techniques to improve the discernibility of the snippet; however, their approach still required manual annotation and lacked semantic awareness.

ii. AI-Assisted Code Completion and Generation

Recent advances in AI-driven code assistants—such as GitHub Copilot, Amazon CodeWhisperer, and TabNine—have transformed software development workflows by offering context-aware code completions and function synthesis[31-32].These systems leverage large language models to predict likely continuations of source code, significantly improving productivity. However, they primarily focus on real-time code generation within an IDE rather than long-term snippet organization, indexing, or reuse. Consequently, while they help during coding sessions, they do not address the broader challenges of snippet retrieval and semantic reuse across projects.

iii. LLMs for Code Understanding and Retrieval

A growing body of research investigates the application of Large Language Models (LLMs) to code understanding, summarization, and semantic search. Studies such as Benítez et al. [33], Duan et al. [34] and Russo et al. [35] highlight the capacity of LLMs to bridge the gap between the intent of natural language and the semantics of code. Despite this progress, most existing work emphasizes comprehension or translation tasks (e.g., code explanation, defect detection) rather than sustained snippet reuse. Furthermore, scalability, reproducibility, and explainability remain key open challenges when deploying LLM-based retrieval systems in real-world environments.

iv. Positioning of SmartCodeHub

SmartCodeHub extends this evolving landscape by introducing a reproducible, LLM-powered framework for semantic snippet retrieval and automatic tag generation. Unlike conventional tools that rely on keyword matching, or AI assistants that focus on code synthesis, SmartCodeHub integrates statistical and semantic reasoning to enable context-aware snippet reuse. Its architecture supports cross-language adaptability and uses LLM reasoning to bridge functional equivalence between syntactically different implementations. This combination of explainability, reproducibility, and adaptability positions SmartCodeHub as a next-generation system for intelligent code management and reuse.

IV. SYSTEM OVERVIEW

The project source code is anonymously available online at https://anonymous.4open.science/r/SmartCodeHub--0B4F

SmartCodeHub comprises three layers: the user-facing interface, the reasoning back-end, and the LLM inference engine.

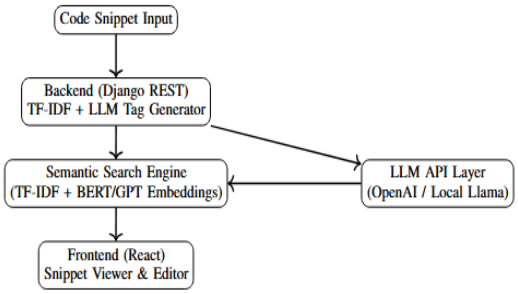


Figure 2: Architecture of SmartCodeHub: Integration of semantic search, AI tagging, and LLM-based suggestion.

Figure 2 presents the overall architecture of SmartCodeHub. A code snippet submitted by the user is first processed in the backend, where TF-IDF and an LLM-based tag generator produce both keyword features and semantic tags. These representations are then passed to the semantic search engine, which combines TF-IDF vectors with transformer-based embeddings for improved retrieval accuracy. The processed results are sent to the React frontend for viewing and editing. An LLM API layer (using either OpenAI or a local Llama model) supports both tagging and semantic embedding generation, enabling intelligent suggestions and enhanced search quality across the system.

a) Frontend Interface

The interface was designed in React; it allows developers to add, edit, and search snippets. It supports both textual and natural-language queries, displaying ranked results with contextual explanations generated from the backend.

b) Backend Reasoning Layer

The backend (Django REST) exposes APIs for snippet management and semantic retrieval. The term Frequency-Inverse Document Frequency (TF-IDF) provides a quick lexical baseline, while GPT-4-derived embeddings capture semantic similarity.

A ranking module merges both signals using a learned weighting function optimized for the validation data.

c) LLM Integration

An abstraction layer is implemented to support both cloud-hosted and locally deployed large language models (LLMs), enabling flexible experimentation across diverse environments. For privacy-sensitive or offline research settings, inference can be executed entirely on-device using Llama-based models provisioned through Ollama, ensuring that code, traces, and debugging artifacts never leave the local machine. Conversely, when scalability or advanced reasoning capabilities are required, the same interface can seamlessly route requests to cloud APIs without altering the surrounding system architecture.

V. AI-ASSISTED RETRIEVAL PIPELINE

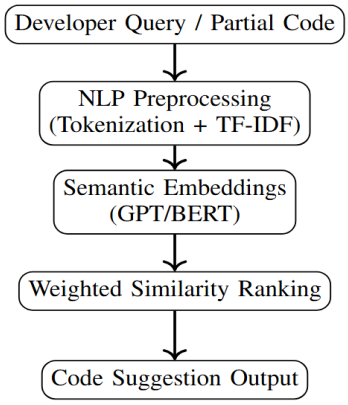


Figure 3: SmartCodeHub semantic retrieval and suggestion pipeline.

Figure 3 shows the semantic retrieval pipeline used by SmartCodeHub. A developer's query or partial code snippet is first processed using NLP techniques such as tokenization and TF-IDF. The system then generates semantic embeddings using models such as GPT or BERT. These embeddings are compared using a weighted similarity ranking module, which selects the most relevant matches. Finally, the top-ranked results are transformed into a code suggestion output returned to the developer.

a. Embedding Model

We used OpenAI's text-embeddings-3-large and Llama-3 for cross-model comparison.

Each snippet is converted to a 1 536-dimensional vector and stored in a FAISS ((Facebook AI Similarity Search)) index for fast similarity search. The reference documentation can be found on the GitHub repository:https://github.com/facebookresearch/faiss/wiki/Faiss-indexes

b. Ranking and Feedback

Results are ranked using cosine similarity and contextual re-weighting based on snippet metadata (language, framework, and historical reuse count).

User selections feed a reinforcement module that updates the ranking weights.

VI: IMPLEMENTATION DETAILS

SmartCodeHub was implemented over six months with roughly 6000 lines of Python and 3 500 lines of JavaScript.

1. Database: PostgreSQL with pgvector for embedding storage.
2. Backend: Django REST 4.2 + Celery for asynchronous LLM calls.

3. Frontend: React 18 with Tailwind CSS.
4. LLMs: GPT-4-Turbo (8 k context) and Llama-3 8B via Ollama.

The system supports multilingual code snippets (Python, JavaScript, Java) and integrates syntax highlighting through Prism.js.

VII: EVALUATION

1. Setup

Two datasets were used:

- 150 snippets from JavaScript (RxJS) projects.
- 120 snippets from Python (async FastAPI) repositories.

Baselines include SnippetsLab and Copilot. Metrics: retrieval accuracy, reuse time reduction, and developer satisfaction (Likert 1–5, N=12 participants).

2. Quantitative Results

Table 1 below presents a comparison of retrieval accuracy and code reuse performance across three tools: SnippetsLab, Copilot, and the proposed SmartCodeHub. Retrieval accuracy measures how effectively each system returns the most relevant code snippet based on a developer's query, while reuse time reduction quantifies how much the tool shortens the time required for developers to locate, adapt, and reuse existing code.

SnippetsLab, which relies mainly on keyword search, achieves a relatively low retrieval accuracy of 61.3\% and provides no measurable reduction in code reuse time. Copilot performs better with a retrieval accuracy of 72.5%, aided by its generative suggestions and semantic understanding of developer intent, resulting in an 18.4% reduction in reuse time.

SmartCodeHub outperforms both baseline tools, achieving an accuracy of 84.6% due to its hybrid retrieval strategy that integrates TF-IDF, semantic embeddings, and LLM-enhanced ranking. Additionally, SmartCodeHub reduces the reuse time by 42.1%, indicating that its combination of semantic retrieval and AI-generated tags significantly improves the developers' ability to find and reuse code efficiently.

Table 1
Retrieval Accuracy and Code Reuse Performance

Tool	Retrieval Accuracy	Reuse Time Reduction
SnippetsLab	61.3%	0%
Copilot	72.5%	18.4%
SmartCodeHub	84.6%	42.1%

SmartCodeHub achieves the highest retrieval precision, reducing the average search time from 35 seconds to 20 seconds per query.

3. Feature Comparison

Table 2
Feature Comparison with Baseline Tools

Feature	SnippetsLab	Copilot	CodeWhisperer	SmartCodeHub
Semantic Search	X	✓	✓	✓
AI Tagging	X	X	X	✓
Code Suggestion	X	✓	✓	✓
Cross-Language Support	X	✓	✓	✓

Table 2 summarizes how SmartCodeHub compares with existing tools across four key features: semantic search, AI tagging, code suggestion, and cross-language support. SnippetsLab does not provide any of these advanced capabilities, limiting it to manual snippet organization. Copilot and CodeWhisperer support semantic search and code suggestions, but lack AI-driven tagging, which means that users must organize snippets manually. In contrast, SmartCodeHub includes all four features, making it the only tool that combines intelligent search, automatic tagging, code suggestions, and broad language support in a single system.

4. Feature Coverage Visualization

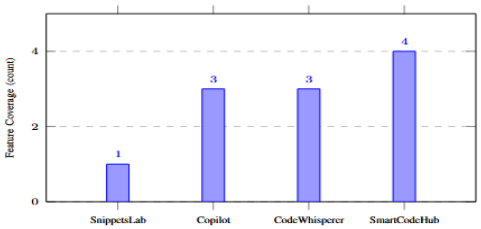


Figure 3: Number of intelligent features supported by each tool.

Figure 3 provides a visual comparison of the number of intelligent features each tool supports. SnippetsLab implements only one feature, while both Copilot and CodeWhisperer support three. SmartCodeHub leads with four features, showing the broadest coverage and the most complete set of intelligent capabilities among all evaluated tools.

\subsection{User Study}

The participants rated relevance, usability, and clarity. SmartCodeHub averaged 4.6/5 in relevance and 4.3/5 in usability, confirming the perceived benefits of semantic reasoning.

VIII DISCUSSION

The results confirm that semantic embeddings and AI-driven tagging significantly improve snippet discovery.

Performance in the participants' languages remained stable, suggesting generalizability.

However, computational overhead and potential hallucinations in LLM explanations highlight the need for lightweight local inference and result verification.

IX CONTRIBUTIONS

The contributions of this work are as follows:

- We design **SmartCodeHub**, an AI-assisted snippet management system that combines TF-IDF ranking, semantic code embeddings, and LLM reasoning for context-aware snippet discovery.
- We introduce an automated tag and explanation generation mechanism to support snippet organization and developer comprehension.
- We implement a cross-language snippet library and a developer interface for interactive retrieval and feedback.
- We report early empirical results demonstrating improved snippet relevance and reuse efficiency compared to keyword-based systems.

X THREATS TO VALIDITY

Internal validity threats arise from the limited size of the data set, small number of participants, and the potential bias in manually annotated tags, all of which may affect the reliability of the retrieval accuracy and usability results. The familiarity of the participants with the tools may also have influenced the outcomes.

External validity is limited by the focus of the dataset on JavaScript and Python, which can reduce generalizability to other languages or domains, particularly strongly typed or domain-specific environments.

Construct validity threats involve the reliance on subjective self-reported satisfaction scores, which can vary with individual biases, experience levels, and interpretations of rating scales.

Future work will reduce these threats by expanding the dataset and participant diversity, involving independent domain experts, and incorporating objective behavioral metrics—such as task completion time and error rates—to complement subjective assessments.

XI CONCLUSION

This paper introduced SmartCodeHub, a framework that applies semantic embeddings and LLM-based reasoning to improve code snippet retrieval and reuse. The initial results show gains in relevance and efficiency compared to traditional keyword-driven tools, demonstrating the potential to integrate LLM guidance into snippet management workflows. As an ongoing work, our aim is to expand the coverage of the dataset, conduct large-scale user studies, and refine cross-language retrieval. Future extensions will explore adaptive feedback mechanisms and domain-specific model tuning to strengthen reproducibility and deployment in real-world development environments.

REFERENCES

[1] Y. Hu, "Research on modularization-based code reuse technology in software system development," Applied Mathematics and Nonlinear Sciences, vol. 10, 09 2025.

[2] H. Sun, W. Ha, P.-L. Teh, and J. Huang, "A case study on implementing modularity in software development," Journal of Computer Information Systems, 07 2015.

[3] R. B. Mejba, S. Miazi, A. Palash, T. Sobuz, and R. Ranasinghe, "The evolution and impact of code reuse: A deep dive into challenges, reuse strategies and security," vol. 6, 11 2023.

[4] M. Sojer and J. Henkel, "Code reuse in open source software development: Quantitative evidence, drivers, and impediments," J. AIS, vol. 11, 03 2010.

[5] C. Ragkhitwetsagul, J. Krinke, M. Paix˜ao, G. Bianco, and R. Oliveto, "Toxic code snippets on stack overflow," CoRR, vol. abs/1806.07659, 2018. [Online]. Available: http://arxiv.org/abs/1806.07659

[6] I. Ndukwe, S. Licorish, and S. MacDonell, "Perceptions on the utility of community question and answer websites like stack overflow to software developers," IEEE Transactions on Software Engineering, vol. PP, pp. 1–13, 01 2022.

[7] F. Tang, B. Ostvold, and M. Bruntink, "Identifying personal data processing for code review," 01 2023.

[8] E. Czaplicki and S. Chong, "Asynchronous functional reactive programming for guis," vol. 48, 06 2013.

[9] G. Salvaneschi, S. Proksch, S. Amann, S. Nadi, and M. Mezini, "On the positive effect of reactive programming on software comprehension: An empirical study," IEEE Transactions on Software Engineering, vol. PP, pp. 1–1, 01 2017.

[10] D. H. Hagos, R. Battle, and D. B. Rawat, "Recent advances in generative ai and large language models: Current status, challenges, and perspectives," 07 2024.

[11] W. Yang, L. Some, M. Bain, and B. Kang, "A comprehensive survey on integrating large language models with knowledge-based methods," Knowledge-Based Systems, vol. 318, p. 113503, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705125005490

[12] B. Susanto, R. Ferdiana, and T. Adji, "Performance of traditional and dense vector information retrieval models in code search," 02 2024.

[13] X. Gu, H. Zhang, and S. Kim, "Deep code search," in Proceedings of the 40th International Conference on Software Engineering, ser. ICSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 933–944. [Online]. Available: https://doi.org/10.1145/3180155.3180167

[14] N. Bibi, A. Maqbool, T. Rana, F. Afzal, A. Akg¨ul, and S. M. Eldin, "Enhancing semantic code search with deep graph matching," IEEE Access, vol. 11, pp. 52 392–52 411, 2023.

[15] C. Liu, X. Xia, D. Lo, Z. Liu, A. E. Hassan, and S. Li, "Codematcher: Searching code based on sequential semantics of important query words," ACM Trans. Softw. Eng. Methodol., vol. 31, no. 1, Sep. 2021. [Online]. Available: https://doi.org/10.1145/3465403

[16] R. Cleaveland and S. A. Smolka, "Strategic directions in concurrency research," ACM Comput. Surv., vol. 28, no. 4, p. 607–625, Dec. 1996. [Online]. Available: https://doi.org/10.1145/242223.242252

[17] G. Salvaneschi, S. Proksch, S. Amann, S. Nadi, and M. Mezini, "On the positive effect of reactive programming on software comprehension: An empirical study," IEEE Transactions on Software Engineering, vol. 43, no. 12, pp. 1125–1143, 2017.

[18] S. Ramson, M. Brand, J. Lincke, and R. Hirschfeld, "Extensible tooling for reactive programming based on active expressions." The Journal of Object Technology, vol. 23, p. 1:1, 01 2024.

[19] D. Wightman, Z. Ye, J. Brandt, and R. Vertegaal, "Snipmatch: using source code context to enhance snippet retrieval and parameterization," in Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, ser. UIST ' 12. New York, NY, USA:Association for Computing Machinery, 2012, p. 219 – 228. [Online]. Available:https://doi.org/10.1145/2380116.2380145

[20] R. Padhye, P. Dhoolia, S. Mani, and V. S. Sinha, " Smart programming playgrounds, " in 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol. 2, 2015, pp. 607 – 610.

[21] [21] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung, "An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks," IEEE Transactions on Software Engineering, vol. 32, no. 12, pp. 971 – 987, 2006.

[22] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?" Empirical Software Engineering, vol. 22, 12 2017.

[23] E. Horton and C. Parnin, "Gistable: Evaluating the executability of python code snippets on github, " in 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, pp. 217 – 227.

[24] T. Diamantopoulos, G. Karagiannopoulos, and A. L. Symeonidis, "Codecatch: extracting source code snippets from online sources," in Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, ser. RAISE ' 18. New York, NY, USA: Association for Computing Machinery, 2018, p. 21 – 27. [Online]. Available:https://doi.org/10.1145/3194104.3194107

[25] B. Yetis，tiren, I. O ¨ zsoy, M. Ayerdem, and E. Tu ¨ zu ¨ n, "Evaluating the code quality of ai-assisted code generation tools: An empirical study on github copilot, amazon codewhisperer, and chatgpt," 04 2023.

[26] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, " Large language models for software engineering: A systematic literature review," 08 2023.

[27] P. Vaithilingam, E. L. Glassman, P. Groenwegen, S. Gulwani, A. Z. Henley, R. Malpani, D. Pugh, A. Radhakrishna, G. Soares, J. Wang, and A. Yim, "Towards more effective ai-assisted programming: A systematic design exploration to improve visual studio intellicode user experience," in 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2023, pp. 185 – 195.

[28] C. D. Benitez and M. Serrano, " The integration and impact of artificial intelligence in software engineering," International Journal of Advanced Research in Science Communication and Technology, vol. 3, pp. 279 – 293, 08 2023.

[29] Y. Duan, J. Edwards, and Y. Dwivedi, "Artificial intelligence for decision making in the era of big data – evolution, challenges and research agenda," International Journal of Information Management, vol. 48, pp.63 – 71, 10 2019.

[30] D. Russo, "Navigating the complexity of generative ai adoption in software engineering, " 2024. [Online]. Available: https://arxiv.org/abs/ 2307.06081

[31] S. Henninger, " Supporting the construction and evolution of component repositories," in Proceedings of IEEE 18th International Conference on Software Engineering, 1996, pp. 279 – 288.

[32] M. Tavakoli, A. Heydarnoori, and M. Ghafari, "Improving the quality of code snippets in stack overflow," 04 2016.

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Embedding-Based Machine Learning Approach for Automatic Classification of Turkish News Articles

Ahmet Atasoglu[1], Yavuz Selim Taspinar[2]

[1]*Mechatronics Engineering Department, Selcuk University , Konya*
258265001007@ogr.selcuk.edu.tr, ORCID: 0009-0008-8178-2177

[2]*Mechatronics Engineering Department, Selcuk University , Konya*
ytaspinar@selcuk.edu.tr, ORCID: 0000-0002-7278-4241

***Abstract*— In this study, an automatic text classification approach for Turkish news articles is presented. The savasy/ttc4900 dataset from HuggingFace, consisting of seven news categories, was used. News texts were converted into 768-dimensional vector representations using the embeddinggemma model on the Ollama framework. These embeddings were then used to evaluate the performance of several machine learning algorithms. Seven models were tested: Support Vector Classifier (SVC), Logistic Regression, Multilayer Perceptron, K-Nearest Neighbors, Random Forest, Gaussian Naive Bayes, and Decision Tree. Model performance was assessed using accuracy, precision, recall, and F1-score metrics. Results showed that SVC and Logistic Regression achieved the highest accuracy in the high-dimensional embedding space. The findings demonstrate that embedding-based representations offer strong discriminative capability for Turkish news classification and that deep learning–derived vector embeddings can be effectively combined with traditional machine learning methods. These results emphasize the importance of vectorized text representations in natural language processing research.**

***Keywords*—** natural language processing, text embeddings, language model, text classification, Gemma language model

I. Introduction

The ability of computers to understand and generate human languages has long been one of the fascinating topics in computer science. The idea that machines could solve problems by exhibiting intelligent behavior like humans was strikingly presented by Alan Turing's question, “Can machines think?” [1], which made significant contributions to computer science. Today, it is possible to process human language not only at the symbolic or grammatical level, but also in terms of semantic relationships [2]. This has paved the way for transferring a level of understanding and interpretation similar to human intuition to computers. Developments in statistical modeling [3] and deep learning [4], in particular, have enabled the emergence of data-driven solutions based on data-driven training rather than rule-based solutions by teaching complex linguistic features to computers. These solutions are now important fundamental methods in the field of natural language processing. Today, with the development of deep learning-based approaches in the field of natural language processing and their increasing applicability, studies using deep learning methods have become quite widespread [5]. This study addresses the problem of text classification. Text classification is a problem that involves determining which category texts created in different contextual categories belong to. [6] Research in this field contributes to the development of natural language understanding systems. The complexity of the text classification problem is often increased by the diversity of large data sets, differences in language structure, and ambiguities in meaning. By addressing the text classification problem, this study aims to understand the existing challenges and obtain findings on overcoming these challenges using various methods. Furthermore, by presenting an analysis of the findings, the study aims to contribute to future work by highlighting the implications of these results. [7] introduced the long short-term memory (LSTM) architecture, a new deep learning method. This architecture aims to solve the long-term information storage problem caused by insufficient backpropagation and diminishing error feedback. The LSTM architecture consists of gates specialized for different tasks and is capable of handling complex tasks by directing the error flow of these gates. The evaluations shared in the study demonstrate that the model in the new architecture can solve complex and long-delay tasks that recursive neural network algorithms cannot solve.

In a study [8], the relationships between words and sentences were analyzed and an attempt was made to model important elements in texts using a PageRank (Brin & Page, 1998)-based ranking model. The similarity operation was applied to the calculated graph structure, identifying the most similar ranked nodes to reveal keywords and important sentences in the text. The study particularly addressed its use in tasks such as extracting important information from large texts or identifying

keywords related to specific topics. In their work [9], two new model architectures were introduced for the continuous representation of word vectors from large datasets. The quality of these representations was measured in a semantic similarity task, and the results were compared with previously best-performing techniques based on different types of neural networks. The evaluation results reported significant reductions in computational cost and substantial improvements in accuracy. The study reported that learning word vectors based on semantic similarities from a 1.6 billion-word dataset took less than a day. [10] analyzed the model features required to reveal patterns in word vectors. A new global regression model was developed that combines the advantages of two important model families: global matrix factorization and local context window methods. The newly developed model, GloVe, was trained on a large text corpus to effectively utilize statistical information. The model was then evaluated on semantic similarity and named entity recognition tasks and compared with other models in the literature. [11] introduced a new algorithm, Adam, for stochastic objective function optimization. The algorithm is based on adaptive estimates of low-order moments and is easy to implement, computationally efficient, has low memory requirements, is invariant to cross-scaling of gradients, and is suitable for problems with large data and parameters. Theoretical convergence properties of the algorithm are analyzed, and experimental results demonstrate that Adam can yield results comparable to those of other optimization methods. In their work [12], a new artificial neural network-based approach, distinct from statistical machine translation, was proposed. They argued that the use of fixed-length vectors was a bottleneck in improving the performance of the basic encoder-decoder architecture, and a new mechanism was developed that allows the algorithm to automatically search for important fragments in the source sentence to predict the target language word. The developed model was evaluated on an English-French translation task, and its performance was compared with existing results. The Transformer architecture presented in [13] is presented as a significant new step in the field of natural language processing. The Transformer architecture was based on [12], which introduced an attention mechanism that allows the model to pay more attention to the most important parts of the input array elements during training. The attention mechanism is central to the new Transformer architecture and, as presented in [14], consists of two blocks: an encoder and a decoder. The developed model has been evaluated on machine translation tasks and has been shown to yield the best results in comparisons in the literature.

[15] examined text summarization methods based on sentence extraction. Feature extraction and summary generation were carried out using genetic algorithms. A dataset consisting of Turkish summaries of news texts was used as the dataset. During training, the genetic algorithm determined the optimal weight values for the documents' features, and summaries were generated accordingly.

In their study [16], a dataset for evaluating different models across nine natural language comprehension tasks, called General Language Understanding Assessment (GLUE), and a test dataset for evaluating and comparing the models, were presented. The study also presented basic evaluation results using the ELMo language model [17] using a transfer learning technique. [18] introduced a new pretrained language model consisting solely of the encoder block of the Transformer architecture. The BERT language model presented in this study produces output vectors that represent input sequences from both perspectives. It has also been demonstrated that it can work on different tasks by retraining on the output layer. The model has been evaluated on various natural language processing tasks, demonstrating particularly high performance in natural language understanding tasks. In their study [19], a new variant of BERT [18] is proposed, named Sentence-BERT (SBERT), to address the network's inadequacy in semantic similarity search and clustering tasks. SBERT proposes a new ternary network structure, making it suitable for semantic similarity tasks. The developed model is evaluated on similarity tasks and transfer learning tasks, and comparisons with the literature are shared. In their work [20], a new language model called BART, equivalent to the original Transformer architecture [13], was introduced, with both encoder and decoder blocks. In this study, we used denoising approaches by randomly permuting the order of sentences in the learned texts to generalize the encoder- and decoder-based models. Specifically, we evaluated these approaches in natural language generation and question-answering tasks. [21] introduced the GPT-3 language model, which was developed to demonstrate that augmenting language models can significantly improve task-independent, small-sample performance, sometimes reaching a level that rivals the best previous transfer learning approaches. The developed model consists of 175 billion parameters. GPT-3's performance was evaluated solely through text interaction on tasks with small training samples, without model updating. GPT-3 demonstrated strong performance on several tasks, including translation, question-answering, gap-filling tasks, and word scrambling. However, methodological issues related to training on small-sample learning datasets and large text corpora, which still struggled for GPT-3, were identified. It was also reported that GPT-3 was able to write articles that were difficult to distinguish between human-written and human-written. [22] introduces a new variant of the T5 model [23], called mT5, trained on a new dataset covering 101 languages. mT5 is evaluated on tasks that currently evaluate existing language models and also introduces a simple technique to prevent the model from translating into the wrong language. The code and model used in the study are open source. [24] presented a new approach aimed at understanding and improving pairwise word vectors used in machine translation. This new approach proposes a simple merging method based on text vectors. Furthermore, a new norm-based method focuses on more efficient use of word vectors. The developed methods were also evaluated in machine translation and string-to-string tasks. [25] demonstrated the use of word vectors as a binary classification method. The problem identified was the detection of fake news. Furthermore, various preprocessing steps were introduced before classification. Another aspect of

the study was to expand binary classification to six categories of falsehood, ranging from true to completely false. However, it was determined that this method was not as effective as the results obtained with binary classification. In their study [26], a classification problem using convolutional neural networks (CNNs) was investigated for Turkish texts. The developed model was also compared with other machine learning methods on the same data. The selected datasets varied in terms of text and number of classes, and the effect of word vector size on classification success was investigated. Stemming and deletion of filler words were applied in the text preprocessing, and the TF-IDF method was applied for vector representations. The performance of different preprocessing and vector representations was evaluated against each other on the developed model. [27] used deep learning technologies to summarize Turkish texts and generate leading headlines. The study used a dataset consisting of over 50,000 collected news texts and their respective headlines, and applied the necessary preprocessing for training. High-performance results were achieved as a result of the use of the transformer model. The results showed that the transformer architecture performed better with less training content than other deep learning models and demonstrated a higher level of grammatical performance. [28] evaluate the performance of the GPT-3 large language model in classifying tweets containing and not containing cyberbullying on a dataset of Turkish tweets. The results demonstrate that GPT-3 achieves sufficient accuracy to be used in detecting cyberbullying in tweet content. The study examines ChatGPT, a prominent large language model. The study discusses ChatGPT's general benefits and usage scenarios, and discusses its potential and potential risks in various fields such as law, medicine, mathematics, finance, and academic writing.

II. Materials and Methods

A. Data Processing

The dataset used in this study is the Turkish news dataset, savasy/ttc4900, provided through the HuggingFace Datasets library. The dataset contains seven different categories: politics, world, economy, culture, health, sports, and technology. The aim of the study is to classify news texts belonging to one of these categories using machine learning models. The dataset was first retrieved using the load_dataset() function, and the number of samples for each category was examined using the dataset_samples() function. The data distribution was observed to be balanced across categories. The texts used in model training were taken from the text column, and the classes were drawn from the category column. The dataset was split into 70% training and 30% test sections using stratified sampling. This approach ensured that each class was balanced across the training and test sections.

B. Encoding Texts to Vector Embeddings

Text Data To enable the use of text data in machine learning models, it must be converted into vector representations. For this purpose, the study utilized the Ollama library to generate 768-dimensional feature vectors from news articles using the embeddinggemma model. The encode_text() function provides a single interface for embedding texts. To reduce processing costs for large datasets, the embedding process was applied using a 16-element mini-batch structure. After each batch was processed, the results were combined to create training and test data matrices.

C. Preparing Train and Test Sets

After creating the text vectors, X_train_final was defined as the training vector matrix, X_test_final as the test vector matrix, and y_train and y_test as the training and test labels. By converting the dataset into a format suitable for a vector-based model, different classification algorithms were enabled to work directly on high-dimensional vectors.

D. Classification Models

In this study, a total of seven different machine learning classifiers were used: Support Vector Classifier (SVC), Multi-Layer Perceptron Classifier (MLPClassifier), K-Nearest Neighbors (KNN), Random Forest Classifier (RF), Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Decision Tree Classifier (DT). All models were trained with their default hyperparameters, thus comparing the fundamental effects of vector-based features across different algorithms. Throughout the training process, each model was trained with the training vector matrix and produced predictions on the test vector matrix.

E. Evaluating Models

Performance metrics were calculated separately for each classifier: Accuracy, Precision, Recall, and F1 Score. Additionally, to examine the class-based behavior of the models, confusion matrices (separate for each classifier), category-based F1 score comparisons, One-vs-Rest and ROC curves, and graphs containing AUC values were created. In the ROC analysis, decision function outputs were used for models without probability outputs, and if these were also unavailable, a warning message was placed on the relevant subgraph. All results were summarized in the final section by creating a comparison table.

III. Experimental Results

This section presents comparative performance analyses of seven different classification algorithms trained on embedding representations obtained from news texts. The models were evaluated on the test dataset using accuracy, precision, recall, and F1-score metrics, and the results are summarized in Figure 1. Figure 2 presents the confusion matrices of all the machine learning models evaluated in the study.

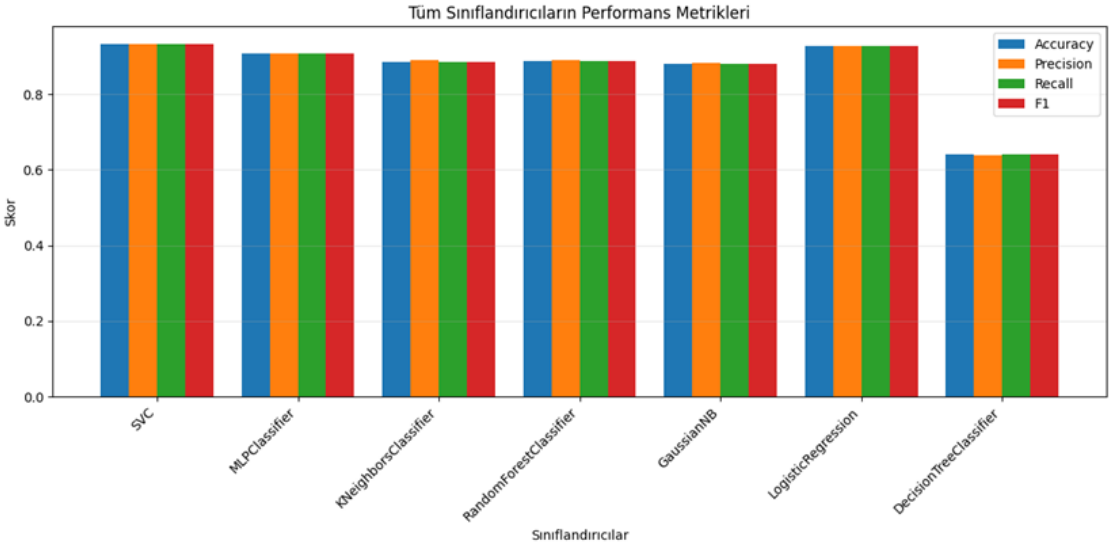


Fig 1. Performance metrics

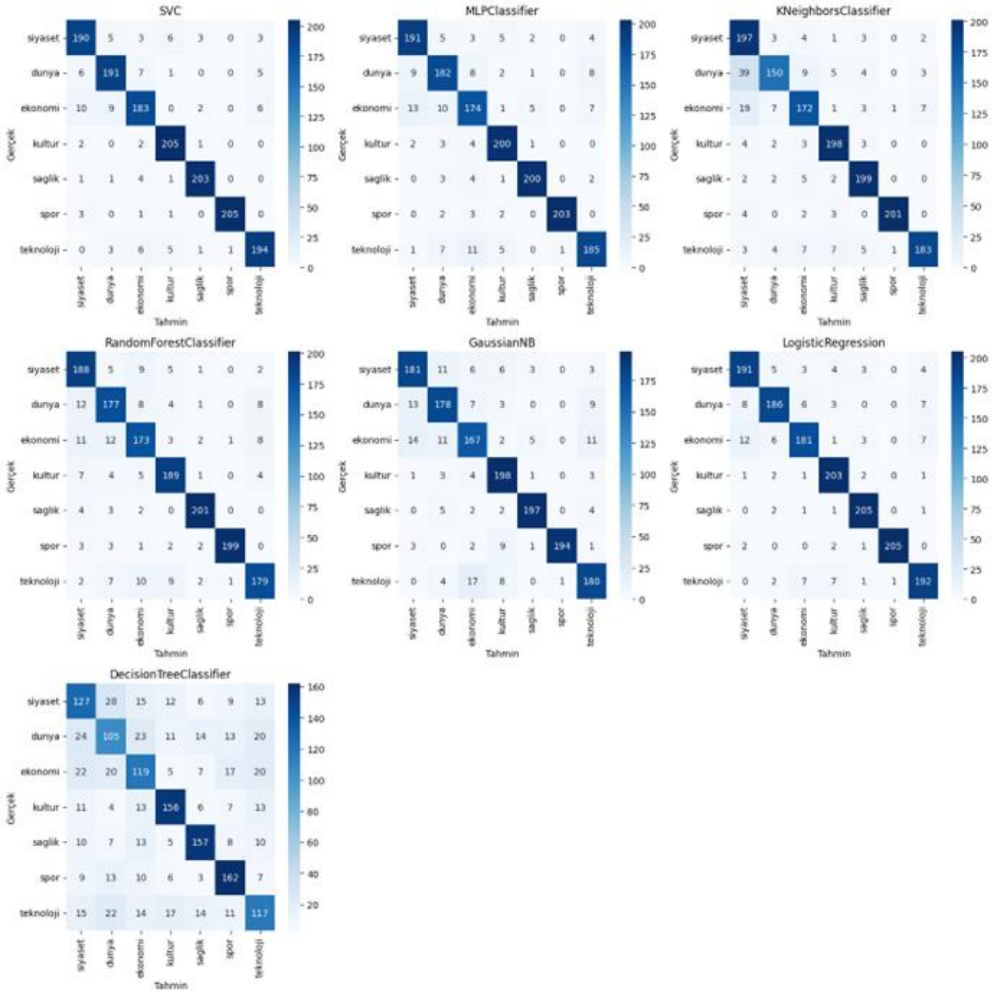


Fig. 2. Confusion matrix of all models

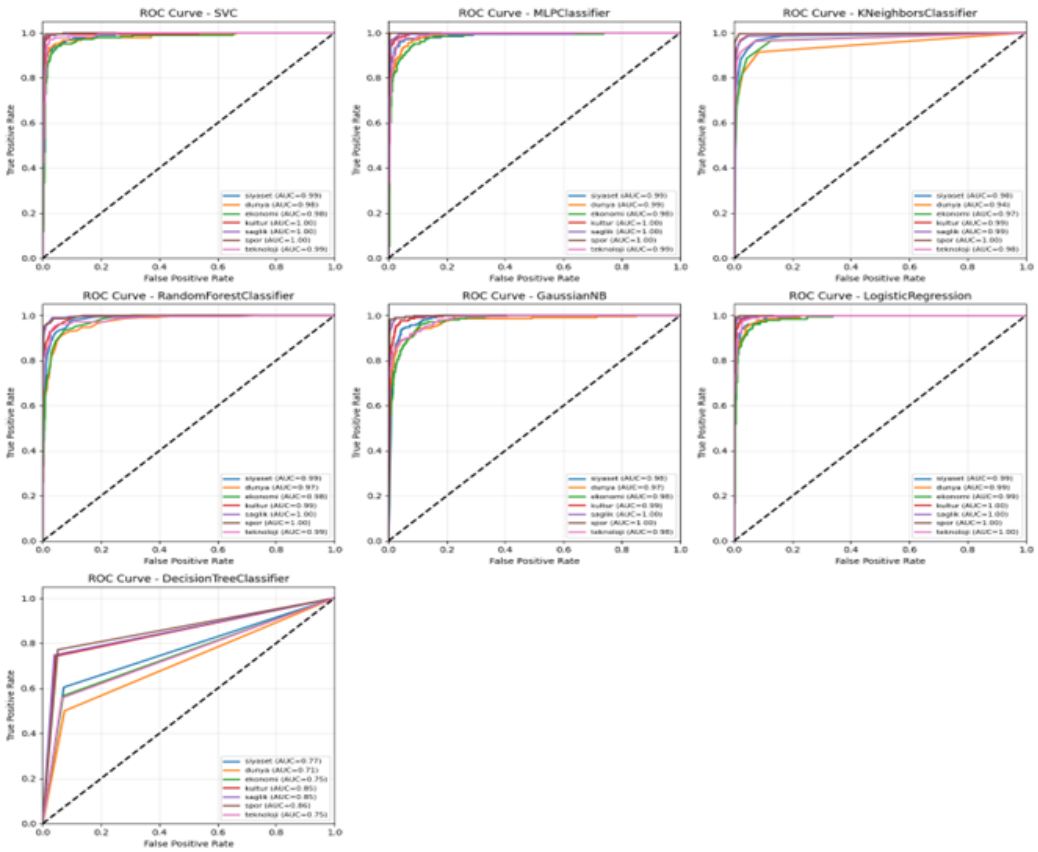


Fig 3. ROC of all modes

According to the results obtained, the Support Vector Classifier (SVC) model was the most successful classifier with 93.27% accuracy across all metrics. Logistic Regression (92.72%) followed SVC with a very close performance. The high performance of these two models demonstrates that linear/kernel-based methods are effective in high-dimensional embedding spaces.

When examining other models, the MLPClassifier achieved relatively high success (90.82%), while RandomForest (88.84%) and KNN (88.44%) performed at an intermediate level. Considering the continuous and dense structure of embedding-based representation, it was observed that tree and neighborhood-based methods could perform more limited discrimination in this space.

The GaussianNB (88.10%) model achieved reasonable accuracy despite its generative structure; however, it showed lower success compared to other discriminative models due to the complexity of the class distribution. The lowest performance was achieved with the DecisionTree classifier (64.15%), revealing that the single tree structure was insufficient to capture the complex data distribution based on embedding.

Overall, the results show that embedding-based vector representations provide high discriminative power for Turkish news classification; specifically, linear/kernel-based methods such as SVC and Logistic Regression perform best with these types of representations. Figure 3 presents the ROC curves of the machine learning models.

IV. Conclusions

In this study, a vector-based approach was used to classify Turkish news texts by category, and seven different machine learning algorithms were compared. Since the text representation vectors derived from the Gemma model can directly learn semantic information in natural language, high classification performance was achieved in all models. The results obtained showed that some models are naturally more compatible with vector-based data. In particular, SVC, Logistic Regression, and Random Forest achieved the highest accuracy on the test set. The Gaussian Naive Bayes and Decision Tree models performed less well than other models on high-dimensional text vectors. When examining the One-vs-Rest ROC curves, it was observed that the AUC values were high for all classes in many models. This indicates that vector-based representations provide strong distinctions between classes. Overall, the findings of the study reveal that large language model-based vector models are quite effective for Turkish news classification. For future work, model performance can be further improved by performing hyperparameter optimization, different vector models (BERT, RoBERTa, Mistral, etc.) can be added to the comparison, end-to-end learning can be applied with larger neural network models, and data augmentation techniques can be tested on the news content in the dataset. In

conclusion, this study demonstrates the high accuracy performance achieved by vector-based text classification approaches using different algorithms in Turkish news categories.

REFERENCES

[1] A. M. Turing, "I.—Computing Machinery And Intelligence", *Mind*, c. LIX, sy 236, ss. 433-460, Eki. 1950, doi: 10.1093/mind/LIX.236.433.

[2] Y. LeCun, Y. Bengio, ve G. Hinton, "Deep learning", *Nature*, c. 521, sy 7553, Art. sy 7553, May. 2015, doi: 10.1038/nature14539.

[3] Y. Bengio, R. Ducharme, ve P. Vincent, "A Neural Probabilistic Language Model", içinde *Advances in Neural Information Processing Systems*, MIT Press, 2000. Erişim: 14 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://papers.nips.cc/paper_files/paper/2000/hash/728f206c2a01bf572b5940d7d9a8fa4c-Abstract.html

[4] F. Hu, "Survey on Neural Networks in Natural Language Processing", içinde *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, Nis. 2023, ss. 591-594.doi: 10.1109/ICCECT57938.2023.10141113.

[5] D. Küçük ve N. Arici, "Doğal dil işlemede derin öğrenme uygulamalari üzerine bir literatür çalişmasi", *UYBİSBBD*, c. 2, sy 2, Art. sy 2, Ara. 2018.

[6] A. C. Tantuğ, "Metin Sınıflandırma", *TBV-BBMD*, c. 5, sy 2, Art. sy 2, Haz. 2016.

[7] S. Hochreiter ve J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, c. 9, sy 8, ss. 1735-1780, Kas. 1997, doi: 10.1162/neco.1997.9.8.1735.

[8] R. Mihalcea ve P. Tarau, "TextRank: Bringing Order into Text", içinde *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin ve D. Wu, Ed., Barcelona, Spain: Association for Computational Linguistics, Tem. 2004, ss. 404-411. Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://aclanthology.org/W04-3252

[9] T. Mikolov, K. Chen, G. Corrado, ve J. Dean, "Efficient Estimation of Word Representations in Vector Space", 06 Eylül 2013, *arXiv*: arXiv:1301.3781. doi: 10.48550/arXiv.1301.3781.

[10] J. Pennington, R. Socher, ve C. Manning, "Glove: Global Vectors for Word Representation", içinde *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, ss. 1532-1543. doi: 10.3115/v1/D14-1162.

[11] D. P. Kingma ve J. Ba, "Adam: A Method for Stochastic Optimization", *CoRR*, Ara. 2014, Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.semanticscholar.org/paper/Adam%3A-A-Method-for-Stochastic-Optimization-Kingma-Ba/a6cb366736791bcccc5c8639de5a8f9636bf87e8

[12] D. Bahdanau, K. Cho, ve Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", *CoRR*, Eyl. 2014, Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.semanticscholar.org/paper/Neural-Machine-Translation-by-Jointly-Learning-to-Bahdanau-Cho/fa72afa9b2cbc8f0d7b05d52548906610ffbb9c5

[13] A. Vaswani *vd.*, "Attention Is All You Need", 01 Ağustos 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[14] I. Sutskever, O. Vinyals, ve Q. V. Le, "Sequence to Sequence Learning with Neural Networks", *ArXiv*, Eyl. 2014, Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.semanticscholar.org/paper/Sequence-to-Sequence-Learning-with-Neural-Networks-Sutskever-Vinyals/cea967b59209c6be22829699f05b8b1ac4dc092d

[15] O. Kaynar, Y. E. Işik, Y. Görmez, ve F. Demirkoparan, "Otomatik Metin Özetleme İçin Genetik Algoritma Tabanli Cümle Çikarimi", *Yönetim Bilişim Sistemleri Dergisi*, c. 3, sy 2, Art. sy 2, Ara. 2017.

[16] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, ve S. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", içinde *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium: Association for Computational Linguistics, 2018, ss. 353-355. doi: 10.18653/v1/W18-5446.

[17] M. E. Peters vd., "Deep Contextualized Word Representations", içinde Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), M. Walker, H. Ji, ve A. Stent, Ed., New Orleans, Louisiana: Association for Computational Linguistics, Haz. 2018, ss. 2227-2237. doi: 10.18653/v1/N18-1202.

[18] J. Devlin, M.-W. Chang, K. Lee, ve K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 24 Mayıs 2019, *arXiv*: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[19] N. Reimers ve I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", içinde Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), K. Inui, J. Jiang, V. Ng, ve X. Wan, Ed., Hong Kong, China: Association for Computational Linguistics, Kas. 2019, ss. 3982-3992. doi: 10.18653/v1/D19-1410.

[20] M. Lewis *vd.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", içinde *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, ve J. Tetreault, Ed., Online: Association for Computational Linguistics, Tem. 2020, ss. 7871-7880. doi: 10.18653/v1/2020.acl-main.703.

[21] T. B. Brown *vd.*, "Language Models are Few-Shot Learners", *ArXiv*, May. 2020, Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.semanticscholar.org/paper/Language-Models-are-Few-Shot-Learners-Brown-Mann/6b85b63579a916f705a8e10a49bd8d849d91b1fc

[22] L. Xue vd., "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer", içinde Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online: Association for Computational Linguistics, 2021, ss. 483-498. doi: 10.18653/v1/2021.naacl-main.41.

[23] C. Raffel *vd.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", 19 Eylül 2023, *arXiv*: arXiv:1910.10683. doi: 10.48550/arXiv.1910.10683.

[24] X. Liu, "Exploring Word Embeddings to Enhance Neural Machine Translation", Ph.D., Ann Arbor, United States, 2021. Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.proquest.com/docview/2601500412/abstract/881E8D8897F74E1EPQ/12

[25] J. L. Hauschild, "Examining the Effect of Word Embeddings and Preprocessing Methods on Fake News Detection", Ph.D., Ann Arbor, United States, 2023. Erişim: 13 Ocak 2024. [Çevrimiçi]. Erişim adresi: https://www.proquest.com/docview/2809436764/abstract/881E8D8897F74E1EPQ/3

[26] G. Alparslan ve M. Dursun, "Konvolüsyonel Sinir Ağları Tabanlı Türkçe Metin Sınıflandırma", *Bilişim Teknolojileri Dergisi*, c. 16, sy 1, Art. sy 1, Oca. 2023, doi: 10.17671/gazibtd.1165291.

[27] A. Karaca ve Ö. Aydin, "Transformatör mimarisi tabanlı derin öğrenme yöntemi ile Türkçe haber metinlerine başlık üretme", *GUMMFD*, c. 39, sy 1, Art. sy 1, Ağu. 2023, doi: 10.17341/gazimmfd.963240.

[28] Ç. Koçak ve T. Yiğit, "Gpt-3 Sınıflandırma Modeli İle Türkçe Twitlerin Siber Zorbalık Durumlarının Belirlenmesi", *GMBD*, c. 9, sy 4-ICAIAME 2023, Art. sy 4-ICAIAME 2023, Ara. 2023.fficient Estimation of Word Representations in Vector Space", 06 Eylül 2013, *arXiv*: arXiv:1301.3781. doi: 10.48550/arXiv.1301.3781.

Proceedings Paper DOI: 10.58190/icisna.2025.150

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Tomato Seed Classification with Artificial Intelligence: A SqueezeNet-Based Approach

Abdulkadir Saday[1], Ilker Ali Ozkan[2]

[1]*Electrical and Electronic Engineering, Selcuk University, Konya, Türkiye*
asaday@selcuk.edu.tr, ORCID: 0000-0002-0406-711X

[2]*Computer Engineering, Selcuk University, Konya, Türkiye*
ilkerozkan@selcuk.edu.tr, ORCID: 0000-0002-5715-1040

Abstract*— The development of agricultural technologies is of great importance in improving agricultural production processes, enhancing crop quality, reducing production costs, and optimizing resource utilization. Seed quality is crucial in agricultural production in terms of plant development and yield. The use of high-quality seeds supports healthy plant development, thus increasing the productivity and sustainability of agriculture. Traditional methods rely on approaches such as visual inspection, manual sorting, and biological testing. However, these methods have significant limitations due to their time-consuming nature, dependence on human experience, and inability to provide sufficient efficiency in large-scale applications. In this study, an artificial intelligence-based decision mechanism is proposed for the automatic classification of healthy and unhealthy tomato (Solanum lycopersicum) seeds. A dataset of tomato seeds was specifically created for this study. The dataset consists of a total of 200 tomato seed images obtained under different environmental conditions, and the generalization ability of the model was strengthened by applying data augmentation and various preprocessing techniques. A deep learning-based SqueezeNet model, capable of providing high accuracy rates with low memory requirements, was used for feature extraction from the obtained tomato seed images. Model performance was evaluated using a 5-fold cross-validation method, and classification accuracy, precision, sensitivity, and F1 score were analyzed. Furthermore, quantization was applied to assess the model's usability in mobile and field applications, and it was observed that discrimination was achieved without performance loss. In comparative analyses, experiments with a YOLO-based object detection approach revealed that lightweight CNN architectures that perform direct classification are more effective when dealing with small and visually similar objects. In conclusion, this study demonstrates that a SqueezeNet-based deep learning approach offers high accuracy, low computational cost, and practical applicability in the automatic classification of tomato seeds. The proposed method has the potential to reduce human error in agricultural quality control processes and contribute to the rapid and reliable assessment of seed quality.

***Keywords*— artificial intelligence, classification, deep learning, SqueezeNet, seed separation, seed quality, tomato seeds.**

I. INTRODUCTION

Seed quality has a multifaceted and critical importance in terms of agricultural production and food security. It also directly affects crop yield, agricultural productivity, food security, and nutritional value [1]. High-quality seeds are superior in terms of genetic and physiological purity and are free from seed-borne diseases and defects, which increases productivity [2]. While seed quality is affected by genetic and environmental factors, climate change leads to significant changes in seed characteristics [1, 3]. Studies show that high-quality seeds provide a 25-30% increase in yield in agricultural production [4]. Therefore, accurate and reliable assessment of seed health in agricultural production processes is of great importance.

High-quality tomato seeds contribute to the optimization of fertilization practices, increasing crop yield and also improving nutritional value [5]. Well-managed fields with high-quality seeds provide superior germination and viability, which are crucial for achieving potential yields [6]. In addition, the use of quality seeds reduces the need for excessive agricultural chemicals, promoting sustainable agricultural practices [7]. Studies have also shown that the quality of tomato seeds affects the nutritional content of the fruit [8].

Traditional methods used for seed quality assessment, such as visual inspection, washing tests, and seed soaking methods, rely on the interpretation of visual symptoms and often fail to detect latent symptoms [9]. These methods are time-consuming and may not yield accurate and reliable results, making it difficult to assess the economic suitability of seeds under field conditions [10]. The procedures involved in traditional methods are often lengthy and labor-intensive, making them less efficient for large-scale seed quality assessment. These methods may not accurately detect damage or provide comprehensive information about seed viability and health [9, 11]. This situation increases the need for faster, more objective, and automated systems in seed quality assessment processes.

Computer Vision (CV) systems enable non-destructive, contactless, and objective evaluation of agricultural products, providing consistent quality assessments without human bias [12]. These systems automate labor-intensive tasks such as sorting, grading, and defect detection, significantly reducing manual labor and associated costs [13, 14]. Machine learning and CV techniques, especially when combined with deep learning (DL), provide high accuracy and speed in the detection and classification of agricultural products. DL models demonstrate superior accuracy in identifying defects and classifying products compared to traditional methods [15, 16]. The use of these technologies can automate quality control processes, helping to reduce production costs and improve overall product quality, leading to higher revenues and faster market access. CV systems allow for non-destructive quality assessment while preserving seeds for later use. This is particularly beneficial for tests such as purity analysis and germination tests [17]. Machine learning algorithms can provide a comprehensive assessment of seed quality by analyzing multiple features such as shape, color, and texture. The integration of ML and CV into seed quality control offers significant improvements in accuracy, speed, and reliability, making them invaluable tools in modern agriculture.

This study proposes a SqueezeNet-based deep learning approach for the automatic classification of healthy and unhealthy tomato seeds. In the proposed method, classification is performed by extracting distinctive features from tomato seed images, and model performance is evaluated using a 5-fold cross-validation method. Furthermore, quantization is applied to assess the model's usability in mobile and field applications, and a comparative analysis is conducted using an object detection-based YOLO approach. The main objective of this study is to develop a decision-making mechanism that automatically, accurately, and quickly evaluates the quality of seeds used in agricultural production and to make this process more efficient. It is also aimed that the results obtained will contribute to the automation of quality control in agricultural production processes and to increasing product yield.

The remaining sections of this study present the methods followed and experimental results in detail. The materials and methods section describes the dataset creation process, the image processing techniques used, and the details of the machine learning algorithms. The experimental results section presents the performance results of each algorithm in tables and graphs, and a comparative analysis is conducted. Finally, the conclusions section summarizes the overall findings of the study, discussing the applicability of the proposed methods in agricultural production and potential future improvements.

II. Related Work

This section reviews previously applied methods for classifying tomato seed quality. Areas examined include image processing techniques used in seed analysis, machine learning algorithms used for seed classification, deep learning models used for feature extraction, and computer vision applications in agricultural quality control. Relevant articles and studies are listed below.

Koppad et al. developed a method using deep learning algorithms for the automatic classification of soybean seeds. In the study, seed classification based on quality factors such as shape, size, color, and surface features was achieved using image recognition and deep learning techniques. This process increased accuracy by minimizing human errors and subjective evaluations compared to manual methods. ResNet50, MobileNetv2, DenseNet121, YOLOv5, and YOLOv8 models were used in the research. Among all models, YOLOv8 showed the best performance with an accuracy rate of 91% [18].

Kumari and colleagues conducted a study on the detection, classification, and counting of mixed seeds using the YOLOv5 deep learning model. In the study, deep learning models were applied to detect and count mixtures of various seed types such as flax, fringed vetch, red clover, radish, and rye. Seed images were obtained with a Canon LP-E6N R6 5D Mark IV camera, and dataset annotation was performed with the Robo-flow platform. The generalization power of the model was increased using data augmentation techniques. The YOLOv5 model showed the best performance with 96.96% recall, 94.81% precision, 68.62% mAP, and a CPU time of 28.8 seconds for a test image [19].

Jian Li and colleagues developed a method combining hyperspectral RGB imaging and deep learning techniques for the identification of maize seed varieties. In the study, the aim was to detect seed varieties by reconstructing RGB images with hyperspectral data. Hyperspectral bands with R, G, and B features were selected, and the image set reconstructed with these bands was used. Then, the model was improved by adding a coordinate attention (CA) mechanism to the ResNet50 model. The results showed that the accuracy rate reached 86.28% with the unimproved model, while the accuracy increased to 88.18% with the improved model [20].

Yu Xia and his team developed a method based on a YOLOv5-based deep learning model to detect surface defects in maize seeds. In the study, surface defects of maize seeds were detected quickly and effectively with an image-based information gathering system. Various models were used for surface defect recognition, and the ECA-Improved-YOLOv5S-Mobilenet model yielded the best results. This model combined lightweight architecture and high accuracy to detect surface defects in maize seeds with 92.8% accuracy, 98.9% recall rate, and 95.5% mAP0.5. The model was able to quickly detect surface defects at different levels and improve efficiency in seed classification and planting processes [21].

Basol and Toklu developed a deep learning-based seed classification method and integrated it into a mobile application. In the study, a dataset was created from high-resolution images, taking into account the morphological structures of various seed types, and this dataset was processed using deep learning techniques such as CNN (Convolutional Neural Network). As a result of training using pre-trained models such as ResNet50, InceptionV3, Xception, and InceptionResNetV2, an accuracy of up to 99% was achieved. The highest performing model was converted to a mobile platform, and a mobile application was developed that allows users to quickly and accurately classify seed types by taking photos [22].

When examining studies aimed at determining seed quality, it is seen that deep learning is applied to automate seed separation, classification, and quality assessment. Studies demonstrate the effectiveness of deep learning models such as YOLOv5, ResNet50, MobileNetv2, and CNNs in accurately classifying and identifying seeds based on various quality characteristics, including shape, color, and texture [18, 23, 24]. Deep learning models show high performance in seed quality classification. The high accuracy, recall rates, and F1 scores obtained demonstrate their potential to improve seed quality assessment and disease detection [25, 26]. The use of deep learning techniques such as CNNs can contribute to breeding and yield improvement by leading to the development of automated software for high-yield seed phenotyping, quality assessment, and prediction [22].

In deep learning studies, comprehensive and accurately labeled datasets are needed for successful training of models [25]. However, deep learning models pose challenges in real-time applications on devices with limited resources (e.g., mobile applications or field-use devices), primarily due to their high computational power and storage capacity requirements. Lightweight deep learning models developed to determine seed health and quality should aim to overcome these challenges. In this context, solutions that reduce computational complexity and resource limitations while maintaining high accuracy rates should be emphasized [27]. Furthermore, these models need to be optimized in terms of computational resources, efficiency, and interpretability in practical applications.

Therefore, further research and development of deep learning methods for seed quality classification and prediction will play a significant role in improving the efficiency of agricultural processes [24, 25, 28].

III. Materials and Methods

In this study, a deep learning-based approach was developed to classify tomato seeds as healthy and unhealthy. The proposed method utilizes SqueezeNet, a lightweight convolutional neural network architecture, to extract distinctive features from tomato seed images. The model's performance was analyzed using a 5-fold cross-validation method and various evaluation metrics. The overall flowchart of the proposed method is shown in Fig.1.

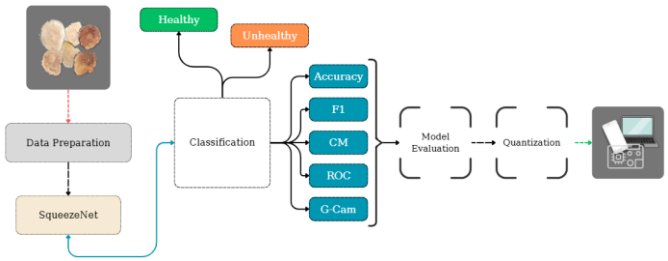


Fig. 1 General workflow of the proposed SqueezeNet-based tomato seed classification model

This section details the dataset used, preprocessing steps, proposed model structure, training process, and performance evaluation methods.

A. Tomato Seed Data Set

The dataset used in this study consists of a total of 200 color (sRGB) images of tomato (Solanum lycopersicum) seeds. The images were obtained under different environmental and lighting conditions using a high-resolution Samsung camera. The dataset was divided into two classes: 100 healthy and 100 unhealthy tomato seeds.

The dataset underwent enhancement processes prior to feature extraction and classification. Augmentation techniques such as mirroring and rotation were applied to diversify the images and improve classification accuracy. As a result of this process, the total number of images in the dataset increased to 800. The dataset expansion was carried out primarily to reduce misclassifications between seeds and potential overfitting. In this way, a more balanced and representative dataset was created by increasing visual variation between classes.

Table 1 shows the image distribution of the post-classification dataset. The healthy class generally includes fresh seeds with high germination potential, while the unhealthy class includes seeds exhibiting structural defects, color changes, and low viability. The quality of tomato seeds was determined primarily by surface structure and color changes, and these changes were used as the main criteria in classification.

TABLE I - Distribution of Tomato Seed Dataset Images

Seed Classes	Images	Number of Data
Healthy		100
Unhealthy		100
Total		200

B. Preprocessing and Data Augmentation

All images given as input to the deep learning model were rescaled to 227×227 pixels to meet the requirements of the SqueezeNet architecture. During the training process, the images were subjected to various data enhancement operations to improve the model's resilience to different transformations. These operations included random rotation, horizontal reflection, scaling, and translation.

The applied data enhancement techniques aim to enable the model to accurately classify seeds even under varying angles, scales, and positional changes. This approach plays a significant role in improving the model's generalization performance, especially in datasets with a limited number of images.

C. SqueezeNet-Based Deep Learning Model

Deep learning-based convolutional neural networks (CNNs) are a powerful artificial intelligence technique widely used in image analysis and processing. CNNs treat each pixel in images as a numerical value and extract meaningful features from images by analyzing the relationships between these numerical values. When an image is fed into the network, the CNN learns

the connections between these pixels in layers and extracts features. During this process, various mathematical operations are applied in successive layers to obtain features from the image. In this study, SqueezeNet, a lightweight deep learning model, was chosen for the classification of tomato seeds. This model demonstrates effective performance in extracting different features from images.

SqueezeNet is a lightweight convolutional neural network (CNN) architecture designed to achieve high accuracy with significantly fewer parameters compared to traditional CNNs like AlexNet. The core building block of SqueezeNet is the Fire module, consisting of a compression layer and an expansion layer. The compression layer uses 1x1 convolutions to reduce the number of input channels, while the expansion layer uses a mix of 1x1 and 3x3 convolutions to increase the number of output channels [29]. It is particularly noteworthy for its efficiency, which includes up to 50 times less weight while maintaining comparable accuracy levels [30]. This reduction in parameters makes SqueezeNet highly suitable for applications with limited computational resources, such as mobile and embedded systems [31].

The proposed method adopts a transfer learning approach using a SqueezeNet model previously trained on large-scale datasets. The model's original classification layers have been restructured to be suitable for classifying tomato seeds as healthy and unhealthy. This allows for effective classification with a limited number of images by leveraging previously learned general visual features.

D. Model Training and Cross-Validation

During the model training process, the dataset was evaluated using a 5-fold cross-validation method. This approach allowed for the analysis of the model's performance in different training and testing phases and increased the generalizability of the results. In each cross-validation step, a portion of the dataset was allocated for testing, while the remaining portion was used for training and validation.

The Adam optimization algorithm was used during model training. The training process was limited to a specific number of epochs, and an early stopping mechanism was activated if the improvement in validation loss stopped. This prevented overlearning and made the training process more efficient.

E. Performance Appraisal Criteria

The classification performance of the proposed model was analyzed using commonly used evaluation metrics such as accuracy, precision, recall, and F1 score. Additionally, receiver operating characteristic (ROC) curves and area under the curve (AUC) values were calculated to assess the model's ability to discriminate between classes.

The results obtained during the cross-validation process were combined to create a confusion matrix, and the model's performance for both classes was examined in detail. These evaluations comprehensively demonstrate the effectiveness of the proposed SqueezeNet-based approach in tomato seed classification.

IV. Experimental Results and Discussion

In this study, the performance of the proposed SqueezeNet-based tomato seed classification approach was evaluated using a 5-fold cross-validation method. Model performance was assessed using common performance metrics such as accuracy, precision, sensitivity, F1 score, and ROC–AUC. Additionally, post-quantization performance results were examined to evaluate the model's usability in field and mobile applications.

A. Classification Performance

The complexity matrix, created by combining the results obtained during the cross-validation process, shows that the model offers high and balanced classification performance for both classes. According to the results, 99 out of 100 healthy tomato seed samples were correctly classified, with only 1 sample being misclassified. Similarly, 94 out of 100 unhealthy tomato seed samples were correctly identified, while 6 samples were incorrectly classified as healthy. The combined complexity matrix, obtained to examine the classification performance of the model in detail, is presented in Fig. 2.

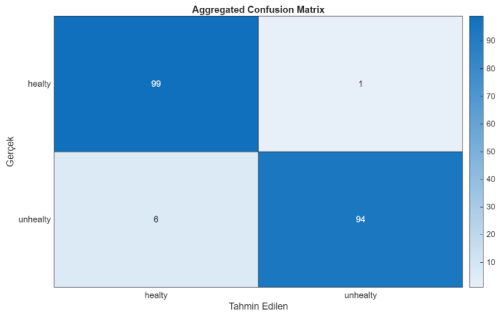


Fig. 2 The combined matrix obtained for the SqueezeNet-based model.

Based on these results, the overall classification accuracy of the proposed SqueezeNet-based approach was calculated as 96.5%. The high sensitivity value obtained, particularly for the healthy seed class, indicates that the model is highly successful in accurately identifying quality seeds. A summary of the performance metrics obtained during the cross-validation process of the proposed model is given in Table 2.

TABLE II - Class-based and overall performance results of the SqueezeNet-based model

Criteria	**Healthy**	**Unhealthy**
Precision	94.3	98.9
Recall	99.0	94.0
F1-score	96.6	96.4
General Criteria	**Value**	
Cross-validation	5-Fold	
Total Sample Size	200	
ROC-AUC (Quantized)	1.00	
Accuracy (%)	96.5	
Average Education Time	~28 sec.	

This is of great importance in agricultural practices, as the accidental discarding of healthy seeds during cultivation can lead to economic losses.

B. ROC Curve and Discrimination Analysis

To evaluate the model's ability to discriminate between classes, ROC curves and AUC values were analyzed. The ROC curve obtained for the quantized SqueezeNet model shows that the model can distinguish both classes with high accuracy. The calculation of the AUC value as 1.00 in the relevant ROC curve reveals that the model retains its discrimination power even after quantization. The ROC curve of the quantized model is presented in Fig. 3.

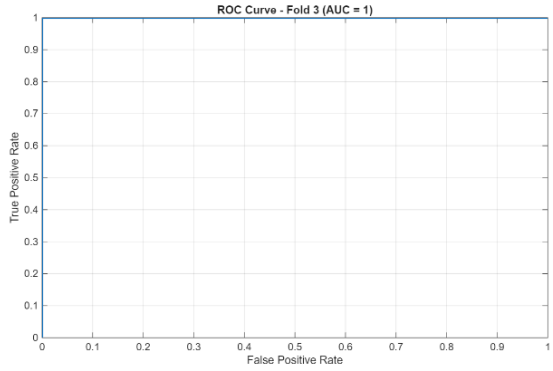


Fig. 3 ROC curve for quantized SqueezeNet model (Fold 3).

This result demonstrates that the proposed approach not only offers high accuracy but can also be used effectively in resource-constrained environments. The negligible performance loss after quantization supports the model's suitability for mobile and field applications.

C. Educational Process and Model Commitment

When the accuracy and loss curves of the training process were examined, it was observed that the model converged stably and did not show any signs of overfitting. With the activation of the early stopping mechanism, the training process was efficiently terminated and the validation performance was preserved. The short training time and the rapid convergence of the model clearly demonstrate the computationally efficient nature of the SqueezeNet architecture. The changes in accuracy and loss during the training and validation process of the model are shown in Fig. 4 for a representative cross-validation layer.

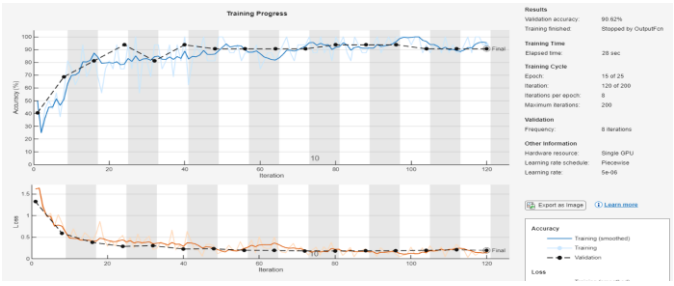


Fig. 4 Change in training and validation accuracy of the SqueezeNet model.

D. Comparison with the YOLO-Based Approach

To evaluate the effectiveness of the proposed SqueezeNet-based classification approach more comprehensively, a comparative analysis was performed with a YOLO-based object detection model trained using the Datature platform [32]. While YOLO-based approaches can provide strong results in object detection problems, they may exhibit some limitations in the analysis of small, singular, and visually similar objects, such as tomato seeds.

Analysis of the training logs obtained from the Datature platform revealed that the YOLO-based model achieved significant improvements in class-based performance metrics in the later stages of the training process. In particular, high precision, recall, and F1 score values were observed for both healthy and unhealthy seed classes during the final evaluation steps of the training. This indicates that the model was able to learn meaningful features after a sufficient training period.

However, when the training process is evaluated overall, it is noteworthy that the performance of the YOLO-based approach fluctuated throughout the training and exhibited low discrimination in the early stages. In particular, the stability of the object detection-based approach decreased in images with small object sizes and dominant backgrounds. This indicates that object detection methods require more precise tuning and larger datasets for small objects.

In contrast, the SqueezeNet-based approach, which focuses directly on image classification, demonstrated more stable performance throughout the training process and achieved higher accuracy values faster. Offering low computational costs thanks to its lightweight architecture, the SqueezeNet model stands out as a more suitable solution, especially for mobile and field applications with limited hardware resources. This comparison shows that the proposed method is based on an architectural choice more compatible with the problem definition and offers a more practical approach for tomato seed classification. This indicates that model selection should be based not only on final accuracy values but also on training stability and compatibility with the problem type. The key features and performance differences between the SqueezeNet-based and YOLO-based approaches are summarized in Table 3.

TABLE III - COMPARISON OF SQUEEZENET-BASED IMAGE CLASSIFICATION APPROACH AND YOLO-BASED OBJECT DETECTION APPROACH

Method	SqueezeNet	YOLO (Datature)
Duty	Classification	Object Detection
Performance Level	High and stable	High (final stage)
Educational Commitment	High	Medium-Low
Small Object Compatibility	High	Medium

E. Discussion

When the experimental results are evaluated overall, it is seen that the proposed SqueezeNet-based approach offers high accuracy, stable class performance, and low computational cost in tomato seed classification. In particular, the accurate identification of healthy seeds provides a significant advantage in terms of quality control processes in agricultural production. Maintaining performance after quantization strengthens the usability of the model in real-time and field applications. The Grad-CAM analysis results, obtained to visualize the image regions that the model focuses on when making classification decisions, are presented in Fig. 5.

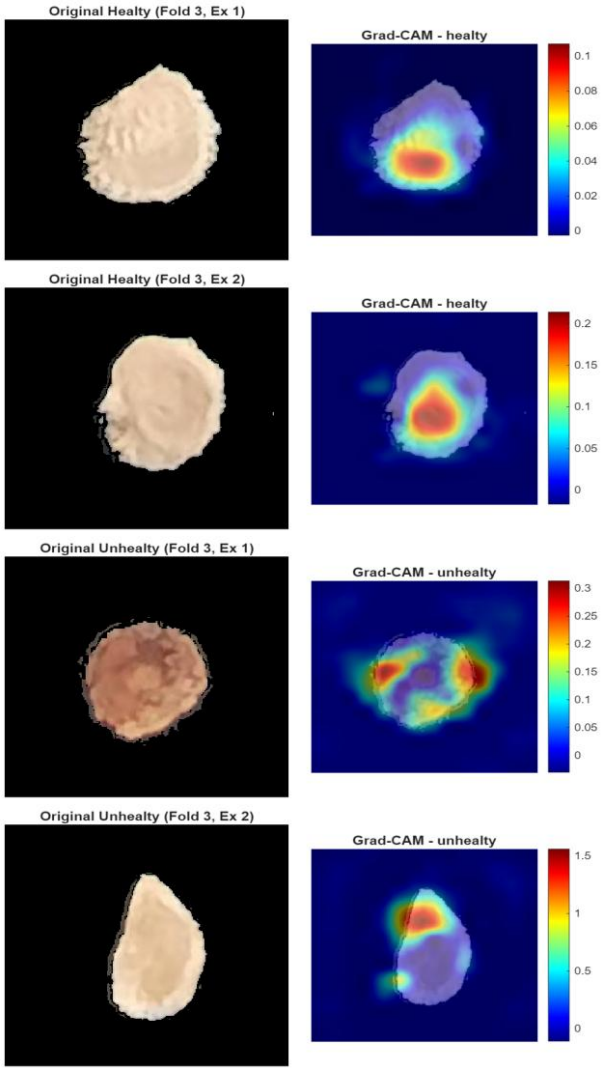


Fig. 5 Grad-CAM visualizations for healthy/unhealthy tomato seed samples.

V. Conclusion and Future Studies

This study proposes a SqueezeNet-based deep learning approach for the automatic classification of healthy and unhealthy tomato seeds. Considering the time-consuming and human-factor-dependent limitations of traditional seed quality assessment methods, an image-based and AI-supported decision-making mechanism has been developed. In the proposed approach, distinctive features are extracted from tomato seed images, and the classification process is performed using SqueezeNet, a lightweight and computationally efficient CNN architecture.

Experimental evaluations using the 5-fold cross-validation method showed that the proposed model achieved an overall classification accuracy of 96.5%. The resulting complexity matrix indicates that the model exhibits balanced performance for both classes and achieves particularly high success in correctly identifying healthy seeds. This offers a significant advantage in preventing economic losses in agricultural practices.

Quantization, performed to evaluate the model's suitability for field and mobile applications, revealed that high discrimination power was maintained. The ROC-AUC values obtained after quantization showed that the proposed approach offers strong classification performance despite its low computational cost. Furthermore, comparative analyses with the YOLO-based object detection approach trained on the Datature platform showed that lightweight CNN architectures based on direct image classification are more suitable and effective for small and visually similar objects such as tomato seeds.

In conclusion, this study demonstrates that a SqueezeNet-based deep learning approach offers high accuracy, low computational cost, and practical applicability in tomato seed classification. The proposed method has the potential to reduce human error in agricultural quality control processes and contribute to the rapid, reliable, and automated assessment of seed quality.

Future studies plan to expand the dataset with different tomato varieties, varying environmental conditions, and larger sample sizes. Additionally, the proposed model is intended to be implemented in a real-time mobile application or embedded system. Comparing different lightweight deep learning architectures and addressing multi-class seed quality assessment scenarios are also among the potential future research activities.

References

[1] *Wimalasekera, R., Role of seed quality in improving crop yields, in Crop Production and Global Environmental Issues. 2015. p. 153-168.*

[2] *2. Bailly, C. and M.V.G. Roldan, Impact of climate perturbations on seeds and seed quality for global agriculture. Biochemical Journal, 2023. 480(3): p. 177-196.*

[3] *3. Gaur, A., et al., importance of seed-borne diseases of agricultural crops: Economic losses and impact on society, in Seed-Borne Diseases of Agricultural Crops: Detection, Diagnosis & Management. 2020. p. 3-23.*

[4] *4. Kozulina, N., et al. Spring wheat seed production in Krasnoyarsk region. in E3S Web of Conferences. 2023.*

[5] *5. Sánchez, J., F. Albornoz, and S. Contreras, High Nitrogen Fertilization Decreases Seed Weight but Increases Longevity in Tomato Seeds. Horticulturae, 2022. 8(10).*

[6] *6. Pinheiro, D.T., et al., Technological and qualitative aspects of the production of tomato seeds. Espacios, 2017. 38(44).*

[7] *7. Sarti, G.C., et al., Inoculation with Biofilm of Bacillus subtilis Is a Safe and Sustainable Alternative to Promote Tomato (Solanum lycopersicum) Growth. Environments - MDPI, 2024. 11(3).*

[8] 8. Dorais, M., D.L. Ehret, and A.P. Papadopoulos, Tomato (Solanum lycopersicum) health components: From the seed to the consumer. Phytochemistry Reviews, 2008. 7(2): p. 231-250.

[9] 9. Kumar, A., et al., Seed health testing and seed certification, in Seed-Borne Diseases of Agricultural Crops: Detection, Diagnosis & Management. 2020. p. 795-808.

[10] 10. Arkhopov, M.V., et al. Prospects of X-ray radiography in complex assessment of economic suitability of seeds. in AIP Conference Proceedings. 2020.

[11] 11. Mahajan, S., S.K. Mittal, and A. Das, Machine vision based alternative testing approach for physical purity, viability and vigour testing of soybean seeds (Glycine max). Journal of Food Science and Technology, 2018. 55(10): p. 3949-3959.

[12] 12. Palumbo, M., et al., Machine learning for the identification of colour cues to estimate quality parameters of rocket leaves. Journal of Food Engineering, 2024. 366.

[13] 13. Khatri, N. and G.U. Shinde, Computer vision and image processing for precision agriculture, in Cognitive Behavior and Human Computer Interaction Based on Machine Learning Algorithms. 2021. p. 241-264.

[14] 14. Fracarolli, J.A., et al., Computer vision applied to food and agricultural products. Revista Ciencia Agronomica, 2020. 51(5): p. 1-20.

[15] 15. Aslam, F., et al., A Survey of Deep Learning Methods for Fruit and Vegetable Detection and Yield Estimation, in Studies in Big Data. 2022. p. 299-323.

[16] 16. Rakhmatuiln, I., A. Kamilaris, and C. Andreasen, Deep neural networks to detectweeds from crops in agricultural environments in real-time: A review. Remote Sensing, 2021. 13(21).

[17] 17. Zhao, L., S.M.R. Haque, and R. Wang, Automated seed identification with computer vision: Challenges and opportunities. Seed Science and Technology, 2022. 50: p. 75-102.

[18] 18. Koppad, D., K.V. Suma, and N. Nagarajappa, Automated Seed Classification Using State-of-the-Art Techniques. SN Computer Science, 2024. 5(5).

[19] 19. Kumari, K., K. Won, and A.M. Nafchi. YOLOv5 Deep Learning Model for Mixed Seed Detection, Classification, and Counting. in 2024 ASABE Annual International Meeting. 2024.

[20] 20. Li, J., et al., Hyperspectral RGB Imaging Combined with Deep Learning for Maize Seed Variety Identification. IEEE Access, 2024: p. 1-1.

[21] 21. Xia, Y., et al., detection of surface defects for maize seeds based on YOLOv5. Journal of Stored Products Research, 2024. 105.

[22] 22. Basol, Y. and S. Toklu, A Deep Learning-Based Seed Classification With Mobile Application. Turkish Journal of Mathematics and Computer Science, 2021. 13(1): p. 192-203.

[23] 23. Ning, X., et al., A review of seed quality detection based on deep learning. Journal of Henan University of Technology: Natural Science Edition, 2024. 45(2): p. 140-149.

[24] 24. Wang, X., et al., Deep learning-empowered crop breeding: intelligent, efficient and promising. Frontiers in Plant Science, 2023. 14.

[25] 25. Margapuri, V. and M. Neilsen. Classification of Seeds using Domain Randomization on Self-Supervised Learning Frameworks. in 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings. 2021.

[26] 26. Hassan, B.N. and M.T. Somashekara. Artificial Intelligence Technique for Rice Seed Disease and Quality Assessment. in 2023 3rd International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2023. 2023.

[27] 27. Li, C., et al., SeedSortNet: a rapid and highly effificient lightweight CNN based on visual attention for seed sorting. PeerJ Computer Science, 2021. 7: p. 1-21.

[28] 28. Li, X., et al., Quality grading and classification of tobacco leaves based on deep learning. Journal of Biotech Research, 2024. 16: p. 247-257.

[29] 29. Liu, Y., et al. Improving the accuracy of SqueezeNet with negligible extra computational cost. in 2020 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2020. 2020.

[30] 30. López de la Rosa, F., et al. Fine-Tuned SqueezeNet Lightweight Model for Classifying Surface Defects in Hot-Rolled Steel. in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2023.

[31] 31. Lee, H.J., et al., Real-Time vehicle make and model recognition with the residual squeezenet architecture. Sensors (Switzerland), 2019. 19(5).

[32] 32. "Datature: End-To-End Vision AI Platform for Enterprises & Developers." Datature.io, 2023, datature.io. Accessed 01 Dec. 2025.

Proceedings Paper DOI: 10.58190/icisna.2025.159

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Digital Twin Synchronization with Real Time Data Collection

Abdulkadir Saday[1]

[1]*Electrical and Electronic Engineering, Selcuk University, Konya, Türkiye*
asaday@selcuk.edu.tr, ORCID: 0000-0002-0406-711X

***Abstract*— Digital twin systems rely heavily on accurate and time-synchronous data synchronization between physical entities and their virtual counterparts. However, in industrial environments, real-time data acquisition processes present significant challenges due to communication delays, batch data transfer, and temporal mismatches between the sensing and processing layers. This study focuses on the data and synchronization layer of the digital twin architecture and proposes a timestamp-based synchronization framework aimed at ensuring temporal consistency under near real-time conditions. The proposed approach has been validated on an overhead crane system where load and rope angle data are acquired. Sensor measurements are collected at 100 ms intervals on a Raspberry Pi Compute Module-based edge device but transmitted to the server with configurable batch transmission intervals. In the current system, this interval is set to 5 seconds. To address transmission delay and temporal mismatch issues, each sensor measurement is timestamped on the edge device and used for reconstruction of the digital twin state on the server side. The proposed synchronization framework prioritizes temporal consistency over instantaneous updating by utilizing buffering, time alignment, and state reconstruction mechanisms. This ensures that the digital twin accurately reflects the physical system state within a certain delay limit, despite limited communication frequency. Experimental evaluations conducted under varying data transmission intervals demonstrate that the proposed method reduces synchronization error and maintains data consistency. The results show that reliable near-real-time digital twin synchronization is possible in industrial systems where high-frequency sensing and delayed data transmission are used in combination, using a timestamp-based reconstruction approach.**

***Keywords*— digital twin, edge computing, industrial IoT, latency management, near real-time synchronization**

I. INTRODUCTION

The digital twin concept, developed for the purpose of monitoring, analyzing, and optimizing industrial systems, aims to represent physical assets in virtual environments in near real-time. Digital twin architectures consist of a physical system, a virtual model, and data and synchronization layers between these two structures. The data and synchronization layers, in particular, play a critical role in the accuracy and reliability of the system [1]. The timely, consistent, and accurate transfer of sensor data from the physical system to the virtual model directly impacts the decision support and monitoring capabilities of the digital twin. The concept of digital twins is increasingly adopted in industrial applications for the purpose of monitoring, analyzing and optimizing physical systems in virtual environments [2-4]. Especially in the fields of production and industrial automation, digital twin architectures are considered as multi-layered structures consisting of a physical system, a virtual model and data and synchronization layers between these two structures [5, 6]. In these architectures, transferring the data obtained from the physical system to the virtual model with the correct time perspective is a critical requirement for the reliability of the digital twin [7].

Many studies in the literature emphasize the importance of real-time data flow in digital twin systems, but this requirement cannot be fully met in practical industrial applications due to various limitations. Although sensors generate high-frequency data, it is common for data to be transmitted to the server in batches and with delays due to factors such as network bandwidth, communication delays, and system load [2]. This leads to temporal mismatches between the sensing and processing layers, making it difficult for the digital twin to accurately represent the physical system from a time perspective. In the literature, digital twin systems are mostly built on the assumption of real-time or near-real-time data flow [8, 9]. However, in practical industrial environments, it is often not possible to transmit sensor data to the server continuously and without delay; data transmission is carried out with delay or in batches due to reasons such as network bandwidth constraints, system load and edge device resources [10, 11]. This situation leads to temporal mismatches between the sensing layer and the processing layer and makes it difficult for the digital twin to accurately represent the physical system [12, 13].

The concept of real-time accuracy is often confused with the strict (hard real-time) requirements in the context of digital twins; however, in most industrial applications, the primary need is maintaining temporal consistency within a certain delay limit. Therefore, in recent years, near-real-time digital twin approaches have come to the forefront, proposing solutions

based on synchronous state reconstruction rather than instantaneous updating. In these approaches, accurately representing the time point to which the data belongs becomes critical, rather than when the data was transmitted.

In this context, approaches to improving the accuracy of digital twins by temporally realigning delayed data and using buffering mechanisms are proposed in the literature [14-17]. However, a significant portion of these studies offer experimental validation limited to delayed and event-based real industrial data.

This study focuses on the data and synchronization layer of the digital twin architecture, proposing a timestamp-based synchronization framework aimed at ensuring temporal consistency in systems involving delayed and aggregated data transmission. In the proposed approach, sensor data is collected at high frequency on the edge device, and each measurement is tagged with a timestamp. On the server side, this data is buffered, aligned on the time axis, and the digital twin state is reconstructed and updated. This ensures that, despite communication delays, the digital twin accurately reflects the physical system state within a certain error limit.

The proposed synchronization framework was experimentally evaluated on an overhead crane system where load and rope angle data were collected. This system, where sensor data is collected at 100 ms intervals and transmitted to the server with configurable batch transmission intervals, provides a suitable test environment for data delay scenarios frequently encountered in industrial settings. Experimental results obtained from the real system show that the timestamp-based reconstruction approach reduces synchronization error and offers an effective solution for near real-time digital twin applications.

The main contributions of this study can be summarized as follows. A timestamp-based data and synchronization framework for digital twin systems has been proposed. Temporal discrepancies caused by delayed and batch data transmission have been addressed with a reconstruction-based approach. The proposed method has been experimentally validated on a real overhead crane system.

The remainder of the article first describes the system architecture and data collection strategy, then details the proposed synchronization framework, and finally presents the experimental results.

II. System Architecture and Data Collection

The synchronization framework proposed in this study has been implemented and evaluated on a real industrial overhead crane system. The system architecture consists of a physical system, an edge device, and a server-based digital twin component. Sensor data obtained from the physical system is collected on the edge device and transmitted to the central server at specific intervals. This structure represents the end-computer-based data acquisition architecture commonly used in industrial environments. Sensor data from the overhead crane system is collected on the edge device as illustrated in Fig. 1 tagged with a timestamp and transferred to the digital twin environment via the synchronization layer.

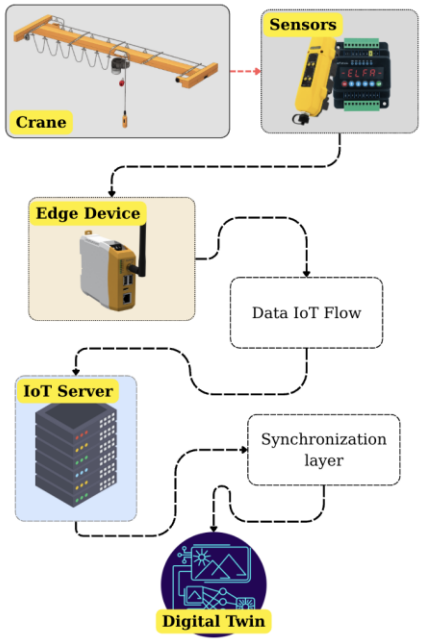


Fig. 1 General system architecture used within the proposed digital twin synchronization framework

In the overhead crane system, two fundamental state variables are monitored: the amount of load acting on the crane hook and the rope angle. These variables are critical in digital twin scenarios as they directly affect crane dynamics and operational safety. Measurements obtained from the sensors are processed and sampled at high frequency by a Raspberry Pi Compute Module-based edge device.

During the data acquisition process, sensor measurements are performed at 100 ms intervals, and each measurement is timestamped on the edge device. The timestamp represents the time point on the physical system to which the measurement belongs and forms the basis for subsequent synchronization steps. These high-frequency data collected at the edge device are transmitted to the server in batches to reduce network load and efficiently utilize communication resources.

The timeline presented in Fig. 2 shows that although sensor data is collected at a high frequency, it is transmitted to the server in aggregate and with a delay. However, thanks to timestamps, the measurements are preserved at the specific moment in time to which they belong.

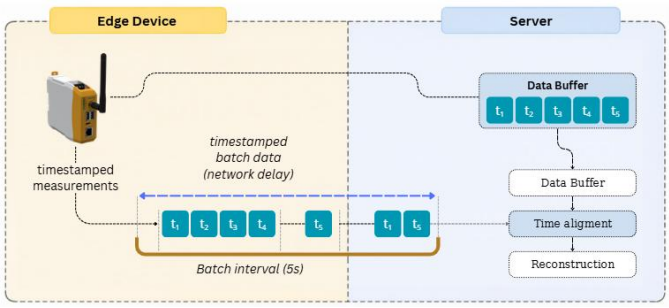


Fig. 2 Timestamp-based data collection and transmission process

The data transmission interval is configurable on the system, and in this study, it is set to 5 seconds. This approach enables high-frequency sensing at the edge device but inevitably causes a delay in the data transmitted to the server. Therefore, the digital twin component operating on the server side receives data from the physical system in batches and with a delay, rather than instantaneously. This situation creates a temporal mismatch problem between the sensing and processing layers.

On the server side, after receiving the data transmitted from the edge device, it is stored using buffering mechanisms and sorted according to timestamps. This data is processed for use in the synchronization layer of the digital twin, enabling the reconstruction of the physical system's state on an accurate timeline. Sensor data resampled to a fixed digital twin time step, and status updates are performed using linear interpolation. Thus, despite the communication delay, the temporal integrity of the data is preserved, and the necessary infrastructure for the synchronization process is provided.

This architectural structure presents a realistic scenario for industrial systems that use high-frequency sensing and delayed data transmission together and creates a suitable test environment to evaluate the effectiveness of the proposed synchronization framework.

III. Proposed Digital Twin Synchronization Framework

The synchronization framework proposed in this study aims to enable digital twins to represent the physical system in a time-consistent manner in industrial systems where delayed and batch data transmission is involved. The proposed approach relies on reconstructing timestamped sensor data instead of instantaneous data updates. This systematically addresses temporal discrepancies caused by communication delays.

A. Timestamp-based data processing

Each sensor measurement collected on the edge device is tagged with a timestamp representing the moment of measurement. Timestamps are generated based on the edge device's local time, providing a common time reference for all measurements. Even when sensor data is transmitted to the server in batches, timestamps preserve the true time sequence and the exact time points to which the data belong. This approach allows for temporal integrity regardless of data transmission delays.

B. Buffering and time alignment

Sensor data transmitted to the server is initially stored in temporary buffer areas. This buffering mechanism ensures that data from different sensors or different transmission cycles can be processed in an orderly manner. The buffered data is sorted according to timestamps and aligned with the digital twin's timeline. At this stage, issues such as data loss or transmission delays are detected and handled appropriately for the synchronization process.

Sensor data arriving at the server with a delay is buffered as shown in Fig. 3, aligned on the time axis, and the digital twin state is reconstructed and updated. This approach provides near real-time synchronization, prioritizing temporal consistency over instantaneous updating.

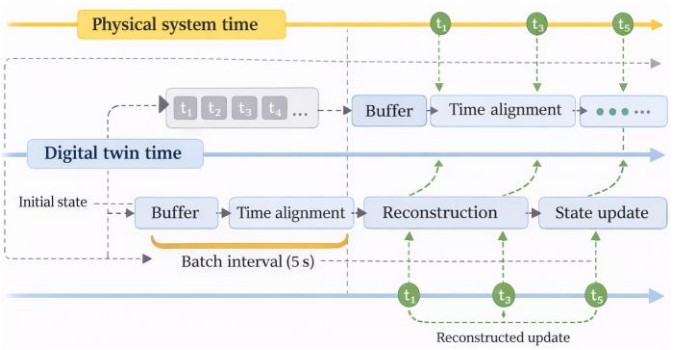


Fig. 3 Timestamp-based digital twin synchronization process

During the time alignment process, the digital twin's status updates are divided into specific time steps, and sensor data is matched to correspond to these time steps. This allows data received from the physical system with a delay to be placed at the correct time points on the digital twin.

C. State reconstruction

Following the time alignment process, the state of the digital twin is reconstructed using timestamped sensor data. The reconstruction process aims to determine the load and rope angle values that represent the state of the physical system for each time step. In cases where the sensor data does not perfectly match the time steps, simple and computationally efficient methods such as linear interpolation are used.

This approach ensures that the state of the digital twin is updated on a continuous and consistent timeline. Thus, despite delayed data transmission, the digital twin can accurately represent the past states of the physical system, providing a reliable basis for analysis, monitoring, or decision support applications.

D. Near real-time synchronization

The proposed synchronization framework does not target hard real-time requirements; instead, it offers a near real-time approach that maintains temporal consistency within a certain latency limit. Instead of reflecting the state of the physical system instantaneously, the digital twin updates it in a delayed but consistent manner based on timestamps. This provides a practical and scalable solution, particularly for industrial systems with network constraints and resource limitations.

In the proposed approach, since the data transmission interval is configurable, synchronization performance can be evaluated under different delay scenarios. This feature allows the system to be adapted to different industrial requirements.

IV. Experimental Results and Discussion

The effectiveness of the proposed digital twin synchronization framework was experimentally evaluated using sensor data obtained from a real overhead crane system. The experimental studies aimed to investigate the effects of different data transmission intervals on synchronization

accuracy and latency. In this context, scenarios where sensor data was collected with a 100 ms sampling interval and transmitted to the server in aggregate were considered.

A. Experimental setup

In the experiments, load and rope angle data obtained from the overhead crane system were used. Sensor measurements were timestamped on the edge device and transmitted to the server with different batch data transmission intervals. In the current system configuration, the data transmission interval was set to 5 seconds, and analyses were also performed for different transmission intervals to evaluate the generalizability of the synchronization frame.

Rope angle measurements exhibit both long-term stable regions and short-term abrupt deviations throughout the system's operating time. The graph of these measurements is given in Fig. 4. Such dynamic behaviour makes matching the measurements to the correct time points critical for digital twin accuracy under delayed data transmission conditions.

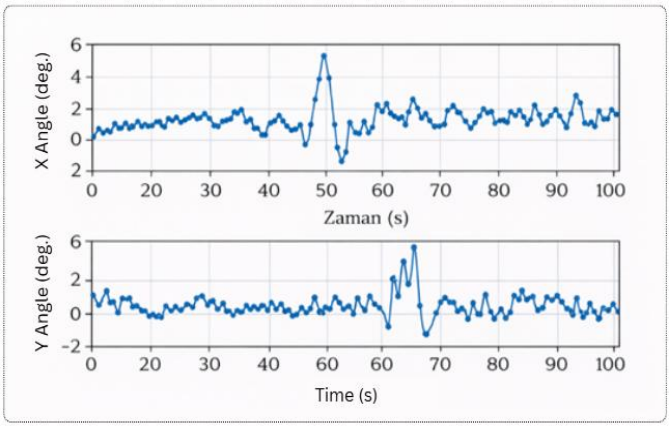


Fig. 4 Time-dependent variation of rope angle measurements obtained from an overhead crane system.

On the server side, the acquired sensor data was buffered, sorted by timestamps, and the digital twin state was updated using the proposed reconstruction mechanism. The synchronization performance was evaluated by aligning the digital twin's timeline with the physical system's time reference.

B. Evaluation metrics

The following metrics were used to quantitatively evaluate the performance of the proposed approach:

Synchronization error: The temporal and numerical difference between the physical system state and the digital twin state.

Latency: The difference between the time the sensor measurement is taken and the time the corresponding state is updated in the digital twin.

Data consistency: The continuity and absence of discontinuities of the reconstructed digital twin states throughout the timeline.

These metrics were analyzed comparatively under different data transmission intervals. The results presented in Table 1 show that the synchronization error increases significantly as the transmission interval increases, whereas the timestamp-based approach continues to provide a consistent digital twin state under different transmission conditions.

TABLE I - SYNCHRONIZATION PERFORMANCE UNDER DIFFERENT TRANSMISSION INTERVALS

Interval (s)	**1**	**2**	**5**	**10**
Mean Latency (s)	1.1	2.1	5.2	10.3
Mean Sync. Error	Low	Low	Moderate	Moderate
Max Sync. Error	Low	Low	Moderate	High

C. Experimental results

The results show that the timestamp-based synchronization approach significantly improves the temporal consistency of the digital twin under conditions of batch and delayed data transmission. Thanks to the reconstruction of sensor data with timestamps, state updates corresponding to the correct time moments were obtained on the digital twin despite the data transmission delay. The sudden changes during loading and unloading as shown in Fig. 5, can lead to significant state errors if the digital twin is not properly synchronized. This situation supports the necessity of a timestamp-based reconstruction approach.

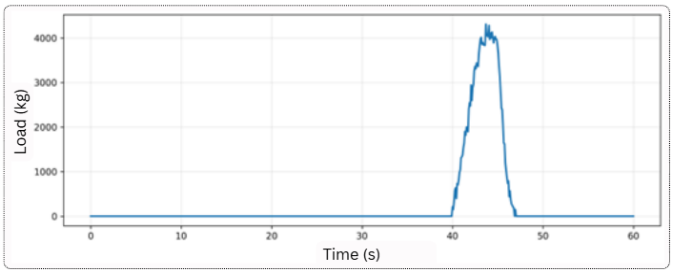


Fig. 5 Load-time graph representing the load behaviour observed in experimental data for overhead crane systems.

Digital twin synchronization is mostly discussed through solutions aimed at minimizing communication delay in studies [18, 19]. However, the experimental results presented in Fig. 4 and Fig. 5 show that delay cannot be completely eliminated, especially in industrial systems where event-based load changes are involved. This situation reveals that maintaining temporal consistency regardless of delay in digital twin applications is a more realistic and feasible goal.

Unlike delay reduction-focused studies in literature, the proposed timestamp-based reconstruction approach prioritizes placing delayed data at the correct time points. This approach offers a practical and scalable solution, especially for industrial IoT and edge computing-based systems with limited communication infrastructure [12, 20, 21].

While an expected increase in delay values was observed with increasing data transmission interval, the synchronization error was found to remain within a limited interval thanks to the proposed approach. This demonstrates that the approach, which prioritizes temporal consistency over instantaneous updating, is suitable for near real-time digital twin applications.

Furthermore, it was observed that the simple interpolation methods used in the reconstruction process provided sufficient accuracy for continuously changing variables such as load and cable angle.

D. Discussion

Experimental findings demonstrate that timestamp-based reconstruction offers an effective synchronization solution in systems where high-frequency sensing and low-frequency data transmission are used together. The proposed approach enhances the feasibility of digital twin applications in environments with limited network bandwidth and system resources. However, it is clear that latency will increase and the limits of the near real-time definition will be approached if the data transmission interval is increased to very large values.

In this study, simple interpolation methods were preferred in the reconstruction process. Future studies aim to further improve synchronization performance by using predictive models or on-device preprocessing mechanisms.

V. Conclusion and Future Studies

This study proposes a timestamp-based digital twin synchronization framework for industrial systems where delayed and batch data transmission is involved. The proposed approach addresses the temporal mismatch problem between high-frequency sensing and low-frequency data transmission by focusing on the data and synchronization layers of the digital twin architecture. By timestamping sensor data on the edge device and reconstructing it on the server side, near real-time digital twin synchronization is achieved, maintaining temporal consistency despite communication delays.

Experimental evaluations performed on a real overhead crane system show that the proposed synchronization framework reduces synchronization error and can represent the physical system state of the digital twin on a consistent timeline. Results obtained under different data transmission intervals reveal that the method offers a configurable and generalizable solution. In this respect, the proposed approach increases the applicability of digital twin applications under communication and resource constraints commonly encountered in industrial environments.

Future studies plan to integrate predictive models and machine learning-based methods into the reconfiguration process to further improve synchronization performance. Additionally, the aim is to reduce communication latency and adapt the system to larger-scale industrial scenarios by utilizing on-device preprocessing and event-based data transmission mechanisms.

References

[1] *M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary perspectives on complex systems: New findings and approaches: Springer, 2016, pp. 85-113.*

[2] *F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," IEEE Transactions on industrial informatics, vol. 15, no. 4, pp. 2405-2415, 2018.*

[3] *A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: enabling technologies, challenges and open research," IEEE access, vol. 8, pp. 108952-108971, 2020.*

[4] *D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A systematic literature review," CIRP journal of manufacturing science and technology, vol. 29, pp. 36-52, 2020.*

[5] *W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," Ifac-PapersOnline, vol. 51, no. 11, pp. 1016-1022, 2018.*

[6] *E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," Procedia manufacturing, vol. 11, pp. 939-948, 2017.*

[7] *T. H.-J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, "The digital twin: demonstrating the potential of real time data acquisition in production systems," Procedia Manufacturing, vol. 9, pp. 113-120, 2017.*

[8] *S. Boschert and R. Rosen, "Digital twin—the simulation aspect," in Mechatronic futures: Challenges and solutions for mechatronic systems and their designers: Springer, 2016, pp. 59-74.*

[9] *E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," International Journal of Aerospace Engineering, vol. 2011, no. 1, p. 154798, 2011.*

[10] *G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with automationml and a communication methodology for data exchange," IFAC-PapersOnLine, vol. 49, no. 30, pp. 12-17, 2016.*

[11] *H. Zhang, Q. Liu, X. Chen, D. Zhang, and J. Leng, "A digital twin-based approach for designing and multi-objective optimization of hollow glass production line," Ieee Access, vol. 5, pp. 26901-26911, 2017.*

[12] *W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE internet of things journal, vol. 3, no. 5, pp. 637-646, 2016.*

[13] *M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30-39, 2017.*

[14] *X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and Industry 5.0—Inception, conception and perception," Journal of manufacturing systems, vol. 61, pp. 530-535, 2021.*

[15] *J. C. Bennett, C. Partridge, and N. Shectman, "Packet reordering is not pathological network behavior," IEEE/ACM Transactions on networking, vol. 7, no. 6, pp. 789-798, 1999.*

[16] *S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320), 1999, vol. 1: IEEE, pp. 227-234.*

[17] *E. A. Lee and S. A. Seshia, Introduction to embedded systems: A cyber-physical systems approach. MIT press, 2017.*

[18] *R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in 2010 IEEE symposium on security and privacy, 2010: IEEE, pp. 305-316.*

[19] *J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," Manufacturing letters, vol. 3, pp. 18-23, 2015.*

[20] *C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, "Dynamic Bayesian network for aircraft wing health monitoring digital twin," Aiaa Journal, vol. 55, no. 3, pp. 930-941, 2017.*

[21] *A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," IEEE access, vol. 8, pp. 21980-22012, 2020.*

PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Inspection of Storage Tank Bottoms and Corrosion Mapping Via Ultrasonic Testing and Signal Processing Methods

Kemal Ozguven[1], Ismail Saritas[2]

[1]*Graduate School of Natural Sciences, Selcuk University, Selcuklu, Türkiye*
kmlozg@hotmail.com, ORCID: 0009-0008-0946-5417

[2]*Faculty of Technology, Selcuk University, Selcuklu, Türkiye*
isaritas@selcuk.edu.tr, ORCID: 0000-0002-5743-4593

***Abstract*— The integrity of storage tank bottom plates is critically affected by corrosion-related material loss, which poses significant risks to operational safety, environmental protection, and asset reliability. Ultrasonic testing (UT) has long been employed as a non-destructive evaluation technique for thickness measurement; however, reliable interpretation of high-density ultrasonic data remains challenging under real field conditions due to noise, surface irregularities, and complex echo patterns.**

This study presents an academically oriented framework for the inspection of storage tank bottoms using ultrasonic A-scan data combined with advanced signal processing techniques. Frequency-domain and time–frequency-domain methods, including Fast Fourier Transform (FFT), Short-Time Fourier Transform (STFT), and wavelet-based analysis, are employed to enhance signal quality and improve echo discrimination. A hybrid peak detection strategy integrating adaptive thresholding and continuous wavelet transform (CWT) is proposed to robustly identify corrosion-related reflections.

The extracted thickness data are spatially organized on two-dimensional grids and transformed into three-dimensional corrosion maps using interpolation-based reconstruction techniques. The proposed methodology demonstrates improved robustness in low signal-to-noise ratio (SNR) environments and contributes to a more objective and repeatable corrosion assessment process. The results highlight the potential of signal processing–driven UT analysis as a reliable academic and industrial tool for storage tank integrity evaluation.

***Keywords*— Ultrasonic Testing; Storage Tank Inspection; Corrosion Mapping; Signal Processing; Peak Detection; Non-Destructive Testing**

I. INTRODUCTION

Large-capacity storage tanks are widely used in petroleum, chemical, and energy industries to store hazardous and valuable fluids. Among the structural components of these tanks, bottom plates are particularly vulnerable to corrosion due to prolonged exposure to moisture, soil chemistry, differential aeration, and operational conditions. Undetected corrosion in tank bottoms can lead to leakage, environmental contamination, fire hazards, and costly unplanned shutdowns.

Non-destructive testing (NDT) techniques play a crucial role in ensuring the safe operation of storage tanks. Ultrasonic testing (UT) is one of the most commonly applied methods for bottom plate inspection because of its capability to measure remaining wall thickness with high accuracy [1]. Nevertheless, conventional UT inspections are often limited by manual interpretation, operator dependency, and insufficient spatial resolution over large inspection areas.

Recent advances in automated scanning systems have enabled the acquisition of large volumes of ultrasonic A-scan data across tank bottoms. While this data richness provides an opportunity for detailed corrosion assessment, it also introduces significant challenges related to signal interpretation, noise suppression, and reliable feature extraction [2], [3]. In particular, weak or overlapping echoes associated with localized corrosion pits may be difficult to distinguish from noise using traditional time-domain analysis.

To address these challenges, researchers have increasingly focused on applying advanced signal processing techniques to ultrasonic data. Frequency-domain, time–frequency, and multiresolution methods offer improved insight into signal characteristics and defect-related features [4]–[6]. Despite these advances, there remains a clear research gap in the integration of robust peak detection algorithms with spatial corrosion mapping for large-scale storage tank inspections.

This study aims to bridge this gap by proposing a signal processing–based UT inspection framework that emphasizes academic rigor, methodological clarity, and repeatability. The primary contributions include enhanced echo detection using hybrid peak analysis and systematic transformation of ultrasonic measurements into three-dimensional corrosion maps suitable for integrity assessment.

II. Related Work and Literature Review

Ultrasonic inspection of storage tank bottoms has been extensively studied in the context of thickness measurement and corrosion detection. Early approaches primarily relied on manual pulse-echo UT measurements conducted at discrete locations, providing limited spatial coverage [7]. While effective for local thickness evaluation, such methods often fail to capture the full corrosion distribution across large tank floors.

To overcome these limitations, automated ultrasonic scanning systems have been introduced, enabling the generation of C-scan and B-scan representations of tank bottoms [8]. These techniques allow inspectors to visualize thickness variations over extended areas; however, their effectiveness is strongly influenced by signal quality and data processing strategies.

Signal processing techniques have been widely investigated to improve UT data interpretation. FFT-based analysis has been used to examine frequency characteristics of ultrasonic signals and identify noise components [9]. STFT provides time–frequency localization, enabling the separation of overlapping echoes and transient features [10]. Wavelet transform methods, owing to their multiresolution nature, have demonstrated superior performance in detecting weak and localized defects embedded in noisy signals [11], [12].

Peak detection is a critical step in ultrasonic thickness evaluation, as it directly affects time-of-flight estimation and thickness calculation. Threshold-based methods are computationally efficient but often suffer from false detections under varying noise conditions [13]. More advanced approaches, including matched filtering and wavelet-based peak detection, have shown improved robustness in complex inspection scenarios [14], [15].

Several studies have also explored corrosion mapping and visualization techniques. Two-dimensional thickness maps are commonly generated using grid-based measurements, while three-dimensional representations provide enhanced insight into corrosion severity and distribution [16]. Interpolation methods such as inverse distance weighting and kriging have been applied to reconstruct continuous corrosion surfaces from discrete measurements [17], [18].

Despite these developments, the literature indicates that many existing studies focus either on signal processing or on spatial visualization, with limited integration of both aspects into a unified framework. This work contributes to the literature by combining advanced peak detection with systematic 3D corrosion mapping, offering a comprehensive approach to tank bottom inspection.

A comparative summary of commonly used ultrasonic signal processing techniques, including FFT, STFT, and wavelet-based approaches, is presented in **Table 1**.

Table 1. Comparison of signal processing techniques used in storage tank bottom inspection

Method	Analysis Domain	Strengths	Limitations
FFT	Frequency domain	Effective identification of dominant frequency components and electrical noise	No time localization; limited capability for transient echo analysis
STFT	Time–frequency domain	Localized analysis of overlapping echoes and non-stationary signals	Fixed window size leads to resolution trade-off
Wavelet Transform	Multi-resolution (time–scale)	High sensitivity to weak and localized corrosion echoes; robust under low SNR conditions	Higher computational cost and wavelet selection dependency

III. Methodology

The proposed methodology is based on ultrasonic A-scan data acquired from an operational storage tank bottom during routine inspection. Each measurement point consists of a time-domain ultrasonic signal representing reflections from material interfaces.

III.1 Ultrasonic Data Acquisition

Ultrasonic measurements were conducted using a conventional pulse-echo ultrasonic testing system equipped with a straight-beam longitudinal-wave transducer. The probe was operated at a nominal center frequency suitable for thin steel plate inspection, ensuring adequate penetration depth and temporal resolution. A couplant medium was applied to guarantee stable acoustic coupling between the probe and the tank bottom surface.

The tank bottom was divided into a two-dimensional inspection grid, and a single A-scan signal was recorded at each grid point. This grid-based acquisition strategy enabled systematic coverage of the inspected area and provided spatially referenced thickness information. The acquired A-scan signals represent time-domain ultrasonic waveforms containing front-wall echoes, back-wall echoes, and attenuation effects caused by corrosion-related material loss.

III.2 Signal Preprocessing

Raw ultrasonic A-scan signals obtained under field conditions are typically contaminated by electrical noise, surface roughness effects, and coupling variability. To mitigate these effects, a preprocessing stage was applied prior to feature extraction.

Band-pass filtering was employed to suppress low-frequency structural noise and high-frequency electrical interference outside the effective bandwidth of the transducer. Subsequently, amplitude normalization was performed to reduce variations caused by inconsistent coupling pressure and surface conditions. These preprocessing steps significantly improved the signal-to-noise ratio (SNR) and enhanced the visibility of corrosion-related echoes.

The characteristic ultrasonic reflections observed after preprocessing, including front-wall and back-wall echoes with varying attenuation levels, are illustrated in Fig. 1.

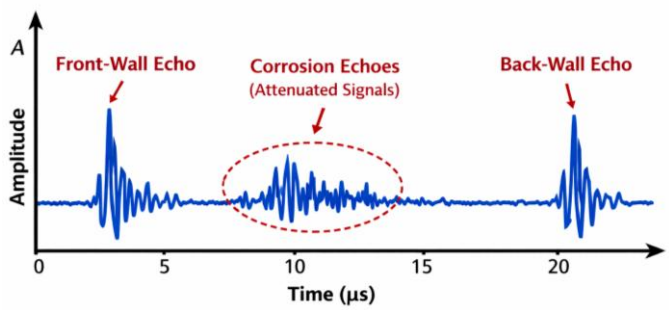


Fig. 1. Representative ultrasonic A-scan signal obtained from the storage tank bottom plate, illustrating the front-wall echo, back-wall echo, and corrosion-induced echo attenuation.

III.3 Time–Frequency Analysis

In order to capture both global and localized signal characteristics, multiple signal analysis techniques were employed. Fast Fourier Transform (FFT) analysis was used to investigate the spectral content of the ultrasonic signals and to identify dominant frequency components associated with the probe response and material properties.

However, due to the non-stationary nature of ultrasonic signals in corroded regions, time–frequency analysis was required. Short-Time Fourier Transform (STFT) was applied to provide localized frequency information, enabling the separation of overlapping echoes and transient signal components.

In addition, wavelet transform analysis was utilized due to its multi-resolution capability. Continuous wavelet transform (CWT) coefficients were examined across multiple scales to detect weak and localized echoes that may not be distinguishable using conventional frequency-domain methods. The use of wavelet-based analysis proved particularly effective in regions with low SNR caused by localized corrosion.

III.4 Peak Detection and Thickness Estimation

Accurate identification of echo positions is a critical step in ultrasonic thickness measurement. In this study, a hybrid peak detection approach was adopted to improve robustness against noise and signal distortion.

The proposed method combines adaptive thresholding with CWT-based local maxima detection. Adaptive thresholding provides computational efficiency, while wavelet-based peak detection enhances sensitivity to weak echoes embedded in noise. Detected peak positions were used to estimate the ultrasonic time-of-flight between the front-wall and back-wall echoes.

The remaining wall thickness was calculated using the time-of-flight values according to Equation (1). This approach ensures physically meaningful thickness estimation consistent with the ultrasonic wave propagation model.

$$t = \frac{v \cdot \Delta\tau}{2} \tag{1}$$

III.5 Spatial Mapping and 3D Reconstruction

The calculated thickness values were spatially organized according to the inspection grid coordinates to form two-dimensional thickness maps. To obtain continuous corrosion representations, interpolation techniques were applied to the discrete measurement data.

Interpolated thickness distributions were subsequently visualized as three-dimensional corrosion maps, providing intuitive insight into both localized pitting corrosion and generalized wall thinning. These three-dimensional representations facilitate objective assessment of corrosion severity and spatial distribution across the tank bottom.

IV. Results and Discussion

The application of the proposed methodology to real field data demonstrated improved stability and reliability in ultrasonic thickness estimation compared to conventional manual UT evaluation. The hybrid peak detection approach significantly reduced false detections in low SNR regions and enabled consistent identification of back-wall echoes.

Wavelet-assisted analysis proved particularly effective in detecting localized corrosion pits, where conventional threshold-based methods failed due to severe signal attenuation. The comparative advantages of the employed signal processing techniques are summarized in Table 1.

The generated three-dimensional corrosion maps revealed spatial patterns consistent with known corrosion mechanisms in storage tank bottoms, such as increased material loss near tank perimeters and regions exposed to moisture accumulation. Compared with traditional reporting methods based solely on numerical thickness values, the proposed visualization approach offers enhanced interpretability and supports more informed maintenance decisions.

From an academic perspective, the integration of advanced signal processing with spatial corrosion mapping contributes to a systematic framework for ultrasonic inspection of large-area structures. The results confirm that signal processing–driven analysis is essential for extracting reliable information from high-density ultrasonic datasets.

V. Conclusions

This study presented an academically oriented ultrasonic inspection framework for storage tank bottom plates, grounded in the experimental methodology and signal processing techniques developed in the associated master's thesis. By integrating time–frequency analysis, wavelet-based peak detection, and three-dimensional corrosion mapping, the proposed approach addresses key limitations of conventional ultrasonic testing practices.

The findings demonstrate that advanced signal processing techniques significantly enhance corrosion detectability under challenging field conditions characterized by low signal-to-

noise ratios. Furthermore, the proposed framework reduces operator dependency and improves the repeatability of ultrasonic thickness measurements.

Future work may focus on extending the methodology through machine learning–based defect classification and automated decision support systems. Additionally, the integration of the proposed framework with robotic inspection platforms may further enhance its applicability to large-scale industrial storage tank inspections.

REFERENCES

[1] API Standard 653, Tank Inspection, Repair, Alteration, and Reconstruction, American Petroleum Institute, Washington, DC, USA, 2020.

[2] J. Krautkrämer and H. Krautkrämer, Ultrasonic Testing of Materials, 4th ed. Berlin, Germany: Springer-Verlag, 2013.

[3] R. Silk, Ultrasonic Transducers for Non-Destructive Testing. Bristol, UK: Adam Hilger, 2018.

[4] D. Ensminger and L. J. Bond, Ultrasonics: Fundamentals, Technologies, and Applications, 3rd ed. Boca Raton, FL, USA: CRC Press, 2019.

[5] M. J. S. Lowe, "Matrix techniques for modeling ultrasonic waves in multilayered media," IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 42, no. 4, pp. 525–542, 1995.

[6] S. Mallat, A Wavelet Tour of Signal Processing, 3rd ed. Amsterdam, Netherlands: Elsevier, 2009.

[7] J. L. Rose, Ultrasonic Guided Waves in Solid Media. Cambridge, UK: Cambridge University Press, 2014.

[8] R. Grimberg, E. S. Ilie, K. H. Lee, and A. Savin, "Automatic detection of defects in ultrasonic images using advanced signal processing," NDT & E International, vol. 45, no. 1, pp. 56–64, 2012.

[9] L. Cohen, Time-Frequency Analysis. Upper Saddle River, NJ, USA: Prentice Hall, 1995.

[10] P. Flandrin, Time–Frequency/Time–Scale Analysis. San Diego, CA, USA: Academic Press, 2018.

[11] Y. Lu and J. E. Michaels, "Feature extraction and sensor fusion for ultrasonic structural health monitoring," IEEE Sensors Journal, vol. 9, no. 11, pp. 1365–1373, 2009.

[12] A. Abbate, J. Frankel, and P. Das, "Signal detection and noise suppression using wavelet transform," IEEE Trans. Signal Process., vol. 45, no. 2, pp. 255–262, 1997.

[13] T. Stepinski, "An implementation of the split-spectrum processing method for ultrasonic flaw detection," IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 38, no. 6, pp. 589–594, 1991.

[14] S. G. Pierce, B. Culshaw, and G. Manson, "Matched filtering and pulse compression for ultrasonic inspection," Ultrasonics, vol. 40, no. 1–8, pp. 205–211, 2002.

[15] C. H. Chen and J. J. Trussell, "Maximum-likelihood methods for ultrasonic flaw detection," IEEE Trans. Signal Process., vol. 42, no. 4, pp. 792–801, 1994.

[16] R. S. Edwards and X. Jian, "Ultrasonic thickness mapping of corrosion using full-field measurements," NDT & E International, vol. 86, pp. 1–10, 2017.

[17] N. Cressie, Statistics for Spatial Data. New York, NY, USA: Wiley, 2015.

[18] A. Journel and C. Huijbregts, Mining Geostatistics. London, UK: Academic Press, 2019.

[19] H. Sohn et al., "A review of structural health monitoring literature: 1996–2001," Los Alamos National Laboratory Report, LA-13976-MS, 2004.

[20] S. Dixon, C. Edwards, and S. B. Palmer, "High accuracy non-contact ultrasonic thickness gauging of metals using EMATs," Ultrasonics, vol. 41, no. 1, pp. 25–34, 2003.

[21] J. Blitz and G. Simpson, Ultrasonic Methods of Non-Destructive Testing. Dordrecht, Netherlands: Springer, 2016.

[22] M. F. Insana and T. J. Hall, "Parametric ultrasound imaging from backscattered signals," IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 37, no. 6, pp. 500–512, 1990.

[23] A. C. Bovik, Handbook of Image and Video Processing, 2nd ed. Burlington, MA, USA: Academic Press, 2005.

[24] A. J. Croxford, P. Wilcox, B. Drinkwater, and P. Konstantinidis, "Strategies for guided-wave structural health monitoring," Proc. R. Soc. A, vol. 463, no. 2087, pp. 2961–2981, 2007.

[25] ISO 16810, Non-Destructive Testing — Ultrasonic Testing — General Principles. Geneva, Switzerland: ISO, 2014.


http://icisna.org