

e-ISSN: 2980-177X

JCEES

Volume: 4 Issue: 2 Year: 2026

**Journal of
Computer &
Electrical and
Electronics
Engineering
Sciences**

Journal of

Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

EDITOR-IN-CHIEF

Asst. Prof. Fuat TÜRK

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

ASSOCIATE EDITORS-IN-CHIEF

Assoc. Prof. Mahmut KILIÇASLAN

Department of Computer Technologies, Vocational School of Nallıhan, Ankara University, Ankara, Türkiye

Assoc. Prof. Selim BUYRUKOĞLU

Department of Computer Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Türkiye

Asst. Prof. Hüseyin AYDİLEK

Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Kırıkkale University, Kırıkkale, Türkiye

EDITORIAL BOARD

Prof. Aydın ÇETİN

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Prof. İsmail Rakıp KARAS

Department of Computer Engineering, Faculty of Engineering, Karabük University, Karabük, Türkiye

Prof. Necaattin BARIŞCI

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Prof. Necmi Serkan TEZEL

Department of Electrical and Electronics Engineering, Faculty of Engineering, Karabük University, Karabük, Türkiye

Assoc. Prof. Erinc KARATAŞ

Department of Computer Technologies, Vocational School of Elmadağ, Ankara University, Ankara, Türkiye

Assoc. Prof. Fatih KORKMAZ

Department of Electrical and Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Türkiye

Assoc. Prof. Hüseyin POLAT

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Assoc. Prof. İbrahim Alper DOĞRU

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Assoc. Prof. Levent GÖKREM

Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Tokat Gaziosmanpaşa University, Tokat, Türkiye

Assoc. Prof. Muhammet Tahir GÜNEŞER

Department of Electrical and Electronics Engineering, Faculty of Engineering, Karabük University, Karabük, Türkiye

Assoc. Prof. Murat LÜY

Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Kırıkkale University, Kırıkkale, Türkiye

Assoc. Prof. Ramazan Kürşat ÇEÇEN

Department of Aircraft Technology, Vocational School of Eskişehir, Eskişehir Osmangazi University, Eskişehir, Türkiye

Assoc. Prof. Sinan TOKLU

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Journal of

Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

Asst. Prof. Ayhan AKBAŞ

Department of Computer Engineering, Faculty of Engineering, Abdullah Gül University, Kayseri, Türkiye

Asst. Prof. Ebru AYDOĞAN DUMAN

Department of Computer Information Systems Engineering, Faculty of Bucak Technology, Burdur Mehmet Akif University, Burdur, Türkiye

Asst. Prof. Elvan DUMAN

Department of Software Engineering, Faculty of Bucak Technology, Burdur Mehmet Akif University, Burdur, Türkiye

Asst. Prof. Enes AYAN

Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Kırıkkale University, Kırıkkale, Türkiye

Asst. Prof. Faruk ULAMIŞ

Department of Electronics and Automation, Vocational School of Hacılar Hüseyin Aytemiz, Kırıkkale University, Kırıkkale, Türkiye

Asst. Prof. İsmail ATACAK

Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Asst. Prof. Mehmet GÜÇYETMEZ

Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Sivas University of Science and Technology, Sivas, Türkiye

Asst. Prof. Mustafa KARHAN

Department of Computer Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Türkiye

Asst. Prof. Mustafa TEKE

Department of Electrical and Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Türkiye

Asst. Prof. Mustafa Yasin ERTEN

Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Kırıkkale University, Kırıkkale, Türkiye

Asst. Prof. Mürsel Ozan İNCETAŞ

Department of Computer Technologies, Vocational School of Alanya Chamber of Commerce and Industry, Alanya Alaaddin Keykubat University, Antalya, Türkiye

Asst. Prof. Rukiye KARAKIŞ

Department of Software Engineering, Faculty of Technology, Cumhuriyet University, Sivas, Türkiye

Asst. Prof. Saadin OYUCU

Department of Computer Engineering, Faculty of Engineering, Adıyaman University, Adıyaman, Türkiye

Asst. Prof. Sevilay VURAL

Department of Internal Medicine Sciences, School of Medicine, Yozgat Bozok University, Yozgat, Türkiye

Asst. Prof. Zafer CİVELEK

Department of Electrical and Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Türkiye

Dr. Salih DEMİR

Department of Computer Engineering, Faculty of Open and Distance Education, Ankara University, Ankara, Türkiye

Nuri Alper METİN, PhD

Department of Electronics and Communications, Vocational School of Kırıkkale, Kırıkkale University, Kırıkkale, Türkiye

Ufuk TANYERİ, PhD

Department of Computer Technologies, Vocational School of Nallıhan, Ankara University, Ankara, Türkiye

Yunus KÖKVER, PhD

Department of Computer Technologies, Vocational School of Elmadağ, Ankara University, Ankara, Türkiye

Journal of

Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

ENGLISH LANGUAGE EDITOR

Assoc. Prof. Esra Güzel TANOĞLU

*Department of Molecular Biology and Genetics, Hamidiye Health Sciences Institute, University of Health Sciences,
İstanbul, Türkiye*

STATISTICS EDITOR

Prof. Turgut KÜLTÜR

Department of Physical Therapy and Rehabilitation, Faculty of Medicine, Kırıkkale University, Kırıkkale, Türkiye

LAYOUT EDITOR

Şerife KUTLU

Engineer, MediHealth Academy Publishing, Ankara, Türkiye

ORIGINAL ARTICLES

Explainable AI-driven IoT botnet detection for 6G networks.	46-55
<i>Güllü, M., Atagün, E., Biroğul, S., & Barışçı, N.</i>	
An integrated deep learning approach for 3D urban scene generation using monocular satellite imagery	56-63
<i>Karadağ, Ö. C., & Altuntaş, N.</i>	
An imitation-enhanced actor-critic approach for maze navigation	64-71
<i>Kalaycı, N., & Erbaş, M. D.</i>	
Android application for multilingual hate speech detection: using XLM-RoBERTa and ONNX optimized.....	72-76
<i>Doghmash, S. T., Doğru, İ. A., & Kılıç, K.</i>	
Explainable brain tumor classification using an attention-augmented CNN and Grad-CAM	77-82
<i>Mustafa, M. A., Erdem, O. A., & Söğüt, E.</i>	
Binary text classification using machine learning and transformer models for small and imbalanced datasets.....	83-91
<i>Şenyurt, İ., & Toklu, S.</i>	

Explainable AI-driven IoT botnet detection for 6G networks

Merve Güllü^{*1}, Ercan Atagün², Serdar Biroğul², Necaattin Barışçı³¹Department of R&D, Türk Telekom, Ankara, Türkiye²Department of Computer Engineering, Faculty of Engineering, Düzce University, Düzce, Türkiye³Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Cite this article as: Güllü, M., Atagün, E., Biroğul, S., & Barışçı, N. (2026). Explainable AI-driven IoT botnet detection for 6G networks. *J Comp Electr Electron Eng Sci*, 4(2), 46-55. <https://doi.org/10.51271/JCEES-0036>

Received: 07.11.2025

Accepted: 30.12.2025

Published: 13.10.2026

ABSTRACT

The rapid proliferation of internet of things (IoT) devices in smart homes, healthcare, and industrial automation introduces severe security challenges, especially with the transition toward 6G intelligent networks. Traditional machine learning models achieve high detection accuracy for IoT botnets but often lack interpretability, limiting their trustworthiness in mission-critical 6G environments. This study proposes an explainable AI-driven IoT botnet detection framework utilizing the N-BaIoT dataset to bridge this gap. The framework integrates correlation-based feature selection, mutual information analysis, ANOVA filtering, and low-variance elimination to construct a lightweight yet discriminative feature space. Several ensemble learning algorithms were evaluated, with light gradient boosting machine (LightGBM) achieving the highest accuracy (99.82%) and exceptional stability across cross-validation folds. To enhance transparency, model decisions were explained using SHapley Additive ExPlanations (SHAP) and local interpretable model-agnostic explanations (LIME), revealing key traffic features that differentiate benign and malicious flows. The proposed system demonstrates not only high detection precision but also human-interpretable reasoning, fulfilling essential trust and reliability requirements for next-generation 6G IoT security architectures.

Keywords: IoT botnet detection, explainable Artificial Intelligence, 6G networks, cybersecurity, feature selection, anomaly detection

INTRODUCTION

The proliferation of internet of things (IoT) devices across smart homes, healthcare systems, industrial automation, and urban infrastructure has increased rapidly (Navneet Kaur, 2025). This expansion has introduced unprecedented security vulnerabilities that threaten the integrity of emerging 6G networks (Emmanuel, 2025). With the rapid global growth of device deployments, the attack surface for botnet exploitation has expanded significantly. IoT botnets such as Mirai and Gafgyt have shown how dangerous compromised device networks can be. They are capable of launching large-scale distributed denial of service (DDoS) attacks that disrupt critical services and threaten user privacy (Son, 2024). Unlike traditional computing systems, IoT devices operate with very limited resources. Their restricted processing power, memory, and energy capacity make it difficult to apply conventional security mechanisms.

Furthermore, the heterogeneity of IoT manufacturers, firmware implementations, and communication protocols creates fragmented security landscapes that adversaries systematically exploit. Machine learning (ML) approaches have emerged as promising solutions for automated botnet detection, achieving high classification accuracy on benchmark datasets (Ali, 2024; Abbasi, 2025; Ullah, 2025). However, the black-box nature of these models poses fundamental challenges for

operational deployment in security-critical domains (Savant, 2024). Network administrators and incident response teams require transparent explanations of classification decisions to validate model behavior, diagnose false positives, and identify adversarial evasion attempts. The absence of interpretability undermines trust in automated systems and hinders regulatory compliance with emerging AI governance frameworks. This gap between detection accuracy and operational explainability motivates the integration of Explainable AI (XAI) techniques into IoT security architectures (Gummadi, 2024).

This paper addresses these challenges by proposing an explainable AI-driven framework for IoT botnet detection optimized for 6G network environments. This paper makes a meaningful contribution by combining high-accuracy classification with actionable explanations, which are essential for building trustworthy AI systems in next-generation network security.

Related Work

Detecting IoT botnets is critical for ensuring the security and stability of the IoT ecosystem. With billions of interconnected devices operating in diverse environments, botnet attacks can exploit network resources to launch DDoS assaults, steal data, and compromise device integrity. Therefore, the early

*Corresponding Author: Merve Güllü, gullu.merv@gmail.com



and accurate identification of malicious IoT traffic is a key requirement for maintaining network integrity and service continuity.

The literature demonstrates the successful application of ML and deep learning (DL) techniques for IoT botnet detection. Publicly available datasets such as N-BaIoT, BoT-IoT, IoT-23, and MedBioT have significantly accelerated research progress by enabling reproducible and comparable evaluations. Experiments on these datasets have shown that traditional classifiers (e.g., Support Vector Machine, k-NN, decision trees), ensemble methods, and deep learning architectures typically achieve detection accuracies exceeding 95-99%.

Beyond N-BaIoT, several other datasets, such as BoT-IoT, MedBioT, and IoT-23 have significantly advanced IoT botnet detection research. The BoT-IoT dataset (Nickolaos Koroniotis, 2019), which includes synthetically generated benign and malicious traffic (e.g., DDoS, scanning, and data exfiltration), provides a comprehensive benchmark for algorithm evaluation. Likewise, (Jadel Alsamiri, 2019) reported that the k-NN algorithm achieved up to 99% accuracy in detecting IoT botnet traffic. These results confirm that traditional supervised learning approaches can reach near-perfect detection rates on IoT malware datasets. However, interpretability and deployment efficiency remain key challenges for resource-constrained IoT devices.

Similarly, (Evanson Mwangi Karanja, 2020) converted N-BaIoT network traffic into grayscale images and extracted Haralick texture features to identify IoT malware. When combined with standard classifiers, this image-based approach achieved approximately 95% accuracy, demonstrating the effectiveness of visual feature extraction in representing malicious IoT traffic patterns. These findings highlight that combining deep learning architectures with creative feature extraction strategies can effectively model malicious behaviors in IoT environments.

More recently, the IoT-23 dataset (Sebastian Garcia, 2020) has provided labeled network captures from real IoT devices, encompassing 20 malicious and three benign scenarios. IoT-23 is frequently used alongside N-BaIoT and BoT-IoT for comparative analysis. Ensemble boosting algorithms such as XGBoost and LightGBM have been evaluated on these datasets, achieving over 99% accuracy in multi-class botnet detection. (Saied, 2023) compared tree-based ensemble techniques comprehensively on N-BaIoT and found that Random Forest achieved 99.999% multi-class accuracy. Empirical findings confirm that ensemble learners enhance detection rates and maintain computational efficiency, making them highly practical for real-world IoT security applications (Al-Othman, 2020) applied multiple ML algorithms to this dataset, after addressing class imbalance using Synthetic Minority Oversampling Technique (SMOTE), and found that ensemble decision tree models achieved the highest accuracy.

Trajanovski and Zhang (2021) proposed IoT-BDA, an integrated honeypot and sandbox framework that automates the capture, analysis, and reporting of IoT botnets and reveals advanced evasion techniques used in real attacks.

In one of the earliest studies following the release of the N-BaIoT dataset, Krishna and Thangavelu (2021) used traditional machine learning pipelines with metaheuristic

feature selection and a random forest classifier to detect IoT attacks.

Similarly, Abbasi et al. (2021) applied classic learning techniques such as logistic regression and artificial neural networks to the newly introduced N-BaIoT dataset to perform anomaly detection in IoT networks.

In this context, (Alkahtani, 2021) applied a hybrid CNN-LSTM model to the N-BaIoT dataset for classifying botnet traffic across different IoT device types (e.g., webcams, thermostats, doorbells). Separate models were trained for each device category, achieving 88-90% detection accuracy in distinguishing between benign and malicious traffic.

The MedBioT dataset (Alejandro Guerra-Manzanares, 2022) built from traffic collected from 83 IoT devices under real-world conditions, contains benign and malicious samples from well-known botnets such as Mirai, Bashlite, and Torii. Using this dataset, supervised learning models such as random forest have demonstrated strong performance, achieving 96.2% accuracy in differentiating botnet and benign traffic. The availability of datasets like MedBioT enables researchers to assess the generalizability of detection models across diverse IoT environments and malware families.

Santhadevi and Janet (2023) introduced an edge-based stacked deep learning system that processes NetFlow data and uses vectorized CNN and LSTM models with optimization methods to detect IoT malware in real time.

Rallapalli et al. (2023) presented a hybrid feature selection approach with a modified CNN that improves IIoT anomaly detection performance on the N-BaIoT dataset compared with existing models.

In recent years, beyond achieving high detection accuracy, the emphasis has shifted toward explainability. Considering the constraints of IoT environments, limited computing resources, energy efficiency, and real-time operational requirements, research has increasingly focused on developing detection systems that are accurate, transparent, interpretable, and deployable at the network edge.

Table 1 summarizes several representative studies that utilized N-BaIoT or similar IoT botnet datasets, highlighting their approaches and key results. While the primary focus in IoT botnet detection has been improving accuracy and efficiency, there is a growing interest in making these detection models more explainable. As complex AI models (e.g., deep neural networks and ensembles) are deployed for security, understanding the reasoning behind their alerts becomes crucial for operator trust and effective mitigation.

While Saied (2023) focused primarily on maximizing detection accuracy through tree-based ensemble techniques and Navneet Kaur (2025) employed XAI methods in 6G-IoT security contexts, the proposed framework distinguishes itself by simultaneously integrating explainability mechanisms with a lightweight LightGBM classifier optimized through multi-stage feature selection, thereby achieving both high detection accuracy and comprehensive interpretability suitable for resource-constrained edge deployments in 6G environments.

Table 1. Comparison of IoT botnet detection studies

Study (year)	Dataset (s)	Approach/model	XAI used?	Performance
Meidan, 2018	N-BaIoT	Deep autoencoder (per-device anomaly detector)	No	100% TPR (binary detection)
Alkahtani, 2021	N-BaIoT	CNN-LSTM hybrid deep neural network	No	~90% accuracy (binary per device)
Evanson Mwangi Karanja, 2020	N-BaIoT	Traffic-to-image+texture feature ML classification	No	80-95% accuracy (varies by class)
Al-Othman, 2020	BoT-IoT	Various ML classifiers; ensemble (J48, RF) best	No	96% accuracy (multi-class)
Jadel Alsamiri, 2019	BoT-IoT	Ensemble of decision trees; feature augmentation	No	99% accuracy (binary)
(Alejandro Guerra-Manzanares, 2022)	MedBioT	Supervised ML (RF, etc.) on new IoT botnet data	No	96.2% accuracy (binary)
(Saied, 2023)	N-BaIoT	Bagging/boosting ensemble trees (RF, XGBoost)	No	99.99% accuracy (multi-class)
(Do, 2024)	N-BaIoT	CNN, LSTM, GRU	No	89% accuracy (multi-class)
(Nazir, 2024)	N-BaIoT	CNN-LSTM hybrid model	No	99.99% accuracy (multi-class)
(Serhane, 2025)	N-BaIoT	Extra Trees Using PCA	No	99.94% accuracy (multi-class)

N-BaIoT, BoT-IoT, and MedBioT are dataset names. IoT: Internet of things, XAI: Explainable Artificial Intelligence, TPR: True positive rate, CNN: Convolutional neural network, LSTM: Long short-term memory, ML: Machine learning, J48: Java decision tree algorithm (C4.5), RF: Random forest, N-BaIoT: Network-based detection of IoT botnet attacks, BoT-IoT: Botnet Internet of Things, MedBioT: Medium-sized Botnet internet of things, XGBoost: Extreme gradient boosting, GRU: Gated recurrent unit, PCA: Principal component analysis

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Methodology

In this section, the process steps followed in the problem and the models used are explained and the architecture of the process is shown in Figure 1.

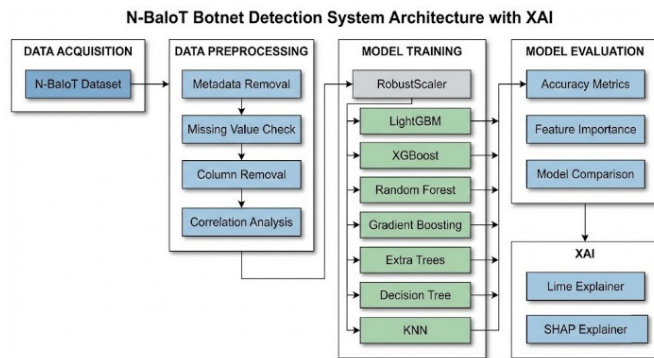


Figure 1. The system architecture for N-BaIoT and XAI

N-BaIoT: Network-based detection of IoT botnet attacks, XAI: Explainable Artificial Intelligence, LightGBM: Light gradient boosting machine, XGBoost: Extreme gradient boosting, KNN: K-nearest neighbors, LIME: Local interpretable model-agnostic explanations, SHAP: SHapley additive exPlanations

N-BaIoT Dataset

The N-BaIoT (Meidan, 2018) was introduced to address the lack of public IoT botnet datasets by providing real network traffic from nine commercial IoT devices infected with Mirai and Bashlite malware. The dataset includes benign traffic and 10 attack types (5 Bashlite and 5 Mirai) and can be used for both anomaly detection (benign vs. malicious) and multi-class classification tasks.

Feature Engineering Pipeline

Correlation analysis: Pearson correlation coefficient was computed to identify and remove highly correlated features, reducing multicollinearity. For feature pairs X_i , X_j the correlation coefficient is:

$$\rho_{\{X_i, X_j\}} = \frac{\text{cov}(X_i, X_j)}{\sigma_{\{X_i\}} \sigma_{\{X_j\}}} = \frac{E[(X_i - \mu_{\{X_i\}})(X_j - \mu_{\{X_j\}})]}{\sigma_{\{X_i\}} \sigma_{\{X_j\}}} \quad (1)$$

Where μ and σ denote mean and standard deviation, respectively. Features with $|\rho| > 0.95$ were considered redundant, and one feature from each highly correlated pair was removed (Max Kuhn, 2013).

Mutual information (MI): MI quantifies the mutual dependence between features and target labels, capturing both linear and non-linear relationships.

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (2)$$

Where $p(x, y)$ is the joint probability distribution, and $p(x)$, $p(y)$ are marginal distributions. Features with MI scores below a threshold were eliminated as they provide minimal information gain for classification.

ANOVA F-test: Analysis of Variance F-test assesses the statistical significance of feature means across different classes (Montgomery, 2017). The F-statistic is calculated as:

$$F = \frac{MS_{\text{between}}}{MS_{\text{within}}} = \frac{\sum_{i=1}^k n_i (\bar{X}_i - \bar{X})^2 / (N - k)}{\sum_{i=1}^k \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 / (k - 1)} \quad (3)$$

Where k is the number of classes, n_i is the sample size of class i and N is the total sample size. Features with higher F-scores demonstrate greater discriminative power between classes.

Low variance elimination: Features with variance below a specified threshold were removed to eliminate near-constant features that contribute minimal discriminative information (Guyon, 2003). The variance of feature X is:

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (4)$$

This technique reduces dimensionality while preserving informative features.

Classification Models

Extra trees classifier: Extremely Randomized Trees (Extra Trees) is an ensemble method that constructs multiple decision trees with random node splits (Pierre Geurts, 2006). Unlike Random Forests, Extra Trees randomly selects split thresholds without searching for optimal values, reducing computational complexity:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(x) \quad (5)$$

Where T is the number of trees, h_t is the prediction of tree t , and x is the input feature vector.

XGBoost: XGBoost (eXtreme Gradient Boosting) implements gradient boosting with regularization to prevent overfitting (Tianqi Chen, 2016). The objective function minimizes:

$$L(\phi) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{k=1}^K \Omega(f_k) \quad (6)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (7)$$

Where l is the loss function, $\Omega(f_k)$ is the regularization term, T is the number of leaves, and w represents leaf weights. The model builds trees sequentially, with each tree correcting errors from previous iterations.

Random forest: An ensemble learning technique called Random forest uses bootstrap aggregating, also known as bagging, to build several decision trees (Breiman, 2001). A random subset of characteristics and data samples are used to train each tree. The final prediction aggregates individual tree predictions:

$$\hat{y} = \text{mode} \{h_1(x), h_2(x), \dots, h_T(x)\} \quad (8)$$

For classification tasks, where mode represents the majority vote.

LightGBM: LightGBM is a decision tree-based gradient boosting framework that has focused on speed and scalability since its development (Wang, 2017). It uses leaf-wise tree growth with depth constraints and histogram-based splitting. The objective function is minimized as:

$$L(\phi) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{k=1}^K \Omega(f_k) \quad (9)$$

Where l is the loss function, $\Omega(f_k)$ is a regularization term, and each tree f_k aims to correct the residuals of the previous iteration.

Decision tree: A decision tree recursively partitions the feature space into regions that minimize impurity, using measures like Gini index or information gain. The predicted output is the mean (regression) or mode (classification) of samples within each leaf:

$$\hat{y} = \text{mode} (y_i \in R_j) \quad (10)$$

where R_j denotes the j^{th} region (leaf) of the tree.

K-Nearest Neighbors (KNN)

The KNN method determines the class of a data point based on the label most commonly held by its k nearest neighbors (Steinbach, 2009). The distances between these neighbors are typically calculated using the Euclidean distance, which corresponds to the straight-line distance between two points. The distance metric is typically Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^m (x_j - x_{ij})^2} \quad (11)$$

The predicted class is determined by:

$$\hat{y} = \text{mode} [y_i: x_i \in N_k(x)] \quad (12)$$

Where $N_k(x)$ denotes the set of k nearest training samples to x .

Explainable AI Techniques

Tree-based feature importance: Tree-based models inherently provide feature importance scores by measuring the total reduction in impurity (e.g., Gini impurity or entropy) contributed by each feature across all trees (Breiman, 2001):

$$\text{Importance}(X_j) = \frac{1}{T} \sum_{t=1}^T \sum_{n \in N_t} \Delta i(n, X_j) \quad (13)$$

Where N_t is the set of nodes in tree t that split on feature X_j , and Δi is the impurity reduction at node n .

SHAP Values

SHAP assigns each feature a contribution value based on game theory (Lundberg, 2017).

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \quad (14)$$

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} K$$

$$K = \left[f_{\{S \cup \{i\}\}}(x_{\{S \cup \{i\}\}}) - f_S(x_S) \right] \quad (15)$$

$$\left[f_{\{S \cup \{i\}\}}(x_{\{S \cup \{i\}\}}) - f_S(x_S) \right]$$

Where F is the set of all features, S is a subset excluding feature i , and f_S represents the model prediction with only features in S . SHAP values satisfy properties of local accuracy, missingness, and consistency.

LIME: LIME explains individual predictions by approximating the model locally with an interpretable linear model (Ribeiro, 2016). For instance x , LIME minimizes:

$$\xi(x) = \text{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (16)$$

Where g is an interpretable model, L measures the approximation fidelity, π_x defines the local neighborhood, and $\Omega(g)$ penalizes model complexity. LIME generates perturbations around x and fits a sparse linear model to explain local behavior.

Experimental Results

Data preprocessing: The preprocessing process of the N-BaIoT dataset consists of a series of systematic steps aimed at ensuring data quality in botnet detection and optimizing

the representation of attributes. The raw dataset comprises 7,062,819 records with 123 distinct features, collected from 92 separate CSV files reflecting the behavior of various IoT devices in both benign and attack scenarios. During the initial data cleaning, 213 metadata entries were removed. As a result of this process, 7,062,606 valid data records remained. These records were divided into a total of 11 classes: five were Mirai attack types (ack, scan, syn, udp, udpplain); five were Gafgyt attack types (combo, junk, scan, tcp, udp); and the last one was normal (benign) network traffic.

Six metadata columns (file name, data count, device ID, device name, feature name, feature description) were removed because they were completely empty. These columns did not contain any useful information for classification. Next, a feature whose value never changed across all data was identified and removed from the list. This reduced the total number of features from 117 to 116. Features with fixed values do not contribute to machine learning models, creating unnecessary overhead and remaining useless. Therefore, eliminating them was important for both speed and efficiency.

The most significant reduction in the dataset occurred through the removal of duplicate records. This process resulted in the elimination of 4,579,930 redundant records (64.8% of the dataset) from the system. This left 2,482,676 unique data records. This intensive deduplication step eliminates repetitive traffic patterns, a common problem in IoT datasets. It also aims to preserve the fundamental behavioral signatures necessary for accurate classification.

Feature Selection Outcomes

During the feature selection phase, a correlation-based redundancy removal method was used to address the issue of multicollinearity and reduce the risk of overfitting in the model. Pearson correlation analysis revealed 93 features that were highly correlated with each other (correlation coefficient >0.95). This corresponded to 80.9% of the original feature set. The vast majority of these redundant features came from the MI direction statistics group (e.g., MI_dir_L3_weight, MI_dir_L3_mean, MI_dir_L3_variance, MI_dir_L1_weight). Their nearly perfect linear relationship with each other indicated that they carried overlapping information about network traffic characteristics.

The number of features was reduced from 115 to 22 (by 80.9%). This radical step was a deliberate choice. This resulted in four key gains: first, in a task where it was difficult to distinguish between seven different types of attacks, the “curse of dimensionality” effect was minimized. Second, processing speed increased approximately fivefold. Third, the comprehensibility of the models was improved by focusing only on the most effective traffic features. Finally, the risk of overfitting was reduced by removing unnecessary data that created false relationships. The remaining 22 features constitute the cleanest dataset needed to capture the fundamental patterns that distinguish normal traffic from botnet attacks. This number retains sufficient information to accurately identify all seven categories.

Classification Results

In tests conducted on the N-BaIoT dataset, seven different machine learning algorithms were compared to detect botnet attacks. As shown in [Table 2](#), ensemble learning methods,

particularly gradient boosting algorithms, were more successful than others in this problem. These algorithms both distinguished attack types more clearly and produced outputs suitable for practical use by keeping the processing load at reasonable levels.

Table 2. Models performance

Model	Test accuracy	CV mean	CV SD
LightGBM	0.998153	0.999146	0.00058
XGBoost	0.996991	0.996998	0.000071
Random Forest	0.996874	0.996889	0.000092
Gradient Boosting	0.996866	0.996752	0.000274
Decision Tree	0.994834	0.921168	0.044626
Extra Trees	0.994544	0.993838	0.000763
KNN	0.832777	0.830706	0.000566

LightGBM: Light gradient boosting machine, XGBoost: Extreme gradient boosting, KNN: K-nearest neighbors, CV: Cross-validation, SD: Standard deviation

The LightGBM algorithm demonstrated the best classification performance. This algorithm achieved high success rates of 99.81% accuracy on the test data and 99.91% on the cross-validation average. LightGBM was followed by XGBoost 99.70%, Random Forest 99.69% and Gradient Boosting 99.68%. It is particularly noteworthy that all four algorithms achieved accuracy rates above 99%. Furthermore, the cross-validation standard deviations, ranging from 0.00058 to 0.000274, were quite low. This low deviation clearly indicates that the models have high generalization ability and are not affected by overfitting.

On the other hand, although the decision tree algorithm achieved a relatively high accuracy rate of 99.48% on the test data, the cross-validation average dropped to 92.12%. Moreover, the standard deviation is also quite high at 0.044626. These results indicate that the model is weak in terms of stability and generalization ability. Such behavior indicates that a single decision tree is sensitive to noise and variance in the dataset, while confirming that ensemble methods successfully overcome this problem. The KNN algorithm, on the other hand, showed the lowest performance with a test accuracy of 83.27%. This low performance supports the idea that distance-based algorithms have limited effectiveness on high-dimensional and large-scale datasets.

XAI Insights

Model interpretability is a critical requirement in cybersecurity applications. In this section, feature importance analysis was conducted to understand the decision-making mechanisms of the AI model and to identify which network features are the most decisive in detecting botnet attacks.

Feature importance consensus: As shown in [Figure 1](#) the top 20 network features identified by the LightGBM algorithm are presented. The feature importances were computed based on each feature’s cumulative contribution to class discrimination during model training, using the split-based importance criterion.

The feature importance analysis of the LightGBM model reveals the most distinctive network traffic attributes for botnet attack detection. As illustrated in [Figure 2](#), the visualization and output data indicate that MI based features contribute the most to the classification performance.

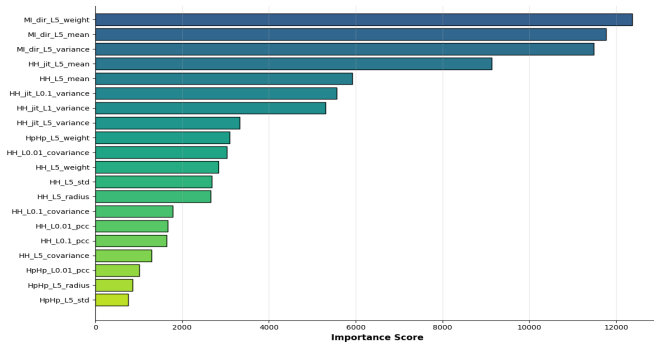


Figure 2. LIME-based feature importance for the LightGBM classifier

MI: MAC-IP, dir: Direction, HH: Host-to-host, jit: Jitter, HpHp: Host-port to host-port, L5: 1-minute time window, L0.1: 500-millisecond time window, L0.01: 100-millisecond time window, pcc: Pearson correlation coefficient, LIME: Local interpretable model-agnostic explanations, LightGBM: Light gradient boosting machine

In the analyses of the N-BaIoT dataset, the MI_dir_L5_weight feature showed the highest impact (importance score: 12.370). It was followed by MI_dir_L5_mean (11.772) and MI_dir_L5_variance (11.490). What is noteworthy is that all three features belong to the MI category and examine the last 5 packets (L5). This highlights how decisive short-term network traffic behavior and inter-packet information flow are in detecting botnet attacks. MI measurements successfully capture botnets' typical command-and-control mechanisms and coordinated attack patterns because they monitor communication between source and destination IP addresses. In other words, these features reflect the digital traces that show how attackers manage their "bot armies".

The second group of important features consists of jitter and statistical descriptors of the host-host (HH) and host-port-host-port (Hp-Hp) types. For example, metrics such as HH_jit_L5_mean (9.129) and HH_L5_mean (5.926) highlight how important indicators fluctuations in packet arrival times (jitter) are in characterizing botnet traffic. The temporal behavioral differences between normal user traffic and botnet traffic have been successfully captured, particularly through second-order statistical moments such as variance, standard deviation, and covariance. The multiple window sizes used (L0.01, L0.1, L1, L5) enabled the model to learn both micro- and macro-scale temporal patterns in network activity. The uneven distribution of feature importance scores, and particularly the clear dominance of the first three features, suggests that these metrics should be prioritized in the dimension reduction and feature selection processes. In conclusion, the high classification performance achieved by the LightGBM model can be attributed to the effective use of Mutual Information and temporal variation metrics. Together, these metrics enabled the model to capture the intricate dynamics of botnet communication behavior.

The LIME analysis in Figure 3 makes the LightGBM model's decision on a single example understandable. Specifically, in black-box models, green-colored features support the prediction (increasing the prediction probability), while red ones work against the decision. For example, the HpHp_L5_weight > 0.00 feature alone increases the prediction probability by 30%. This shows that communication patterns at the host machine port level function as a "digital fingerprint" for botnet attacks. Conversely, the HH_jit_L5_mean feature being nearly zero (-0.12 effect) provides an important clue: Regular time intervals between packets indicate normal network

behavior, not botnet traffic. Other red features similarly show how staying below or above certain threshold values affects the model. Another noteworthy point is that features highlighted by LIME, such as MI_dir_L0.01_weight and MI_dir_L5_mean, confirm the list of "most critical features" from previous analyses. In other words, the model not only makes accurate predictions but also provides intelligent justifications for each decision. This transparency is crucial in cybersecurity as it enhances reliability and auditability.

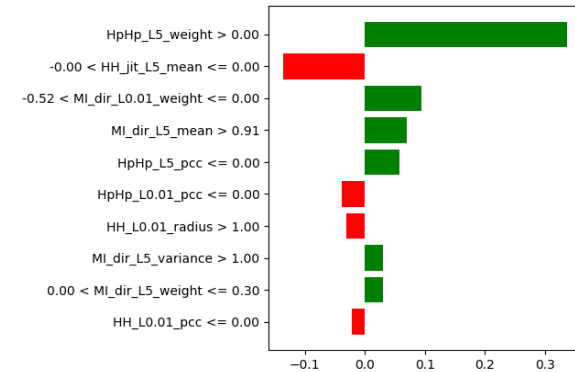


Figure 3. LIME-based local explanation of feature contributions for a representative Mirai test instance (ID: 121958). The visualization highlights both positive (green) and negative (red) influences on the LightGBM classifier's decision

MI: MAC-IP, dir: Direction, HH: Host-to-host, jit: Jitter, HpHp: Host-port to host-port, L5: Lambda (λ)=5 decay window, L0.01: Lambda (λ)=0.01 decay window, pcc: Pearson correlation coefficient, LIME: Local interpretable model-agnostic explanations, LightGBM: Light gradient boosting machine, ID: Identifier

Overall, the local interpretability provided by the LIME analysis shows that the model not only achieves high accuracy but also offers a logical and traceable rationale for each individual decision an aspect of great significance for reliability and auditability in cybersecurity applications.

SHAP analysis: SHAP values were used to examine how the LightGBM-based botnet detection model explains its decisions. This method is based on a solid mathematical foundation derived from cooperative game theory and clearly shows which features are most influential in the model. SHAP stands out as an ideal tool for making model decisions transparent and reliable in critical areas such as cybersecurity.

The SHAP analysis in Figure 4 shows the ten most influential features used by the LightGBM model when predicting the "benign" traffic class and the average effect sizes of these features. Unlike LIME, which provides local-level interpretation, SHAP analysis allows us to understand the model's behavior on a global scale. It does this by reporting the mean absolute SHAP values across the entire dataset.

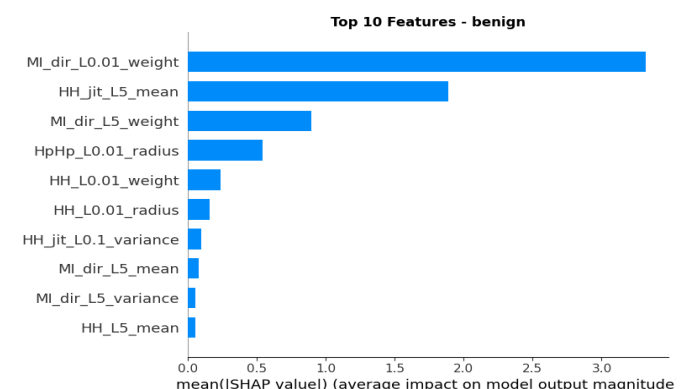


Figure 4. SHAP-based feature importance ranking for benign traffic classification

MI: MAC-IP, dir: Directional, HH: Host-to-host, jit: Jitter, HpHp: Host-port to host-port, L5: λ =5 decay factor, L0.1: λ =0.1 decay factor, L0.01: λ =0.01 decay factor, SHAP: SHapley additive exPlanations

The visualization shows that `MI_dir_L0.01_weight` is the most critical discriminative feature in detecting benign traffic, with a SHAP value of approximately 3.0. This metric represents the weighted average of directional mutual information within a very short time window (L0.01) and characterizes the regular and predictable communication patterns of normal user traffic.

The second most effective feature, `HH_jit_L5_mean` (average SHAP value approximately 2.0), shows that the average host-level delay (jitter) duration is a very important indicator in determining normal traffic. The relatively consistent arrival times and low delay values of packets in normal network traffic explain why this feature has such strong separation power. Although `MI_dir_L5_weight` (average SHAP value approximately 1.0) and other Mutual Information-based metrics have lower SHAP scores, this indicates that while these features are important across all classes, they are less decisive in specifically characterizing the benign class. Features with moderate SHAP values, such as `HpHp_L0.01_radius`, `HH_L0.01_weight`, and `HH_L0.01_radius`, indicate that spatial and statistical metrics at the host-port level play a supporting role in identifying benign traffic. SHAP's global perspective systematically reveals which features the model relies on for each class. This provides invaluable insights for feature engineering and model optimization. Furthermore, the mathematical consistency and aggregability of SHAP values ensure that the model's decisions can be explained in a transparent and auditable manner. This is a fundamental requirement in cybersecurity applications.

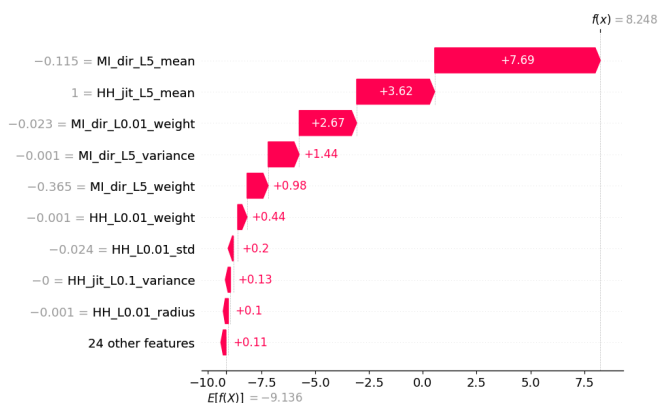


Figure 5. SHAP-based waterfall plot illustrating feature-level contributions for a representative prediction

MI: MAC-IP, dir: Directional, HH: Host-to-host, jit: Jitter, L5: 1-minute time window, L0.1: 500-millisecond time window, L0.01: 100-millisecond time window, SHAP: SHapley additive exPlanations

The SHAP waterfall diagram in **Figure 5** shows step-by-step how the LightGBM model makes a decision for a single example. Starting from the baseline value ($E[f(x)] = -9.136$), the diagram reaches the final result ($f(x) = 8.248$) by summing the positive/negative effects of each feature.

In **Figure 5**, red-colored features support the prediction. `MI_dir_L5_mean` has the largest effect (+7.69). This feature strongly influences the model even though its actual value on the example (-0.115) is low. `HH_jit_L5_mean` (value=1, +3.62) ranks second. Inter-packet timing deviation (jitter) plays a critical role in identifying attack traffic. Other MI-based features (`MI_dir_L0.01_weight`, `MI_dir_L5_variance`, etc.) also confirm the global analysis results with positive contributions. Host-level statistics (`HH_L0.01_weight`, `HH_jit_L0.1_variance`, etc.) have small but significant effects, while

the total contribution of the remaining twenty four features is only +0.11. This reveals that the model bases its decision on the top ten features.

6G Integration and Applications

6G wireless networks, expected to be commercialized around 2030 with standardization efforts beginning in 2025, are poised to bring significant change to the IoT ecosystem. 6G is designed as “Intelligent Networks of Everything.” These networks will offer high speed and capacity, utilizing integrated Artificial Intelligence/machine learning (AI/ML) from core to edge. Thus, they will serve as a universal IoE, offering advanced functions in applications such as computing, sensing, positioning, and energy management (Khan Waqar, 2025). While this major transformation creates unique opportunities for IoT environments, it also brings serious security challenges. Therefore, the attack detection methods developed in this study are particularly important for this future ecosystem.

Our enhanced LightGBM-based botnet detection system can serve as a digital proxy in the dense IoT ecosystems of 6G networks. The terahertz frequencies offered by 6G, AI-driven management, and the simultaneous connection of billions of devices necessitate a shift in our security approaches.

Our model is ideal for real-time attack detection in these environments, with 99.82% accuracy and high stability (cross-validation standard deviation: 0.00058). SHAP and LIME analyses show how traffic metrics such as `MI_dir_L5_weight` and `HH_jit_L5_mean` distinguish botnets like Mirai-Gafgyt or DDoS-scanning attacks. Thanks to 6G's edge computing architecture, this model can be distributed to every corner of the network. Thus, dependency on central servers is eliminated, response time reduction is targeted, by working in conjunction with network slicing and software-defined networking (SDN), security rules specific to each service layer (eMBB, URLLC, mMTC) can be automatically applied. This allows the logic behind every “attack” alert to be traced. 6G operators can monitor decisions and trust the system. Because cyber defense now relies on evidence-based transparency, not speculation.

DISCUSSION

Security Imperatives in 6G-IoT Ecosystems

Protecting 6G IoT networks from threats, especially next-generation attacks, is one of the most significant challenges this infrastructure must overcome. This situation has necessitated the development of innovative architectures and paradigms enhanced by intelligence, software-defined approaches, and infrastructure virtualization (Asma Alotaibi, 2023). The N-BaIoT dataset used in this research includes real-world botnet attacks such as Mirai and Bashlite on commercial IoT devices. This reveals critical information about attack patterns that will persist and evolve in 6G-enabled environments. As Wireless Sensor Networks scale up within 6G frameworks, their exposure to cyber threats will rise sharply due to diverse communication protocols, dynamic node topologies, AI-enabled breaches, adversarial attacks targeting AI models, and resource-constrained device architectures.

The attack detection model developed in this study provides a foundation for the fundamental security requirements of future 6G networks.

In the IoT world of 6G, which is extremely dense and diverse, traditional security methods will no longer suffice. In this environment, active and AI-powered defense systems have become essential. The LightGBM-based detection system we present in our study has the capacity to meet the layered security needs of future 6G networks. This is because 6G networks will carry data at terabit speeds and are expected to have low response times. This will lead to botnet attacks becoming both more complex and faster. Existing systems will not be able to respond at this speed; defenses will have to be as fast and intelligent as the attacks.

Feature importance analysis results indicate that temporal mutual information and jitter metrics must be monitored in real time. In particular, metrics such as MI_dir_L5_weight (importance score: 12.370) and HH_jit_L5_mean (importance score: 9.129) have become a security requirement in this context. And also, as seen with SHAP analysis, the MI_dir_L0.01_weight feature (SHAP value approx. 3.0) plays a dominant role in detecting normal traffic. This role forms the basis for accurately characterizing the normal behavior profiles of 6G-IoT devices and enabling sensitive anomaly detection. Furthermore, eliminating highly correlated features (correlation > 85%) using the correlation filtering approach ensures that the model can be efficiently deployed on edge computing infrastructures.

Future Perspectives: 6G Security and XAI Integration

The critical role of explainable AI in 6G security: The integration of XAI techniques in this study represents a fundamental requirement for 6G network security rather than merely an enhancement. As 6G manages a wide range of mission-critical services, increased explainability to enable trust is critical, particularly as the migration from traditional mathematical model-dependent optimization to data-dependent deep learning decreases the insight and trust in optimization modules. The loss of model explainability creates vulnerabilities to adversarial attacks and regulatory compliance challenges.

Using XAI techniques in this study is one of the new research topics in 6G security. 6G will manage critical systems such as healthcare, energy, or transportation. The transition from traditional mathematical models to deep learning-based systems turns decision-making processes into a black-box. This creates security vulnerabilities and complicates legal oversight. For example, techniques such as LIME and SHAP were used for similar purposes in a 6G-IoT security study published in 2025 (Navneet Kaur, 2025). In our N-BaIoT-based analysis, XAI also allows experts to see “how each attack was detected.” This capability is critical, especially for rapidly changing threats such as Mirai or DDoS. The biggest problem with black-box models was that they broke down the trust barrier between humans and machines. For it to be reliable, the logic behind every “attack” alert must be traceable. This is where our model comes into play: SHAP and LIME clarify the reasoning behind each prediction. This allows security teams to use automated systems not by questioning them, but by understanding them. This ensures control in high-risk 6G networks and allows humans to trust machines.

Application scenarios in 6G-enabled smart cities: The attack detection capabilities demonstrated in this study are applicable in emerging 6G use cases. 6G technology, combined with Wireless Sensor Networks, is poised to fundamentally

transform smart city infrastructure (Khan Waqar, 2025). With ultra-fast communication, efficient real-time data transfers, and high reliability, 6G will be the preferred choice, particularly in areas such as smart transportation systems, environmental monitoring, and public safety. The botnet attacks analyzed in this study using the N-BaIoT dataset (particularly those targeting security cameras, thermostats, and other IoT devices) represent potential threat vectors that could spread to 6G-enabled smart city infrastructures.

Addressing 6G-specific challenges: The ability to detect zero-day attacks is a truly important area of research for 6G networks. Signature-based systems cannot detect these attacks; due to the complexity of 6G, experts prefer anomaly-based solutions (Iftikhar Rasheed, 2025). Our model learns “normal traffic” from N-BaIoT data and can detect deviations. This lays the foundation for detecting new types of attacks that will emerge as 6G technology matures. The distributed and heterogeneous nature of 6G requires security solutions that are both flexible and scalable. As proven in tests, our model works seamlessly even on IoT devices with limited resources. This makes it fully compatible with devices in 6G edge networks. Moreover, thanks to XAI integration, security teams in different regions can understand threats in real time and coordinate their response. This approach, which learns threat behavior rather than their source, shapes security in future smart networks through human-machine collaboration.

Future Works

The integration of the methodologies developed in this research with next-generation 6G technologies presents several promising avenues for future investigation. Adapting the proposed models trained on the N-BaIoT dataset to accommodate the substantially increased data throughput and ultra-low latency characteristics of 6G networks will necessitate the optimization of both detection algorithms and XAI interpretation mechanisms. Future research will focus on developing lightweight inference pipelines capable of maintaining detection accuracy while meeting the stringent real-time processing demands of 6G architectures.

Future studies will investigate the generalization capability of the proposed framework across diverse IoT botnet datasets, including BoT-IoT, IoT-23, and MedBIoT, to validate its robustness against heterogeneous attack patterns and network configurations.

Exploring federated learning approaches for distributed model training across edge devices without centralizing sensitive network traffic data constitutes another significant research direction. This approach would address privacy concerns while enabling collaborative model improvement across geographically distributed 6G IoT deployments.

Assessing the resilience of the proposed detection framework against adversarial evasion attacks and developing countermeasures to enhance model robustness in adversarial environments will be addressed in subsequent research.

CONCLUSION

The integration of the methods developed in this research with next-generation 6G technologies opens up many promising avenues for further research. First, adapting our models trained on N-BaIoT to manage the increased data speeds and reduced latency of 6G networks will require the

optimization of both detection algorithms and Explainable Artificial Intelligence (XAI) interpretation mechanisms. Second, integrating our XAI-enhanced detection capabilities with blockchain-based security frameworks could enhance data integrity and trust in 6G smart city applications.

The experimental results obtained using the N-BaIoT dataset in this study, when combined with the latest XAI techniques, form a foundation for ensuring the security of next-generation wireless networks. As 6G networks transition from theoretical designs to real-world deployment, rigorously tested attack detection and analysis methods will become indispensable. These techniques are essential for preserving security, safety, and reliability in network environments that are rapidly growing more complex and autonomous.

Emerging AI governance frameworks and regulatory standards increasingly mandate algorithmic transparency and explainability for automated security systems. The integration of SHAP and LIME methodologies in this study ensures compliance with these evolving requirements, facilitating the adoption of AI-driven security solutions in regulated industries such as healthcare, critical infrastructure, and financial services.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: MG, EA, SB, NB; Design: MG, EA, SB, NB; Control: MG, EA, SB, NB; Resources: MG, EA, SB, NB; Materials: MG, EA, SB, NB; Data Collection and/or Processing: MG, EA, SB, NB; Analysis and/or Interpretation: MG, EA, SB, NB; Literature Review: MG, EA, SB, NB; Writing the Article: MG, EA, SB, NB; Critical Review: MG, EA, SB, NB.

REFERENCES

- Abbasi, F., Naderan, M., & Alavi, S. E. (2021, May). Anomaly detection in internet of things using feature selection and classification based on logistic regression and artificial neural network on N-BaIoT dataset. In 2021 5th International Conference on Internet of Things and Applications (IoT) (pp. 1-7). IEEE. <https://doi.org/10.1109/IoT52625.2021.9469605>
- Abbasi, F., Naderan, M., & Alavi, S. E. (2025). Dimensionality reduction with deep learning classification for botnet detection in the internet of things. *Expert Systems with Applications*, 267, 126149. <https://doi.org/10.1016/j.eswa.2024.126149>
- Alani, M. M., & Damiani, E. (2023). XRecon: an explainable IoT reconnaissance attack detection system based on ensemble learning. *Sensors*, 23(11), 5298. <https://doi.org/10.3390/s23115298>
- Ali, S., Ghazal, R., Qadeer, N., Saidani, O., Alhayan, F., Masood, A., ..., & Gupta, D. (2024). A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks. *Alexandria Engineering Journal*, 103, 88-97. <https://doi.org/10.1016/j.aej.2024.05.113>
- Alkahtani, H., & Aldhyani, T. H. H. (2021). Botnet attack detection by using CNN-LSTM model for internet of things applications. *Security and Communication Networks*, 2021, 3806459. <https://doi.org/10.1155/2021/3806459>
- Alotaibi, A., & Barnawi, A. (2023). Securing massive IoT in 6G: Recent solutions, architectures, future directions. *Internet of Things*, 22, 100715. <https://doi.org/10.1016/j.iot.2023.100715>
- Al-Othman, Z., Alkasasbeh, M., & Al-Haj Baddar, S. W. (2020). An efficient approach to detect IoT botnet attacks using machine learning. *Journal of High Speed Networks*, 26(3), 241-254. <https://doi.org/10.3233/JHS-200641>
- Alsamiri, J., & Alsubhi, K. (2019). Internet of things cyber attacks detection using machine learning. *International Journal of Advanced Computer Science and Applications*, 10(12), 627-634. <https://doi.org/10.14569/IJACSA.2019.0101280>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. <https://doi.org/10.1023/A:1010933404324>
- Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939785>
- Do, P. H., Le, D. T., Vishnevsky, V. M., Berezkin, A., & Kirichek, R. (2024). A horizontal federated learning approach to IoT malware traffic detection: an empirical evaluation with N-BaIoT dataset. In 2024 26th International Conference on Advanced Communications Technology (ICACT) (pp. 1494-1506). IEEE. <https://doi.org/10.23919/ICACT60172.2024.10471929>
- Emmanuel, E. J. (2025). Systematic review of 6G-IoT privacy risks, emerging threats, mitigation strategies, and cybersecurity. *Asian Journal of Advanced Research and Reports*, 19(9), 180-190. <https://doi.org/10.9734/ajarr/2025/v19i91151>
- Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). IoT-23: a labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.4743746>
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63, 3-42. <https://doi.org/10.1007/s10994-006-6226-1>
- Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., & Nömm, S. (2022). Using MedBIoT dataset to build effective machine learning-based IoT botnet detection systems. In S. Furnell, P. Mori, E. Weippl, & O. Camp (Eds.), Information systems security and privacy: 6th International Conference, ICISSP 2020, revised selected papers (pp. 222-243). Springer. https://doi.org/10.1007/978-3-030-94900-6_11
- Gummadi, A. N., Napier, J. C., & Abdallah, M. (2024). XAI-IoT: an explainable AI framework for enhancing anomaly detection in IoT systems. *IEEE Access*, 12, 71024-71054. <https://doi.org/10.1109/ACCESS.2024.3402446>
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.
- Karanja, E. M., Masupe, S., & Jeffrey, M. G. (2020). Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things*, 9, 100153. <https://doi.org/10.1016/j.iot.2019.100153>
- Kaur, N., & Gupta, L. (2025). Securing the 6G-IoT environment: a framework for enhancing transparency in Artificial Intelligence decision-making through explainable Artificial Intelligence. *Sensors*, 25(3), 854. <https://doi.org/10.3390/s25030854>
- Khan, W., Usama, M., Khan, M. S., Saidani, O., Al Hamadi, H., Alnazzawi, N., ..., & Ahmad, J. (2025). Enhancing security in 6G-enabled wireless sensor networks for smart cities: a multi-deep learning intrusion detection approach. *Frontiers in Sustainable Cities*, 7, 1580006. <https://doi.org/10.3389/frsc.2025.1580006>
- Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-IoT dataset. *Future Generation Computer Systems*, 100, 779-796. <https://doi.org/10.1016/j.future.2019.05.041>
- Krishna, E. S. P., & Thangavelu, A. (2021). Attack detection in IoT devices using hybrid metaheuristic lion optimization algorithm and firefly optimization algorithm. *International Journal of System Assurance Engineering and Management*, 1-14. <https://doi.org/10.1007/s13198-021-01150-7>
- Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer. <https://doi.org/10.1007/978-1-4614-6849-3>
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765-4774.

- Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., ..., & Elovici, Y. (2018). N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3), 12-22. <https://doi.org/10.1109/MPRV.2018.03367731>
- Montgomery, D. C. (2017). Design and analysis of experiments (9th ed.). John Wiley & Sons.
- Nazir, A., He, J., Zhu, N., Qureshi, S. S., Qureshi, S. U., Ullah, F., ..., & Pathan, M. S. (2024). A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. *Ain Shams Engineering Journal*, 15(7), 102777. <https://doi.org/10.1016/j.asej.2024.102777>
- Rallapalli, S., Tiwari, V. K., Sagar, S. D. V., & Pareek, P. K. (2023). Detecting the anomaly attacks in industrial IoT using hybrid feature selection based modified convolutional neural network. In 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT) (pp. 1-7). IEEE. <https://doi.org/10.1109/EASCT59475.2023.10393443>
- Rasheed, I., & Mostafa, H. (2025). Federated learning-based anomaly detection for zero-day attack prevention in 6G network slices. *Computer Networks*, 268, 111376. <https://doi.org/10.1016/j.comnet.2025.111376>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>
- Saied, M., & Guirguis, S. (2025). Explainable Artificial Intelligence for botnet detection in internet of things. *Scientific Reports*, 15, 7632. <https://doi.org/10.1038/s41598-025-90420-6>
- Saied, M., Guirguis, S., & Madbouly, M. (2023). A comparative analysis of using ensemble trees for botnet detection and classification in IoT. *Scientific Reports*, 13, 21632. <https://doi.org/10.1038/s41598-023-48681-6>
- Santhadevi, D., & Janet, B. (2023). Stacked deep learning framework for edge-based intelligent threat detection in IoT network. *The Journal of Supercomputing*, 79(11), 12622-12655. <https://doi.org/10.1007/s11227-023-05153-y>
- Savant, R. V., M, S., Mohan, P. K., Bhaskaran, S., & Rajagopal, S. M. (2024). A comparative analysis of black-box and white-box models for IoT botnet detection. In 2024 IEEE 9th International Conference for Convergence in Technology (I2CT). IEEE. <https://doi.org/10.1109/I2CT61223.2024.10544270>
- Serhane, A., Ibrahim, K., Hamzaoui, E.-M., Jouhari, M., & Ben-Othman, J. (2025). IoT intrusion detection using machine learning classifiers and PCA dimensionality reduction for N-BaIoT dataset. In ICC 2025-IEEE International Conference on Communications (pp. 5809-5814). IEEE. <https://doi.org/10.1109/ICC52391.2025.11161305>
- Son, B. D., Hoa, N. T., Chien, T. V., Khalid, W., Ferrag, M. A., ..., & Debbah, M. (2024). Adversarial attacks and defenses in 6G network-assisted IoT systems. *IEEE Internet of Things Journal*, 11(11), 19168-19187. <https://doi.org/10.1109/JIOT.2024.3373808>
- Steinbach, M., & Tan, P.-N. (2009). kNN: K-nearest neighbors. In X. Wu & V. Kumar (Eds.), The top ten algorithms in data mining (pp. 151-161). Chapman & Hall/CRC. <https://doi.org/10.1201/9781420089653-15>
- Trajanovski, T., & Zhang, N. (2021). An automated and comprehensive framework for IoT botnet detection and analysis (IoT-BDA). *IEEE Access*, 9, 124360-124383. <https://doi.org/10.1109/ACCESS.2021.3110188>
- Ullah, S., Wu, J., Lin, Z., Kamal, M. M., Mostafa, H., ..., & Chuah, T. C. (2025). Comparative analysis of deep learning and traditional methods for IoT botnet detection using a multi-model framework across diverse datasets. *Scientific Reports*, 15, 31072. <https://doi.org/10.1038/s41598-025-16553-w>
- Wang, D., Zhang, Y., & Zhao, Y. (2017). LightGBM: an effective miRNA classification method in breast cancer patients. In Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics (pp. 7-11). Association for Computing Machinery. <https://doi.org/10.1145/3155077.3155079>

An integrated deep learning approach for 3D urban scene generation using monocular satellite imagery

Ömer Can Karadağ^{*1}, Nihal Altuntaş²

¹Department of Software Engineering, Faculty of Engineering and Architecture, İstanbul Gelişim University, İstanbul, Türkiye

²Department of Computer Engineering, Faculty of Engineering, İstanbul University-Cerrahpaşa, İstanbul, Türkiye

Cite this article as: Karadağ, Ö. C., & Altuntaş, N. (2026). An integrated deep learning approach for 3D urban scene generation using monocular satellite imagery. *J Comp Electr Electron Eng Sci*, 4(2), 56-63. <https://doi.org/10.51271/JCEES-0037>

Received: 09.12.2025

Accepted: 27.05.2026

Published: 13.10.2026

ABSTRACT

Aims: The aim of this study is to build a pipeline leveraging deep learning for 3D reconstruction of urban environments using only single high-resolution satellite images, while comparing strengths and weaknesses of the proposed system with its alternatives like multi-view or LiDAR-based methods.

Methods: We proposed a pipeline system consisting of several steps with different deep learning models. Each step works independently to complete different tasks. Firstly, the system generates digital surface model (DSM) using a Pix2Pix conditional GAN with perceptual loss, then a mask R-CNN model is used to build segmentation and finally a ResNet-50 model classifies roof type of buildings. Models were trained on SpaceNet Urban 3D, Inria aerial labeling, and GATE roof datasets, and MSE, IoU/mAP and accuracy/F1 scores are preferred as evaluation metrics for elevation accuracy, segmentation quality and classification performance, respectively.

Results: DSM generation achieved MSE of 986.70 (mean absolute deviation of 31 gray levels), with 90% of pixels maintaining errors below 50 levels. Instance segmentation attained 0.712 mAP@IoU50, declining to 0.358 at stricter IoU 75 thresholds, revealing boundary precision challenges. Roof classification reached 80% accuracy with strong performance on common types (hip roofs: F1=0.91). The system synthesizes these outputs into exportable 3D scenes combining terrain meshes, extruded building geometries, and vegetation elements.

Conclusion: This work validates that monocular satellite imagery can be sufficiently used as a sole input for a system using different deep learning models to generate functionally useful 3D urban models. The proposed system offers a practical solution for rapid urban analysis when additional data like multi-view image or laser scan is unavailable. The coordinated integration of multiple deep learning components yields promising results for urban reconstruction using conventional satellite imagery while geometric precision remains below production-grade standards.

Keywords: Digital surface model, Pix2Pix, mask R-CNN, remote sensing

INTRODUCTION

The demand for satellite imagery continues to increase each year. The information extracted from these images supports a wide range of applications including urban planning, disaster management, telecommunication infrastructure, solar power potential assessment and military strategy development. Recent advances in deep learning and computer vision have significantly expanded the types and quality of information that can be derived from static imagery. As a result, both the use of satellite data and research on deep learning methods for satellite image analysis have grown substantially.

However, the two-dimensional nature of these images imposes significant limitations. The ability to generate accurate 3D models from existing 2D data can unlock new possibilities for analysis and visualization. Until recent years, this domain was dominated by traditional methods such as LiDAR (light detection and ranging), photogrammetry and structure-from-motion (SfM). These methods are renowned for producing

highly accurate and detailed 3D models. LiDAR actively measures distances with laser pulses, directly generating dense point clouds, while photogrammetry and SfM derive 3D structure by identifying and triangulating corresponding points across multiple overlapping images. [Table 1](#) provides a comparison of traditional 3D reconstruction methods to contextualize their strengths and limitations.

Despite their proven accuracy, these conventional approaches present significant practical limitations. LiDAR surveys, whether airborne or terrestrial, require specialized, costly hardware and are often logistically complex to deploy. Similarly, multi-view photogrammetry and SfM are heavily dependent on the availability of high-quality, overlapping imagery from multiple satellite passes or aerial flight paths. This dependency not only increases data acquisition costs but also complicates and delays the reconstruction process,

*Corresponding Author: Ömer Can Karadağ, krdg.omercan@hotmail.com



Table 1. Comparison of traditional 3D reconstruction methods

Feature	Photogrammetry	LiDAR	SfM
Principle	Image analysis	Laser scanning	Multi-view geometry
Data source	RGB images	Point cloud	RGB images
Accuracy	Medium-high	Very high	Medium-high
Cost	Low-medium	High	Low-medium
Processing time	Slow-medium	Fast	Slow
Texture/color information	Rich	Limited	Rich
Key limitations	Weather conditions, textureless areas, shadows	High cost, limited texture	Computational speed, moving objects, image quality
Key applications	Visualization, mapping	High-precision modeling	Research, volume estimation, change detection

3D: Three-dimensional, LiDAR: Light detection and ranging, SfM: Structure-from-motion, RGB: Red, green, blue

making it infeasible for rapid, large-scale analyses in areas where such multi-view data is unavailable.

In response, deep learning has emerged as a powerful tool for automating complex feature extraction from satellite imagery. Significant research has demonstrated its capability to perform specific tasks such as estimating digital surface models (DSMs) (Chen et al., 2023; Liu et al., 2020), extracting instance-level building footprints (Wei et al., 2023), and classifying architectural attributes like roof types (Buyukdemircioglu et al., 2021) directly from single optical images. However, these advancements have largely progressed in isolation, solving discrete problems without converging on a unified solution for holistic 3D scene generation.

This study proposes and evaluates a novel pipeline that leverages deep learning to reconstruct 3D urban environments from a single monocular satellite image, a challenging, yet cost-effective alternative to traditional methods like stereo photogrammetry or LiDAR. The primary contribution of this work is the design and evaluation of a modular deep learning pipeline for monocular 3D urban reconstruction rather than achieving meter-level elevation accuracy. While the pipeline integrates several deep learning models to extract and assemble the necessary components for a complete 3D scene, the central goal of this work is to critically evaluate the feasibility of such fully monocular approaches. We systematically analyze the performance and crucially, the limitations and weaknesses inherent in each component of the pipeline. By providing detailed assessment of what is achievable with current methods and where the principal bottlenecks lie, this study aims to establish a clear benchmark and pave the way for more robust and accurate monocular reconstruction systems in the future.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Study Design

We propose an integrated, monocular pipeline for 3D urban reconstruction from a single satellite image. The core challenge of this task lies in inferring a complete 3D scene encompassing terrain, buildings, and vegetation from a single 2D RGB input, an inherently ill-posed problem. To address this, our methodology decomposes the problem into a sequence of more tractable sub-tasks, each handled by a specialized deep learning model. This modular approach allows for targeted optimization of individual components such as DSM generation, building segmentation, and roof type classification before their coordinated integration. The overall pipeline is designed to sequentially extract and assemble the necessary geometric and semantic components, which are subsequently fused to construct a coherent 3D model. The following subsections detail the architecture, training, and implementation of each component.

To infer elevation from a single RGB image, we employed a conditional Generative Adversarial Network (cGAN) based on the Pix2Pix architecture (Isola et al., 2017), which is designed for image-to-image translation tasks. The model learns a mapping from an input RGB satellite image to a corresponding DSM.

The Pix2Pix framework consists of two core components: a generator and a discriminator, which are trained adversarially.

The generator is based on a U-Net architecture (Ronneberger et al., 2015). Its symmetric encoder-decoder structure compresses the input image into a latent representation via a series of convolutional layers (with batch normalization and Leaky ReLU/ReLU activations) and then reconstructs it to the target output size through transposed convolutions. A critical feature of U-Net is its use of skip connections, which concatenate feature maps from the encoder to the corresponding decoder layers. This helps preserve low-level details and enables the generation of sharper, more detailed output DSMs. To mitigate overfitting, dropout layers were incorporated in some of the upsampling blocks. The final layer uses a Tanh activation function to produce a normalized, single channel DSM output.

The discriminator utilizes a PatchGAN architecture. Instead of classifying the entire image as real or generated, it operates on local image patches, classifying each $N \times N$ patch independently. This forces the generator to produce realistic local textures and high-frequency details. The discriminator takes a concatenated channel of the input RGB image and either the real or generated DSM. Its architecture comprises convolutional layers with spectral normalization, instance normalization, and Leaky ReLU activations, progressively reducing the spatial dimension to produce a matrix of real/fake predictions for each patch.

The model was trained with a composite loss function. The adversarial loss, provided by the discriminator, uses Binary Cross-Entropy with Logits to distinguish real from generated DSMs. The total discriminator loss L_D is the average of the losses from real and generated pairs (Eq. 1)

$$L_D = 0.5 \times (L_{BCE}(D(x, y), 1) + L_{BCE}(D(x, G(x)), 0)) \quad (1)$$

The generator's loss L_G is a weighted combination of the adversarial loss L_{GAN} (Eq. 2) and a custom designed Elevation Loss L_{Elev} (Eq. 3).

$$L_{GAN} = L_{BCE}(D(x, G(x)), 1) \quad (2)$$

$$L_G = L_{GAN} + \lambda \times L_{Elev} \quad (3)$$

Elevation Loss is a key contribution to the training process. It is a composite function that integrates:

- A standard L1 loss for pixel-wise accuracy
- A weighted mechanism that assigns greater importance to a predefined critical elevation range (144-200 for our model), focusing the model's learning on structurally significant building heights.
- A VGG19-based perceptual loss to enhance structural and semantic similarity between the predicted and ground-truth DSMs.

The Adam optimizer was used with a learning rate of 1e-3 and betas of (0.5, 0.999). During training, input and target images were resized to 512x512 pixels. Data augmentation, including color jitter, was applied to the input RGB images. All images were normalized, and the data was processed in batches.

The generated DSM represents the elevation of the uppermost surfaces. To isolate the bare-earth elevation, a digital terrain model (DTM) was subsequently derived from the DSM using a cloth simulation filter (Zhang et al., 2016).

For the task of extracting building footprints, we employed mask R-CNN (He et al., 2020), a leading deep learning model for instance segmentation. This framework was selected for its ability to not only classify pixels as buildings but also to distinguish between individual building instances, which is critical for modeling separate structures in dense urban areas.

The model architecture utilized a ResNet-50 backbone fortified with a feature pyramid network (FPN). FPN enhances the model's capability to detect and segment objects at various scales by leveraging multi-scale feature maps, a crucial feature for accurately identifying buildings of different sizes in satellite imagery. The model was initialized with weights pre-trained on the COCO dataset to leverage its robust feature extraction capabilities. We then fine-tuned the entire network on our curated building dataset. Both the box predictor head (responsible for object classification) and the mask predictor head (responsible for generating pixel-wise masks) were adapted to our specific binary classification task, distinguishing only between building and background.

The model was optimized using stochastic gradient descent (SGD). As a multi-task framework, mask R-CNN is trained to minimize a combined loss function L_{total} that integrates several components (Eq. 4)

$$L_{total} = L_{rpn_cls} + L_{rpn_box} + L_{cls} + L_{box} + L_{mask} \quad (4)$$

where:

- L_{rpn_cls} and L_{rpn_box} are the classification and regression losses from the Region Proposal Network (RPN).
- L_{cls} and L_{box} are the classification and bounding-box regression losses for the detected objects.
- L_{mask} is the average binary cross-entropy loss for the mask head, which is calculated for each proposed region of interest.

During inference, the raw masks generated by mask R-CNN often exhibited irregular and noisy boundaries. To produce geometrically consistent footprints suitable for 3D modeling, a post-processing pipeline was applied to each predicted instance mask. This involved:

- **Morphological closing:** A morphological closing operation with a 5x5 rectangular kernel was used to fill small gaps and smooth the edges of the binary masks.
- **Contour extraction:** The largest connected contour was extracted from the refined mask.
- **Boundary simplification:** The contour was simplified using the Douglas-Peucker algorithm to reduce vertex count while preserving the overall shape, resulting in a regularized polygon.
- **Polygon filling:** The simplified polygon was finally filled to produce a clean, geometrically consistent building footprint.

For roof type classification, we implemented a ResNet-50 architecture (He et al., 2016) pre-trained on ImageNet. The model was adapted through targeted fine-tuning: only the final residual block (layer4) and classification head were unfrozen to preserve general features while adapting to roof-specific characteristics. The final fully-connected layer was replaced with a custom classifier (512 units, ReLU, Dropout p=0.3) matching the number of roof classes.

To address class imbalance, we employed weighted random sampling based on inverse class frequencies. The model was trained with cross-entropy loss using Adam optimizer with dual learning rates (1e-5 for backbone, 1e-4 for classifier). Training utilized standard data augmentation (random cropping, flipping, color jitter), and the best model was selected based on validation accuracy over 25 epochs (batch size=32).

Tree crown detection is performed using the DeepForest model (Weinstein et al., 2019; Weinstein et al., 2020) to identify vegetation locations and enhance the visual completeness of the reconstructed scene. Unlike the three core components, DeepForest serves a supplementary visualization role and was not quantitatively evaluated.

The final 3D scene was constructed by integrating outputs from all deep learning components using the Open3D and Trimesh libraries. The pipeline transforms 2D model predictions into a coherent 3D urban model through several systematic stages.

A base terrain mesh was created using the DTM derived from the Pix2Pix-generated DSM. Each DTM pixel was converted to a 3D vertex, with X-Y coordinates normalized and Z-values scaled by a height factor to approximate real-world elevation. Triangular faces connect these vertices to form a continuous surface, textured using the original RGB imagery and equipped with calculated surface normals for realistic lighting.

Building footprints from mask R-CNN instance segmentation masks were extruded into 3D volumes. Base heights were determined from average DTM values within each footprint, while building heights were calculated from normalized DSM values within a predefined realistic range. Walls were generated by connecting base and roof contours, with roof structures shaped according to ResNet-50 classifications.

Complex roof geometries were triangulated using Delaunay algorithms, and separate textures were mapped to walls and roofs using UV coordinates.

Tree locations identified by DeepForest were incorporated using template tree models in GLB format. Each detected tree was positioned at its bounding box center, scaled to appropriate dimensions, and grounded to the terrain using local DTM elevation data.

The complete scene, combining terrain, buildings, and vegetation, was exported in GLB format for visualization and interoperability with web-based viewers and 3D modeling software.

The data used for this research was sourced from several publicly available datasets. For the task of DSM generation, SpaceNet Urban 3D challenge dataset (Goldberg et al., 2017) was used. The dataset consists of high-resolution aerial images of three different cities in RGB and DSM format. This dataset also contains ground-truth instance labels of buildings, which were used in building segmentation task of this project. For the building segmentation task; previously mentioned Urban 3D Challenge was used alongside Inria Aerial Labeling Dataset (Maggiori et al., 2017). This dataset consists of high-resolution aerial orthorectified imagery as well as the building mask images. The dataset covers an area of 810 km² and was added to increase the image size and help the model understand different types of buildings. The roof classification model was trained using the dataset provided by GATE institute, consisting of orthophotos in 10 cm/pix resolution (Hristov et al., 2023). A small roof classification dataset from RoboFlow was also added into the mix (Surya, 2024).

The performance of each component in the pipeline was assessed quantitatively. The accuracy of the DSM generation was measured using mean squared error (MSE), root MSE and Structural Similarity Index (SSIM). The building segmentation performance of the mask R-CNN model was evaluated using the intersection over union (IoU) metric. After inference, each predicted building mask was post-processed to reduce noise and irregular boundaries. The binary masks were refined using morphological closing with 5x5 rectangular kernel to fill small gaps and smooth the edges. The largest connected contour was then extracted and simplified using the Douglas-Peucker algorithm followed by polygon filling to produce geometrically consistent footprints. For the roof type classification, standard metrics including accuracy, precision, recall, and F1-score were calculated. Because the tree detection model lacks ground-truth tree location data, no metric calculations were performed at that stage.

RESULTS

The results demonstrate the viability of the monocular approach for 3D reconstruction. The Pix2Pix-based model for DSM generation was iteratively improved over four versions. The final version, which incorporated a VGG19 based perceptual loss function, achieved the best performance with a final MSE of 986.70, RMSE of 31.4 and SSIM of 0.697 on the test dataset. The MSE was computed by comparing the absolute per-pixel intensity differences between predicted and ground-truth DSMs across all test images (8-bit scale, range 0-255). The SpaceNet Urban 3D dataset originally provides DSM values in meters. During preprocessing, these were normalized to 8-bit (0-255) for compatibility with our

Pix2Pix implementation. Based on typical elevation ranges for Jacksonville (approximately 0-50 meters), 1 gray level corresponds to roughly 0.2 meters, and our RMSE of 31.4 gray levels translates to approximately 6.2 meters. All quantitative results in this study are reported on the normalized scale for consistency with the training pipeline. Each pixel error was squared and averaged across the dataset. A quantitative comparison, shown in Figure 1, reveals that the model using VGG19 based perceptual loss produces significantly sharper and more structurally accurate elevation maps compared to initial versions, which suffered from more pronounced blurring and artifacts. Although the absolute value appears numerically high, it corresponds to an average absolute deviation of approximately 31 gray levels per pixel, or about 12% of the full dynamic range. This indicates that the majority of regions maintain moderate accuracy, while a smaller subset of pixels with large deviations, amplified by the squaring operation, contributes disproportionately to the overall error. Approximately 34% of the test pixels exhibit an absolute difference within 0-5 gray levels, suggesting locally stable performance despite the global MSE being influenced by outliers and high-contrast regions. Table 2 shows that 90% of pixels had an absolute error of less than 50 gray levels. The SSIM of 0.697 indicates that the predicted DSMs preserve structural similarity with ground-truth despite absolute pixel wise errors. To further analyze elevation accuracy across different scene components, we performed a stratified evaluation using building masks generated by our mask R-CNN model. Pixels were classified into three categories: building interiors, building boundaries (3-pixel buffer) and non-building areas. Building interiors exhibited the highest error (RMSE: 37.8, MAE: 28.2), followed by building boundaries (RMSE: 30.9, MAE: 23.1), while non-building areas showed the lowest error (RMSE: 30.4, MAE: 20.9). This confirms that elevation estimation is most challenging on building structures. To visually illustrate the spatial distribution of errors, error heatmaps from selected test samples are presented in Figure 2. The highest errors were concentrated at building edges and in areas of deep shadow, where visual cues for elevation are ambiguous.

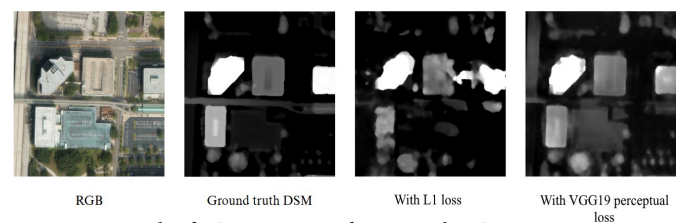


Figure 1. Example of DSM generation from a single RGB image

DSM: Digital surface model, RGB: Red, green, blue, VGG19: Visual geometry group 19-layer network

Table 2. Pix2Pix model absolute error-pixel count	
Absolute error range	Pixel count
0-5	18,559,062
5-10	6,252,400
10-20	10,001,808
20-30	6,840,507
30-50	7,375,825
50-100	4,497,386
100-150	716,163
150-255	201,466

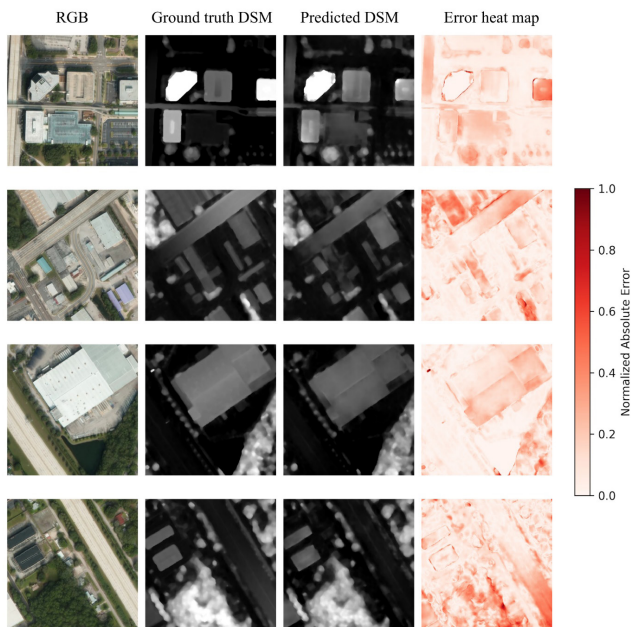


Figure 2. Error heatmaps of predicted DSM images

DSM: Digital surface model, RGB: Red, green, blue

The performance of the fine-tuned mask R-CNN model for building segmentation was evaluated using mean average precision (mAP) metric. The model achieved an mAP of 0.712 at an IoU threshold of 0.50, indicating a strong capability to correctly identify and locate building instances at a moderate overlap level. This demonstrates that the model can successfully distinguish between building and non-building areas for general detection purposes. However, the model's performance in precisely delineating boundaries was limited, as evidenced by a significant drop to 0.358 mAP at the stricter IoU threshold of 0.75. This sharp decline suggests the model struggles to define precise building contours and may face challenges in segmenting adjacent structures in dense urban environments. The overall COCO mAP (averaged over IoU thresholds from 0.50 to 0.95) was 0.383. Nevertheless, the obtained values remain acceptable for many building detection applications and were deemed sufficient for the purposes of this project, which prioritizes a complete, functional pipeline over production-grade segmentation accuracy. An example RGB image along with its ground-truth and predicted mask is shown in [Figure 3](#).



Figure 3. Comparison of ground-truth mask and predicted mask

RGB: Red, green, blue

The roof classification model achieved an overall accuracy of 80.0% across five classes. The performance was particularly strong for the most common class, hip roofs, which achieved an F1-score of 0.91. Performance varied across the remaining classes, with the model effectively identifying complex roofs (0.73 F1-score) but showing lower precision and recall for less frequent types, such as gable and flat. A pyramid class was also

included; however, it contained only three samples in the test set, making the results unreliable and not representative of the model's actual performance for this category. The weighted average F1-score was 0.80, indicating robust classification performance overall. A summary of class-wise performance is illustrated in the confusion matrix ([Figure 4](#)), which highlights both the strengths in recognizing common roof types and the limitations associated with underrepresented classes.

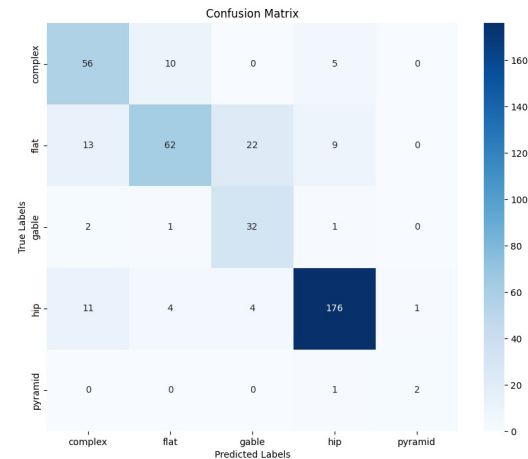


Figure 4. Confusion matrix of roof classification model

Tree crown detection was performed using the pre-trained DeepForest model to enrich the visual completeness of the reconstructed scene. The model identified vegetation locations and generated crown boundaries that were integrated into the final 3D reconstruction. As this component served a supplementary visualization role rather than a core analytical function, quantitative evaluation was not performed, as ground-truth tree annotations were unavailable.

We compared the performance of each pipeline component against recent studies employing similar methodologies. It should be noted that direct numerical comparison is challenging due to difference in datasets, evaluation metrics and output representations across studies. Nevertheless, the following comparisons provide a qualitative assessment of our method's relative performance.

For DSM generation, Ghamisi and Yokoya (2018) reported RMSE values of 3.89 m, 2.58 m and 3.66 m on Potsdam, Vaihingen and Stockholm datasets respectively, using a cGAN based approach. Our method achieves an RMSE of 31.4 on the normalized 8-bit scale, which corresponds to roughly 6.2 m. While this places our absolute accuracy below specialized urban DSM studies, several factors should be considered such as their dataset's different elevation distribution and complexity. Importantly, our SSIM of 0.697 indicates strong structural fidelity, demonstrating that perceptual loss effectively preserves elevation patterns despite the scale differences.

Table 3 compares our building segmentation results with recent studies using similar architectures. Hou and Li (2024) reported approximately 95% classification accuracy on UAV aerial imagery, while Ramadhani and Alya (2024) achieved a precision of 0.85 on UAV orthophoto data. More recently, Vincent and Varalakshmi (2024) obtained a mAP@50 of 0.79 on the WHU and Inria datasets using mask R-CNN. Although direct numerical comparison is limited by differing datasets and evaluation metrics, these results contextualize

Table 3. Comparison of recent building segmentation studies

Study	Method	Dataset	Metric	Result
Ramadhani & Alya (2024)	Mask R-CNN	UAV orthophoto	Precision	0.85
Hou & Li (2024)	Mask R-CNN (optimized)	UAV aerial	Classification accuracy	95%
Vincent & Varalakshmi (2024)	Mask R-CNN	WHU+Inria	mAP@50	0.794
This study	Mask R-CNN	Inria+SpaceNet	mAP@50	0.712

R-CNN: Region-based convolutional neural network, UAV: Unmanned aerial vehicle, WHU: Wuhan University, mAP: Mean average precision

the performance of mask R-CNN for building extraction tasks. Our lower mAP@50 of 0.71 can be attributed to the use of the SpaceNet dataset which includes more dense vegetation, presenting additional challenges compared to the datasets used in comparative studies.

Our roof classification model's performance aligns with the upper tier of results in current remote sensing literature, where architectural complexity often limits accuracy. A comparative analysis can be seen in [Table 4](#).

Table 4. Comparison of recent roof type classification studies

Study	Method	Classes	Accuracy
Buyukdemircioglu et al. (2021)	CNN	6 types	80%-86%
Shailja et al. (2025)	Swin transformer	4 types	75%
Ölçer et al. (2023)	Siamese network	3 types (flat, gable, hip)	73%
This study	ResNet-50	5 types	80%

CNN: Convolutional neural network, ResNet-50: 50-layer residual network

The integration pipeline successfully synthesized all component outputs into a coherent 3D urban model, as illustrated in [Figure 5](#). The final reconstruction demonstrates the practical viability of the monocular approach, producing a visually consistent scene with recognizable terrain features, building masses, and vegetation distribution.

**Figure 5. Final 3D scene**

3D: Three-dimensional

Qualitative assessment reveals several key outcomes; the terrain mesh accurately reflects the topographic relief of the area, building footprints maintain plausible geometric relationships, and roof types are correctly instantiated according to classification results. The scene successfully captures the overall urban layout and spatial distribution of structures.

However, systematic analysis also identified characteristic limitations. Primary failure modes include: (1) occasional building height inaccuracies in densely shadowed regions where DSM estimation proved challenging, (2) simplified roof geometries that occasionally miss intricate architectural details, and (3) minor misalignments between building footprints and terrain mesh at steep gradients. These limitations reflect the inherent constraints of monocular reconstruction

but do not fundamentally compromise the scene's utility for visualization and urban analysis applications.

The exported GLB model maintains all semantic groupings (terrain, buildings, vegetation) and proved compatible with major 3D viewing platforms, confirming the pipeline's practicality for downstream applications.

DISCUSSION

The extraction of semantically meaningful information from satellite imagery through deep learning has emerged as a prominent research direction in recent years. Within this domain, 3D reconstruction leveraging deep learning methodologies has constituted an active area of investigation. However, 3D reconstruction from satellite imagery presents distinct challenges attributable to the inherent characteristics of the data modality. Prevailing approaches in the literature predominantly depend on multi-stereo imagery (Tarabalka et al., 2022) or structure-from-motion (SfM) techniques, which utilize overlapping views to triangulate depth information through photogrammetric principles. In contrast, our monocular approach operates under significantly weaker geometric constraints, trading strict data requirements for inherent limitations in reconstruction accuracy. As Panagiotou et al. (2020) observe, single-image elevation estimation represents an ill-posed problem that inherently yields lower fidelity compared to multi-view systems. The principal advantage of our method lies in its reduced data dependency (requiring only a single satellite pass) though this practical benefit comes at the cost of diminished geometric precision. Furthermore, while off-nadir imagery has been exploited for height estimation in other studies (Li et al., 2024), our work is exclusively limited to nadir-facing orthoimagery, and consequently does not address the potential advantages of oblique viewing angles.

For DSM generation, we employ a Pix2Pix conditional generative adversarial network (cGAN) to infer elevation data from single RGB inputs. This approach aligns with established literature demonstrating that cGANs can effectively learn complex mappings from optical imagery to elevation profiles (Panagiotou et al., 2020). Our implementation initially utilized a standard L1 loss function; however, empirical evaluation revealed that a custom loss function incorporating VGG-based perceptual loss yielded superior performance in preserving structural details. Nevertheless, monocular DSM estimation remains fundamentally constrained by its inherent limitations. The absence of multiple viewpoints prevents the recovery of occluded surfaces and undercuts geometries, resulting in incomplete scene reconstruction. In practice, our Pix2Pix-generated DSMs exhibit high fidelity in planar regions with distinctive textural patterns but demonstrate reduced reliability in areas characterized by abrupt topographic transitions or shadow occlusion. Furthermore,

dataset limitations present additional challenges. Improved generalization would require more geographically diverse training data encompassing varied architectural styles and urban morphologies. Future work could explore more advanced monocular depth estimation architectures, such as DPT or MiDaS, which may offer enhanced performance for satellite-based elevation modeling.

For building segmentation, we employ mask R-CNN for instance-aware extraction, aligning with established methodologies in recent literature. This approach follows the precedent set by Zhao et al. (2018), who adopted mask R-CNN “as a base network due to its simplicity” for generating polygonal building masks from satellite imagery. The framework produces pixel-precise instance masks that enable precise rooftop delineation. This instance-level paradigm proves particularly valuable in dense urban environments, where it effectively disambiguates adjacent structures that might otherwise merge in semantic segmentation approaches. However, a recognized limitation of mask R-CNN is its tendency to generate masks with irregular, noisy boundaries. Our observations corroborate this phenomenon: while the model achieves reasonable detection accuracy (mAP 0.712 at IoU = 0.50), the resulting building contours frequently exhibit jagged or fragmented edges. Although a morphological post-processing step was implemented, it yielded only marginal improvements in IoU. This suggests that the underlying segmentation quality, rather than boundary smoothing alone, limits refinement potential. Several studies have proposed methods for regularizing building footprints (Kong et al., 2023), such post-processing refinements were not implemented in our current pipeline. Additionally, dataset limitations analogous to those affecting DSM generation persist. Improved generalization would benefit from expanded training data encompassing greater architectural diversity and geographic variation. Therefore, future work should investigate footprint regularization methods and incorporate additional training data to improve generalization across architectural styles and regions.

For roof type classification, we implemented a fine-tuned ResNet-50 convolutional neural network, a methodological choice consistent with established practices in the literature. This approach aligns with the work of Buyukdemircioglu et al. (2021), who demonstrated that a fine-tuned ResNet-50 architecture achieves competitive accuracy across six roof categories using a dataset of 10,000 roof images. Their findings confirmed the model’s efficacy on very-high-resolution aerial imagery, a result our study corroborates. We observe similar performance trends, with ResNet-50 effectively learning discriminative features for common roof types including flat, gable, and hip configurations when trained on adequate labeled exemplars. Nevertheless, classification performance remains contingent upon dataset scale and diversity, as noted in prior research. In our implementation, misclassifications predominantly occur among categories with minimal visual distinction or when roof features are partially occluded. While future enhancements could explore expanded class inventories or more sophisticated architectures, ResNet-50 provides a robust baseline, as substantiated by its consistent performance across multiple studies in the domain.

Tree crown detection was implemented using the DeepForest model, a pre-trained deep learning framework specifically designed for individual tree detection in RGB imagery. Unlike

the other pipeline components, which were trained or fine-tuned on task-specific datasets, DeepForest was deployed in its pre-trained state. While the model successfully identified tree locations and generated crown polygons for 3D scene integration, the absence of ground-truth tree annotations in our dataset precluded quantitative validation. This limitation is common in urban reconstruction pipelines where vegetation mapping is secondary to build infrastructure.

Overall, the favorable aspects of our approach are that it operates with minimal input data and integrates DSM generation, segmentation and roof classification in a single pipeline. It requires only one optical image to produce useful 3D urban models, and aligns with recent evidence that deep learning can approximate geometric reconstruction in a purely monocular setting. On the other hand, the limitations are equally clear. The DSM estimates lack the precision of multi-view methods, the segmentation masks require additional refinement to be production-ready, and the roof classifier depends on limited classes and labeled data. Our study was also restricted to high-resolution urban imagery, leaving its applicability to rural or lower resolution data untested.

CONCLUSION

As a result, the study demonstrates that a monocular deep learning-based pipeline can indeed generate useful 3D urban scenes, combining DSM prediction, building segmentation, tree crown detection and roof classification. The method reduces acquisition demands and computation compared to traditional multi-view stereo approaches, though with an expected loss in geometric accuracy. Future work should focus on integrating post-processing for building outlines, enriching the dataset and exploring additional modalities. Despite their limitations, the results show that monocular pipelines are viable for rapid urban analysis when only single view satellite data are available, echoing findings in prior literature.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

This study was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the 2209-A Research Project Support Programme for Undergraduate Students and SAYZEK-ATP.

Author Contributions

Concept: ÖCK, NA; Design: ÖCK, NA; Control: ÖCK, NA; Resources: ÖCK, NA; Materials: ÖCK, NA; Data Collection and/or Processing: ÖCK, NA; Analysis and/or Interpretation: ÖCK, NA; Literature Review: ÖCK, NA; Writing the Article: ÖCK, NA; Critical Review: ÖCK, NA.

Acknowledgments

The work was also carried out within the framework of the SAYZEK-ATP (Artificial Intelligence Talent Cluster of Defence Industry-Academic Thesis Programme). The authors would like to express their gratitude to Cevahir Çığla from ASELSAN, who served as project mentor under the SAYZEK-ATP program and provided valuable guidance throughout the study.

REFERENCES

- Buyukdemircioglu, M., Can, R., & Kocaman, S. (2021). Deep learning based roof type classification using very high resolution aerial imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B3-2021, 55-60. <https://doi.org/10.5194/isprs-archives-XLIII-B3-2021-55-2021>
- Chen, S., Shi, Y., Xiong, Z., & Zhu, X. X. (2023). HTC-DC Net: monocular height estimation from single remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 1-18. <https://doi.org/10.1109/TGRS.2023.3321255>
- Ghamisi, P., & Yokoya, N. (2018). IMG2DSM: height simulation from single imagery using conditional generative adversarial net. *IEEE Geoscience and Remote Sensing Letters*, 15(5), 794-798. <https://doi.org/10.1109/LGRS.2018.2806945>
- Goldberg, H., Brown, M., & Wang, S. (2017). A benchmark for building footprint classification using orthorectified RGB imagery and digital surface models from commercial satellites. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1-7). IEEE. <https://doi.org/10.1109/AIPR.2017.8457973>
- He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386-397. <https://doi.org/10.1109/TPAMI.2018.2844175>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778). IEEE. <https://doi.org/10.1109/CVPR.2016.90>
- Hou, T., & Li, J. (2024). Application of mask R-CNN for building detection in UAV remote sensing images. *Heliyon*, 10(19), e38141. <https://doi.org/10.1016/j.heliyon.2024.e38141>
- Hristov, E., Petrova-Antonova, D., Petrov, A., Borukova, M., & Shirinyan, E. (2023). Imagery dataset for rooftop detection and classification. *Zenodo*. <https://doi.org/10.5281/zenodo.7633595>
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5967-5976). <https://doi.org/10.1109/CVPR.2017.632>
- Kong, L., Qian, H., Wu, Y., Niu, X., Wang, D., & Huang, Z. (2023). Simplification and regularization algorithm for right-angled polygon building outlines with jagged edges. *ISPRS International Journal of Geo-Information*, 12(12), 469. <https://doi.org/10.3390/ijgi12120469>
- Li, W., Yang, H., Hu, Z., Zheng, J., Xia, G. S., & He, C. (2024). 3D building reconstruction from monocular remote sensing images with multi-level supervisions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 27728-27737). IEEE. <https://doi.org/10.1109/CVPR.2024.02702>
- Liu, C.-J., Krylov, V. A., Kane, P., Kavanagh, G., & Dahyot, R. (2020). IM2ELEVATION: building height estimation from single-view aerial imagery. *Remote Sensing*, 12(17), 2719. <https://doi.org/10.3390/rs12172719>
- Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 3226-3229). IEEE. <https://doi.org/10.1109/IGARSS.2017.8127684>
- Ölçer, N., Ölçer, D., & Sümer, E. (2023). Roof type classification with innovative machine learning approaches. *PeerJ Computer Science*, 9, e1217. <https://doi.org/10.7717/peerj-cs.1217>
- Panagiotou, E., Chochlakis, G., Grammatikopoulos, L., & Charou, E. (2020). Generating elevation surface from a single RGB remotely sensed image using deep learning. *Remote Sensing*, 12(12), 2002. <https://doi.org/10.3390/rs12122002>
- Ramadhani, A., & Alya, N. F. (2024). Building extraction from unmanned aerial vehicle imagery using mask-RCNN (case study: Institut Teknologi Sepuluh Nopember, Surabaya). *E3S Web of Conferences*, 600, 06003. <https://doi.org/10.1051/e3sconf/202460006003>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention-MICCAI 2015 (Lecture Notes in Computer Science, Vol. 9351, pp. 234-241)*. Springer. https://doi.org/10.1007/978-3-319-24574-4_28
- Shailja, Yadava, M., Dwivedi, R., & Pardeshi, P. J. (2025). Classifying roof types from orthophotos: a Swin Transformer-based approach. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-5/W2-2025, 577-584. <https://doi.org/10.5194/isprs-annals-X-5-W2-2025-577-2025>
- Surya. (2024, July). *Roof-Type-Classifier dataset* [Data set]. Roboflow Universe. <https://universe.roboflow.com/surya-dmfwf/roof-type-classifier-bafod>
- Tarabalka, Y., Girard, N., Tripodi, S., Larrosa, C., Bauchet, J. P., Fonteix, G., ..., & Madelain, V. (2022). Reconstruction of 3D environments from satellite images by AI and computational geometry for exploitation in mixed reality. *STO-MP-MSG-197*. NATO Science and Technology Organization.
- Vincent, J. M., & Varalakshmi, P. (2024). Extraction of building footprint using mask-RCNN for high resolution aerial imagery. *Environmental Research Communications*, 6(7), 075015. <https://doi.org/10.1088/2515-7620/ad5b3d>
- Wei, S., Zhang, T., Ji, S., Luo, M., & Gong, J. (2023). BuildMapper: a fully learnable framework for vectorized building contour extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197, 87-104. <https://doi.org/10.1016/j.isprsjprs.2023.01.015>
- Weinstein, B. G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., & White, E. P. (2020). DeepForest: a Python package for RGB deep learning tree crown delineation. *Methods in Ecology and Evolution*, 11(12), 1743-1751. <https://doi.org/10.1111/2041-210X.13472>
- Weinstein, B. G., Marconi, S., Bohlman, S. A., Zare, A., & White, E. P. (2019). Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. *Remote Sensing*, 11(11), 1309. <https://doi.org/10.3390/rs11111309>
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., ..., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 501. <https://doi.org/10.3390/rs8060501>
- Zhao, K., Kang, J., Jung, J., & Sohn, G. (2018). Building extraction from satellite images using mask R-CNN with building boundary regularization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 242-2424). IEEE. <https://doi.org/10.1109/CVPRW.2018.00045>

An imitation-enhanced actor-critic approach for maze navigation^aNurgül Kalaycı^{*1}, Mehmet Dinçer Erbaş²¹Department of Software Engineering, Faculty of Engineering, Iğdır University, Iğdır, Türkiye²Department of Computer Engineering, Faculty of Engineering, Bolu Abant İzzet Baysal University, Bolu, Türkiye

Cite this article as: Kalaycı, N., & Erbaş, M. D. (2026). An imitation-enhanced actor-critic approach for maze navigation. *J Comp Electr Electron Eng Sci*, 4(2), 64-71. <https://doi.org/10.51271/JCEES-0038>

Received: 27.01.2026

Accepted: 01.04.2026

Published: 13.10.2026

ABSTRACT

Aims: This paper proposes the imitation-enhanced actor-critic (IEAC) method that integrates imitation learning (IL) into the actor-critic (AC) algorithm and evaluates its effect on the learning speed of agents in both obstacle and obstacle-free mazes. The study aims to evaluate and compare the learning efficiency of IEAC against both standard AC and tabular Q-learning algorithms in diverse maze environments.

Methods: The proposed approach combines AC with a probabilistic observational imitation model that operates without direct internal access or inter-agent experience sharing. IEAC's performance was benchmarked against AC, tabular Q-learning, and behavioral cloning (BC) through comprehensive simulations in 20x20 and 50x50 maze configurations.

Results: Simulation results indicate that IEAC significantly accelerates learning and achieves higher stability compared to the standard AC and Q-learning baselines. Specifically, IEAC agents reached optimal paths with substantially fewer environmental steps. Unlike the BC baseline, which failed in obstacle-heavy settings, IEAC successfully balanced guided imitation with independent exploration to overcome local optima.

Conclusion: The findings demonstrate that augmenting the AC framework with stochastic imitation enhances convergence speed and robustness, particularly as environmental complexity increases. The proposed IEAC approach offers a scalable and efficient solution for autonomous navigation tasks where rapid learning from observation is required.

Keywords: Social learning, reinforcement learning, multi-agent systems

^aThis study was presented as a conference paper at the International Conference on ICE 2025 (ICE2025), held in Ankara, Türkiye, on November 6, 2025, and its abstract was published in the conference proceedings.

INTRODUCTION

Social learning is an innate process observed in both animals and humans. Due to the limitations and inefficiencies of individual trial-and-error learning, social learning strategies, in which individuals learn from observing or interacting with others, have evolved as advantageous mechanisms (Henrich, 2016). Imitation, one of the main forms of social learning, has attracted attention in both biological and artificial systems. In autonomous agents, imitation learning (IL) has been achieved through experience sharing (Bredeche & Fontbonne, 2022; Christianos et al., 2021; Uwano et al., 2018), learning from demonstrations provided by a leader or expert (Argall et al., 2009; Block et al., 2024; Guenter & Billard, 2007; Mathewson & Pilarski, 2017; Rao et al., 2007), or imitation through observation (Bentivegna et al., 2005; Erbas et al., 2014), which closely reflects the imitation process in humans by reproducing superior observed behaviours. The imitation strategy adopted in this study relies on observational learning, where the agent reproduces observed behaviours without direct internal access to the leader's parameters.

Reinforcement learning (RL) is a fundamental paradigm in machine learning that enables agents to learn through trial and error by updating their policies according to feedback

from rewards. Although RL enables agents to discover solutions autonomously, the exploration process often results in slow convergence, especially in complex environments. Since IL relies on community-based rather than individual exploration, integrating IL with reward-based RL algorithms—especially in multi-agent systems—offers potential benefits. Indeed, this idea has been explored extensively in the literature (Christianos et al., 2021; Codevilla et al., 2018; Erbas et al., 2014; Guenter & Billard, 2007; Guo et al., 2019; Ndousse et al., 2021; Nguyen, 2024; Sallab et al., 2017; Uwano et al., 2018; Yang et al., 2024).

The maze solving problem is one of the most widely used benchmarks for testing learning algorithms, with applications in robotics and autonomous navigation (Rao et al., 2007; Erbas et al., 2014; Uwano et al., 2018; Ndousse et al., 2021; Sharma et al., 2023). Using mazes with obstacles increases the complexity of navigation, making the task more representative of real-world conditions and allowing a clearer evaluation of algorithmic performance and generalization.

This study proposes the imitation enhanced actor-critic (IEAC) model, which combines IL with the actor-critic (AC) algorithm to improve learning efficiency. The aim of this

*Corresponding Author: Nurgül Kalaycı, nurgul.kalayci@igdir.edu.tr



study is to simulate and compare the learning performance of agents navigating from a starting point to a goal in both obstacle and obstacle-free maze environments. The primary objective is to evaluate the performance disparities between the standard AC and the proposed IEAC models, thereby demonstrating the efficacy of the IEAC approach across diverse maze configurations.

Rather than proposing a fundamentally new learning paradigm, this study investigates a simple imitation mechanism integrated into the classical AC framework. Unlike complex hybrid IL+RL models that rely on continuous experience sharing or deep neural networks, the proposed approach isolates the impact of storing the leader's path in a discrete memory and allowing the follower agent to utilize it through a probabilistic mode-switching mechanism. The primary objective is to experimentally analyze the isolated effect of this stochastic imitation on learning speed and stability in maze navigation tasks.

The following section introduces the hybrid learning method and the problem definition. Then, the experimental results are presented, followed by a concluding discussion.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Study Design

This study examines the goal-reaching performance of two agents in the same maze environment: one is trained with the classical AC algorithm and the other with the proposed IEAC model. This experimental setup allows a direct comparison of learning speed and stability. Before introducing the experimental environment, some definitions are useful for clarifying the agents' learning mechanisms.

Prediction problems in RL can be addressed using different estimation approaches, most notably Monte Carlo (MC) methods and temporal difference (TD) learning. While MC methods rely on complete episodes to update value estimates, TD learning performs incremental updates based on bootstrapped predictions. Among these, TD methods are particularly effective because they update predictions based on subsequent estimates. A specific and widely used form of TD learning is TD (0), which performs single-step updates based on the current reward and the estimated value of the next state. The core of this approach is the TD-error denoted by δ_t , whose value at time t is defined as:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t), \quad (1.1)$$

where R_{t+1} is the reward received after transitioning from state S_t to S_{t+1} , γ is the discount factor determining the importance of future rewards, $V(S_{t+1})$ is the estimated value of the next state, and $V(S_t)$ is the current estimate. This error updates $V(S_t)$ toward a more accurate value (Sutton & Barto, 2014). The actor and critic update their policy and state-value function according to:

$$V(S_t) \leftarrow V(S_t) + \alpha_c \times \delta_t \quad \text{and} \quad \pi(A|S_t) \leftarrow \pi(A|S_t) + \alpha_a \times \delta_t, \quad (1.2)$$

where $\alpha_c=0.1$ and $\alpha_a=0.2$ are the learning rates for the critic and actor, respectively. The discount factor is $\gamma=0.8$ and the exploration followed an ϵ -greedy policy with $\epsilon=0.1$, for limited random exploration.

The experimental environment consists of two both 20x20 and 50x50 mazes: one without obstacles (empty maze, **Figure 1 and 2** for 20x20 and 50x50 dimensions, respectively.) and the other containing obstacles at specific locations (obstacle maze, see **Figure 3 and 4** for 20x20 and 50x50 dimensions, respectively). The agents, without any direct information exchange or leader-follower interaction, attempt to reach the goal at the bottom-right corner starting from the top-left corner. As shown in **Figure 1**, the agents can move in eight different directions. The shortest paths leading to the goal in the empty and obstacle mazes are illustrated in **Figure 2 and 4**, respectively. At each step, agents receive a reward of +300 upon reaching the goal, a penalty of -10 for hitting a wall or crossing the maze boundary, and a step cost of -0.5 for each movement. If an agent fails to reach the goal within 50 steps, it returns to the starting position and begins a new trial. Each trial includes 90,000 steps, and the agents update their policies over a total of 100 trials.

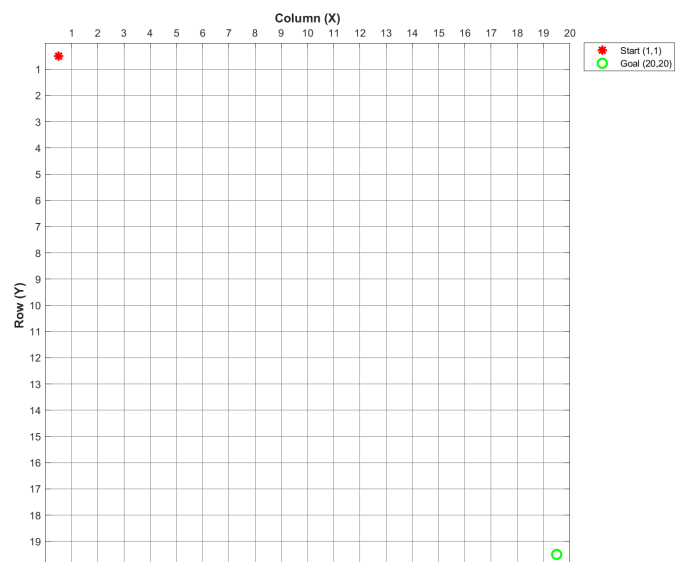


Figure 1. The 20x20 maze without obstacles

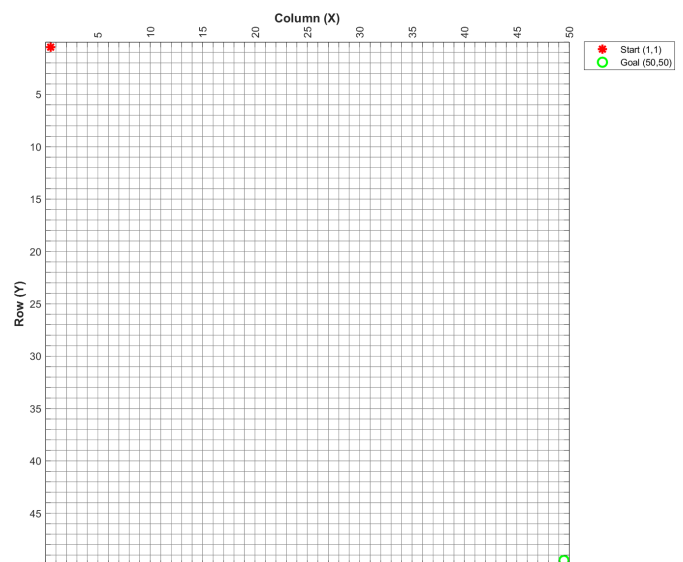


Figure 2. The 50x50 maze without obstacle

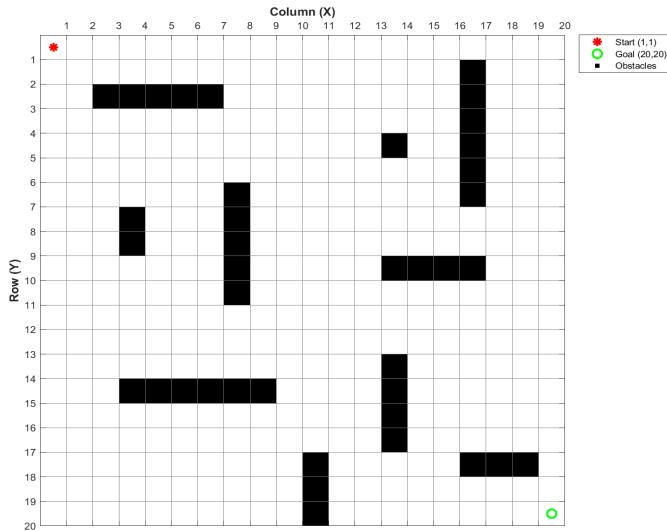


Figure 3. The 20x20 maze with obstacles

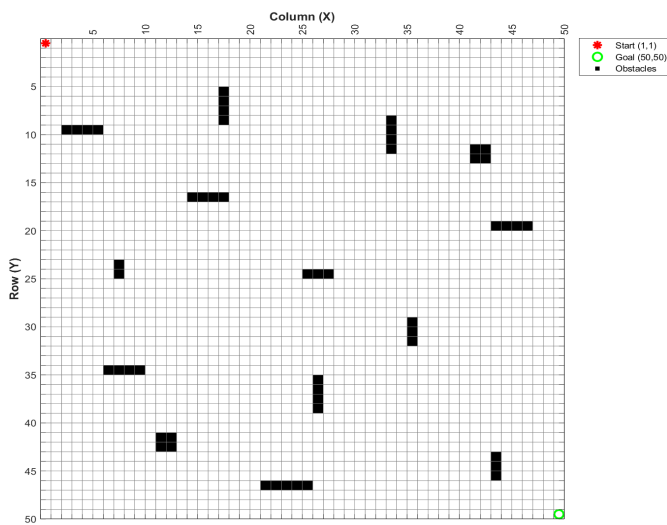


Figure 4. The 50x50 maze with obstacles

In the AC algorithm, the actor is responsible for action selection, while the critic estimates state values. That is, the actor updates its policy based on the temporal-difference (TD) error, while the critic adjusts the state-value function using the same error value.

In this study, the tabular AC method is utilized for the training of the agents, where state and action spaces are maintained in discrete matrices. To balance the exploration-exploitation trade-off, an ϵ -greedy ($\epsilon=0.1$) approach is adopted. The agent does not use a probabilistic distribution like Softmax during the decision-making phase; instead, it directly selects the action with the highest preference value during exploitation. Policy parameters are updated directly using the TD error (δ), without relying on function approximation or deep neural networks. The single-step AC algorithm developed by Sutton and Barto (2014) is provided below as a baseline:

Pseudocode of AC Algorithm

Input:

- State-action preference matrix: $P(s,a)$
- State-value matrix: $V(s)$

Parameters:

- Learning rates : $\alpha_a, \alpha_c > 0$
- Discount factor: $\gamma \in (0,1)$

Loop each experiment:

- Initialize S
 - Loop for each step: (if S is not terminal)
 - Select action A using ϵ -greedy based on $P(S, \cdot)$
 - Take action A , observe R and next state S'
 - $\delta = R + \gamma V(S') - V(S)$, (When S_{t+1} is terminal $V(S_{t+1}, w) = 0$)
 - $V(S) \leftarrow V(S) + \alpha_c \delta$
 - $P(S, A) \leftarrow P(S, A) + \alpha_a \delta$
 - $S \leftarrow S'$

In this study, the first agent is trained using the standard AC algorithm and records its best-performing path. The path coordinates generated by this agent are stored in a static memory matrix, designated as Path, which serves as the reference for the imitation process. Within each run, a path is considered successful and stored in memory only if the agent reaches the goal and the path length is shorter than the currently recorded path.

In the IEAC model, the second agent operates in two modes: a normal mode and an imitation (copy) mode. In normal mode, the agent learns through its own AC policy, whereas in imitation mode, it observes and occasionally imitates the successful actions of the first agent. The agent switches between modes when it collides with a wall, moves outside the maze, or exceeds 30 consecutive steps in the same mode. This mechanism prevents continuous imitation and ensures a more dynamic learning process.

The initial mode is selected probabilistically based on a predefined imitation probability, p ($p \in \{0.1, 0.3, 0.5\}$). Before each trial begins, the probability of imitation determines whether the agent starts in imitation or normal mode. During learning, agents compute the TD-error based on their actions to update both the state-value function and the policy. The detailed pseudocode for this procedure, incorporating the probabilistic initialization and update rules, is given in the following algorithm.

When the follower agent (agent 2) enters the copy mode, an alignment step is executed if its current position (S_t) is not on the recorded reference Path. The agent does not teleport to the beginning of the path or start copying blindly; instead, it calculates the Euclidean distance between its current position and all nodes on the Path to identify the closest node ($S_{closest}$). The next action is then determined by $A_t = \text{sign}(S_{closest} - S_t)$, ensuring the agent takes the shortest step toward the reference path. This step is not an independent heuristic but a geometric alignment necessary for meaningful imitation. Furthermore, to address how different distance metrics affect the learning curves, Euclidean and Manhattan distance metrics were compared (Figure 5 and 6). The analysis reveals that while the distance metric does not alter the ability to reach the optimal solution, using the Euclidean metric provides slightly faster initial learning across all imitation probabilities.

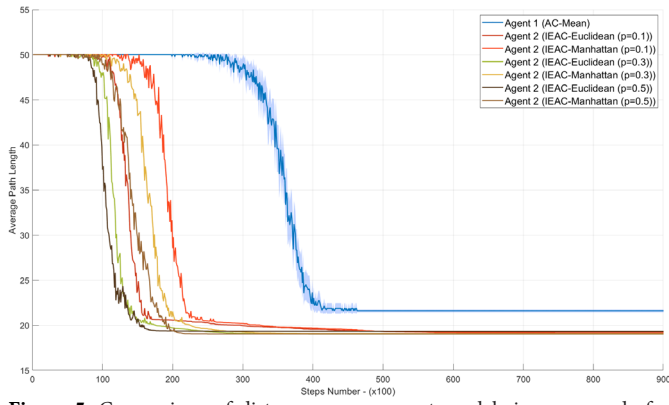


Figure 5. Comparison of distance measurement models in copy mode for 20x20 empty maze

AC: Actor-critic, IEAC: Imitation enhanced actor-critic

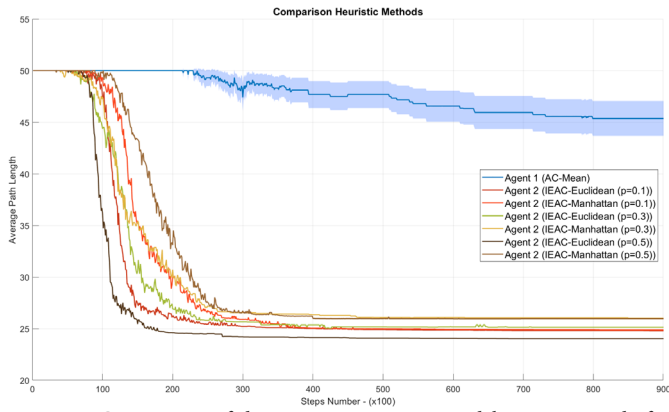


Figure 6. Comparison of distance measurement models in copy mode for 20x20 obstacle maze

AC: Actor-critic, IEAC: Imitation enhanced actor-critic

Pseudocode of Imitation Enhanced Actor-Critic (IEAC) Algorithm

Input:

- **State-action preference matrix:** $P(s, a)$ and State-value matrix: $V(s)$

Parameters:

- **Learning rate:** $\alpha_a, \alpha_c > 0$ (for actor and critic respectively)
- **Discount factor:** $\gamma \in (0, 1)$
- **Imitation probability:** $p \in \{0.1, 0.3, 0.5\}$

Memory:

Path: Optimal path matrix recorded by the Leader agent (stores state-action pairs).

For each experiment do:

Agent 1 training (leader):

- Learn for t time steps using AC algorithm (see above pseudocode.)
- If (target reached) & (current path is shortest) then
 - $Path \leftarrow \{(s_0, a_0), (s_1, a_1), \dots\}$

Agent 2 training (observer):

- Initialize: S_0 at $t = 0$ in start position)
- While S is not terminal (target) do
 - Select mode (copy or normal) based on probability p .

- If (mode=copy) then
 - **Check:** Is $S \in Path$?
 - If ($S \in Path$) then
 - $A \leftarrow Path(S)$ (select recorded action)
 - else
 - $A \leftarrow ClosestState(S)$ (select appropriate action to approach the path)
- else
 - Sample action $A_t \sim P(s, a)$ (standard action selection)
- Execute action A_t , observe reward R_{t+1} and new state S_{t+1} .
- If ($R_{t+1} = R_{collision}$) or (mode_step_count > max_step) then
 - Switch mode (normal $\Rightarrow$ copy or copy $\Rightarrow$ normal)
- $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ (If S_{t+1} is terminal, $V(S_{t+1}) = 0$)
- $V(S_t) \leftarrow V(S_t) + \alpha_c \delta_t$
- $P(S_t, A_t) \leftarrow P(S_t, A_t) + \alpha_a \delta_t$
- $S_t \leftarrow S_{t+1}$ and $t \leftarrow t + 1$

The mode_step_count variable defined in the IEAC algorithm tracks the number of consecutive steps the follower agent spends in the selected mode. This variable establishes an upper limit (max_step) to prevent the agent from getting stuck in an initial suboptimal mode or blindly copying the entire path without integrating its own exploration. To determine the optimal upper limit, a sensitivity analysis was conducted evaluating thresholds of 15, 30, and 45 steps (Table). The analysis reveals that setting the limit to 15 steps results in rapid convergence (12.100 steps) but yields a suboptimal final path length (25.90) with higher variance. Conversely, a limit of 45 steps guarantees a 100% success rate with very low variance (1.36) but drastically slows down convergence (43,700 steps) and restricts the agent's exploration, resulting in a longer path (27.00). Therefore, max_step=30 was selected as the optimal threshold, providing the best balance: achieving the shortest final path length (25.44) and rapid convergence (17,800 steps) while maintaining a high success rate (98%).

Table. Sensitivity analysis for mode switching threshold

Max_step	Convergence step	Final path length	SD (stability)	Success rate (%)
15	12.100	25.90	4.33	97%
30	17.800	25.44	4.27	98%
45	43.700	27.00	1.36	100%

SD: Standard deviation, Max: Maximum

The overall working principle of the model is presented visually in Figure 7. The following section presents the comparative results of the AC, IEAC, behavioral cloning (BC) and Q-learning.

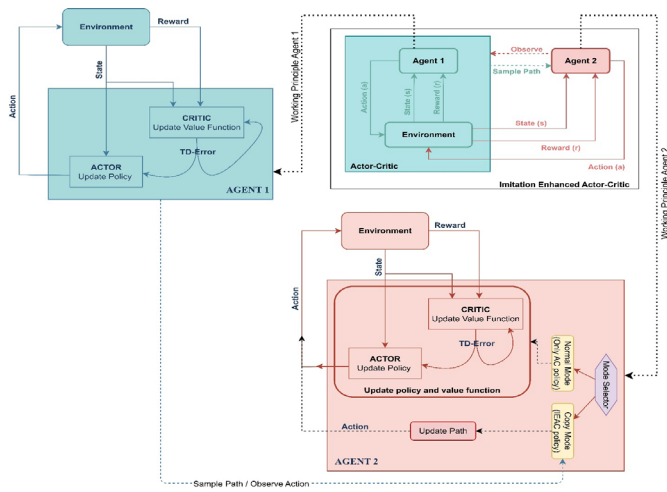


Figure 7. Diagram of working principle for IEAC method

IEAC: Imitation enhanced actor-critic

RESULTS

To ensure the statistical reliability of the results, each experiment was conducted over \$ n=100 \$ independent runs with different random initial conditions. Instead of an episode-based evaluation, a continuous step-based evaluation was utilized. The x-axis in the graphs represents every 100 steps taken by the agents. Every 100 steps, the shortest path from the start to the goal is calculated and recorded using the agent's current policy matrix. The shaded areas in the graphs denote the standard deviation (SD) and the 95% confidence interval (CI). In the empty maze (Figure 8), the standard AC algorithm started learning in approximately 30,000 steps and converged to the optimum path length around 40,000 steps. The agent using the IEAC method, however, started learning much earlier (at approximately 10,000 steps for \$ p=0.5 \$) and converged to the shortest path length in roughly 18,000 steps. As illustrated in Figure 8, the IEAC agent reaches the optimal path approximately 22,000 steps earlier than the standard AC agent. Furthermore, to evaluate the efficacy of the IEAC method from a broader perspective, comparative analyses were performed alongside tabular Q-learning and BC. As observed in Figure 8 and 9, although the Q-Learning method provides an improvement over standard AC, it fails to achieve the early convergence provided by the IEAC method. The BC method, which relies solely on imitation, completely fails in complex obstacle environments due to its inability to conduct independent exploration.

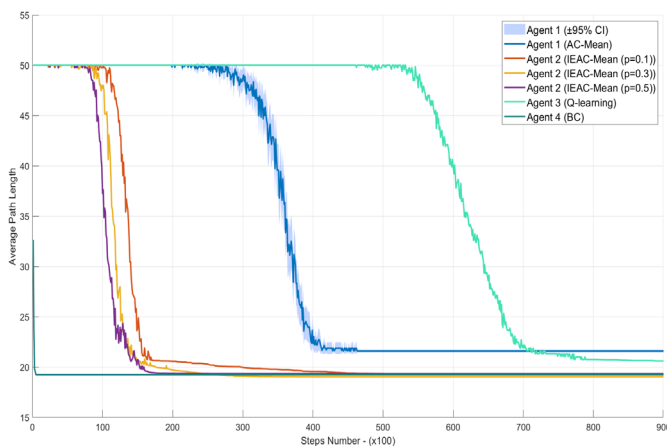


Figure 8. Comparison learning curves of AC, BC, Q-learning, and IEAC with different probabilities in the 20×20 empty maze

AC: Actor-critic, BC: Behavioral cloning, IEAC: Imitation enhanced actor-critic

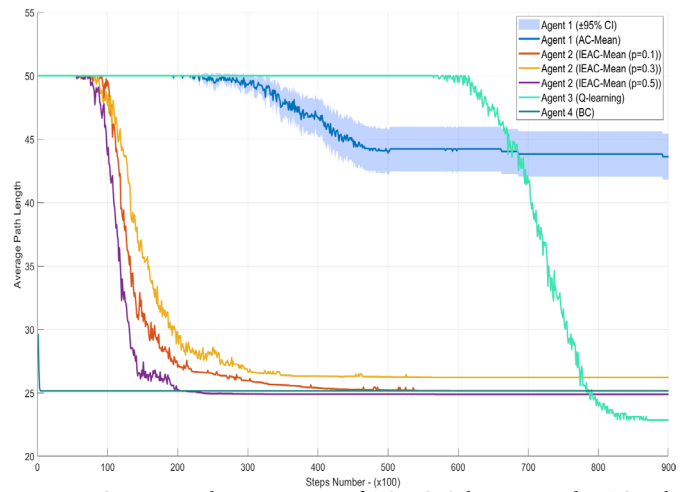


Figure 9. Comparison learning curves of AC, BC, Q-learning, and IEAC with different probabilities in the 20×20 obstacle maze

AC: Actor-critic, BC: Behavioral cloning, IEAC: Imitation enhanced actor-critic

For all probability values ($p \in \{0.1, 0.3, 0.5\}$), the IEAC method reached the optimum solution much faster than the AC method. Moreover, the minimal differences among imitation trials with different probabilities indicate that, in the obstacle-free environment, the imitation process provides a strong learning acceleration regardless of probability. In this environment, the learning curve of the AC agent exhibits a narrow 95% CI due to its relatively high learning rate, whereas the learning curves of the IEAC agents mostly remain outside this interval.

When examining the SD graph for the probability value $p=0.3$ in Figure 10, it is observed that the variance of the IEAC method narrows much earlier than that of the AC method for all probability values. This finding proves that social learning support not only increases speed but also reduces the instability (variance) of the agent in the exploration process, thus offering a more stable learning process. It should be noted that SD graphs are presented only for $p=0.3$ as a representative example, as similar variance patterns were observed across all probability values tested.

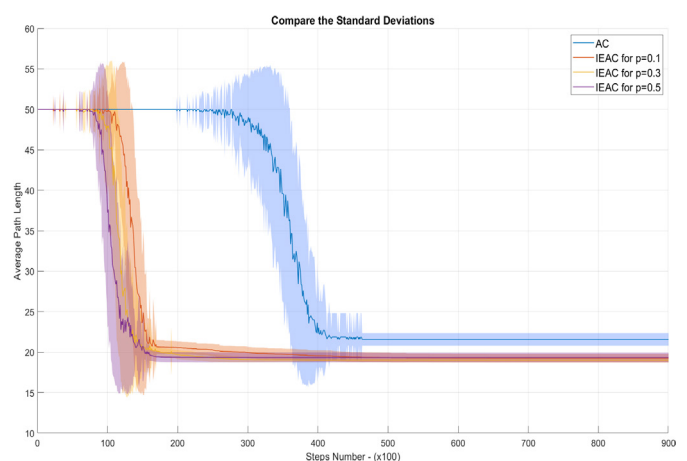


Figure 10. Standard deviation areas for all possibilities of IEAC in the 20×20 empty maze

IEAC: Imitation enhanced actor-critic

It is essential to present the mode imitation statistics, which quantify the extent to which the IEAC-trained agent imitates the leader over time. These results are critical as they demonstrate that the observer agent (agent 2) does not merely replicate the leader's actions exactly; rather, it utilizes

the imitation mechanism strategically to enhance its own learning process. The analysis illustrating the duration the agent remains in “copy mode” over time, for all probability values considered in this study, is provided in Figure 11. In this figure, the copy mode statistics for the learner agent were calculated by taking the average number of steps the agent remained in copy mode (i.e., imitated) over a rolling window of 100 steps.

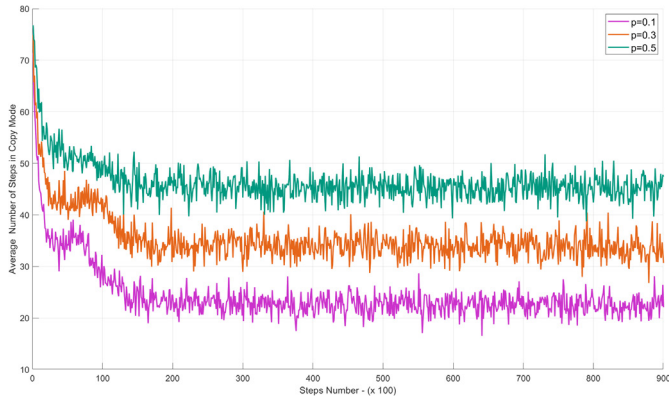


Figure 11. Copy mode statistics for all possibilities of IEAC in the 20x20 empty maze

IEAC: Imitation enhanced actor-critic

A close examination of the graph reveals a consistency between the imitation probability values and the duration spent in copy mode. Furthermore, it is observed that the imitation rate is initially high at the onset of training but exhibits a gradual downward trend as time progresses. This indicates that the learner agent, rather than merely replicating the leader exactly, utilizes the imitated actions to refine its own learning policy and reduces its reliance on imitation as its policy becomes optimized. However, it should be noted that the imitation rates never converge to zero. This persistence is inherent to the structure of the algorithm, which stochastically compels the agent to enter imitation mode at certain intervals based on the predefined probability values.

The results for the obstacle maze (Figure 9) show that the IEAC agent started learning earlier and converged much faster. The AC agent started learning after approximately 30,000 steps and got stuck in a local optimum around 40,000 steps with an average path length of approximately 44 steps. In contrast, the IEAC agent started learning significantly earlier (after approximately 10,000 steps for $p=0.3$) and reached the optimum solution around 25,000 steps with an average path length of 25 steps. Excessive imitation ($p=0.5$) slightly delays this process as it forces the agent to follow the leader's initially suboptimal path more frequently. Therefore, moderate imitation probabilities ($p=0.3$) provide a more effective balance.

Scalability analysis (50x50 environments): In addition to the 20x20 environments, the proposed models were tested in larger 50x50 maze configurations to evaluate scalability (Figure 12-17). The results in both empty and obstacle-based 50x50 mazes demonstrated consistent patterns: the IEAC method maintained its superiority in convergence speed and stability over the standard AC and Q-learning baselines, proving that the probabilistic imitation mechanism remains highly effective as the state space expands.

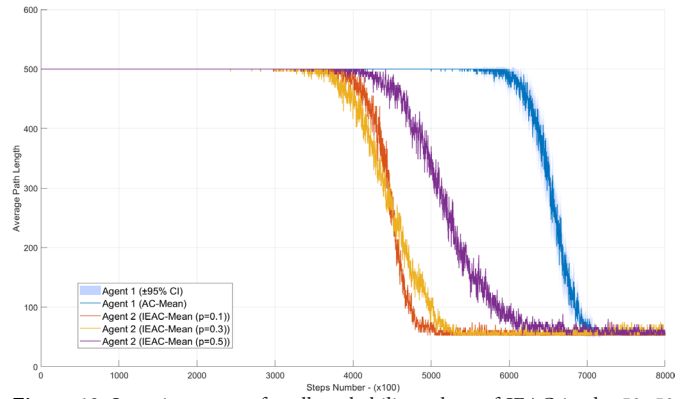


Figure 12. Learning curves for all probability values of IEAC in the 50x50 empty maze

IEAC: Imitation enhanced actor-critic

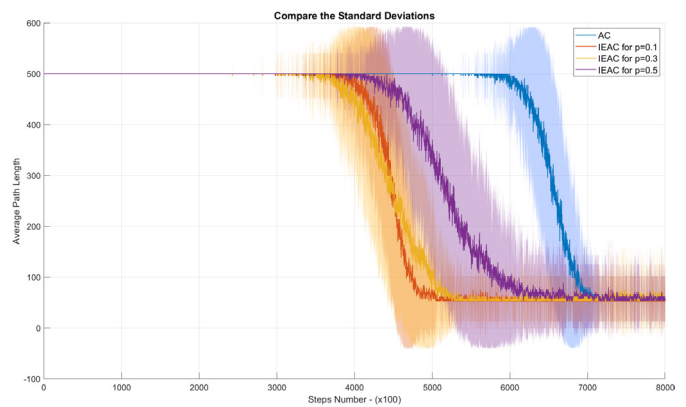


Figure 13. Standard deviation areas for all probability values of IEAC in 50x50 empty maze

IEAC: Imitation enhanced actor-critic

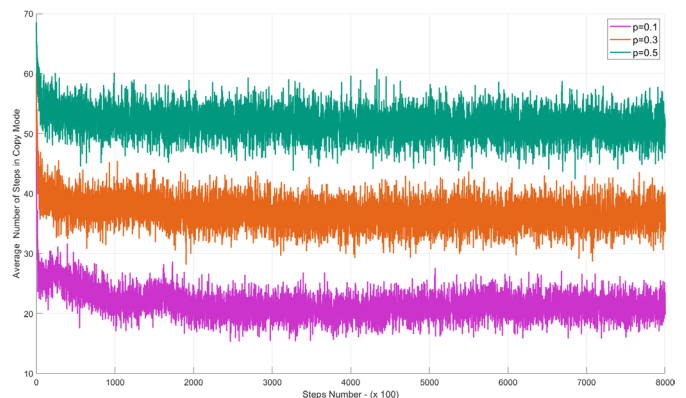


Figure 14. Copy mode statistics for all possibilities of IEAC in the 50x50 empty maze

IEAC: Imitation enhanced actor-critic

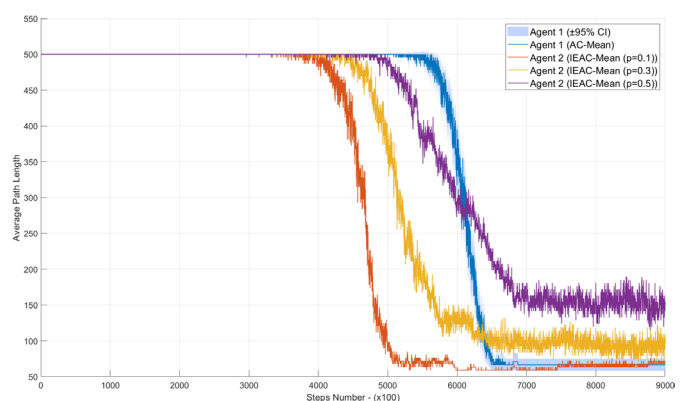


Figure 15. Learning curves for all probability values of IEAC in the 50x50 obstacle maze

IEAC: Imitation enhanced actor-critic

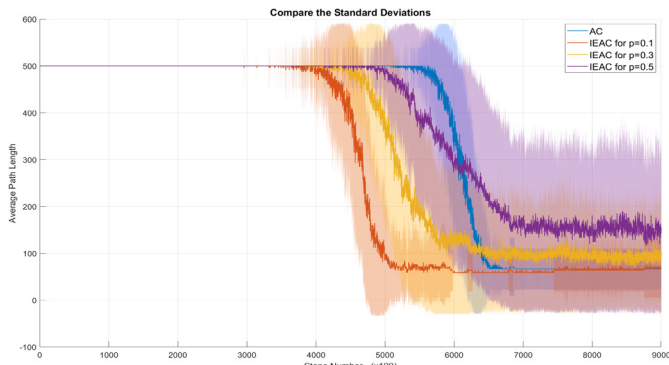


Figure 16. Standard deviation areas for all probability values of IEAC in 50x50 obstacle maze

IEAC: Imitation enhanced actor-critic

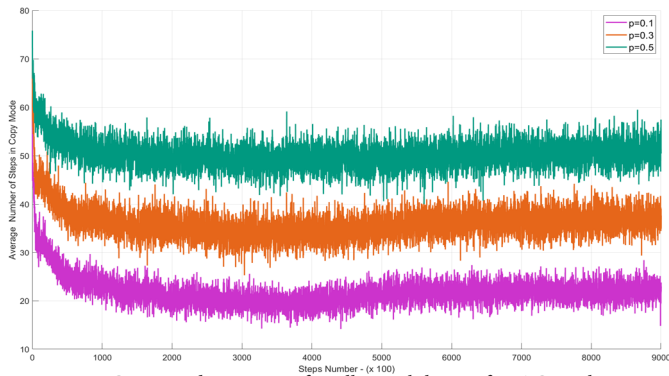


Figure 17. Copy mode statistics for all possibilities of IEAC in the 50x50 obstacle maze

IEAC: Imitation enhanced actor-critic

For each reporting point, the mean and SD were calculated over $n = 100$ independent runs. The 95% CI was calculated as “mean $\pm$ 1.96 $\cdot(\sigma/\sqrt{n})$ ”, where σ represents the sample SD across trials. Additionally, to compare the stability of the learning methods, the SD value was calculated in this study as $\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$.

Furthermore, we also analysed the stability of learning in a fixed obstacle maze environment by considering SDs for a probability value of $p=0.3$. As shown in Figure 18, the IEAC agent initially exhibited a wide range of SDs, and this range gradually narrowed over time. This indicated that imitation-based learning initially increased variance but became more stable as learning progressed. In contrast, the AC agent reflected a less stable learning process, maintaining higher variance over a longer period.

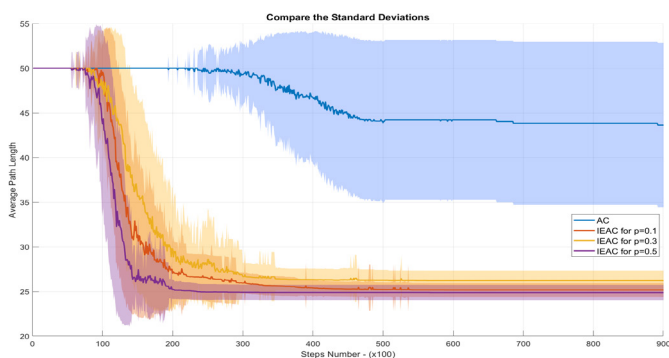


Figure 18. Standard deviation areas for all possibilities of IEAC in the 20x20 obstacle maze

IEAC: Imitation enhanced actor-critic

In summary, in an obstacle-based environment, the wide SD range of the AC agent indicates that the learning curve

oscillates, meaning that training consistency decreases. However, it is clear that the SD range of the IEAC agent is narrower than that of the AC method throughout the entire runtime. This suggests that in a fixed obstacle maze environment, the IEAC method is a more stable method than the AC algorithm.

Finally, following the key findings presented throughout the study, Figure 19 shows the statistics for all probability values, giving the average total number of imitation steps for agent 2 every 100 steps. Here, although the number of imitation steps was initially observed to be around 70 for all probability values, the average number of imitation steps decreased rapidly within the first 10,000 steps of the study (which corresponds to approximately the middle of the second trial), and then the number of imitation steps remained almost constant for each probability value. When Figure 18 and 19 are considered together, it is clear that the results are consistent.

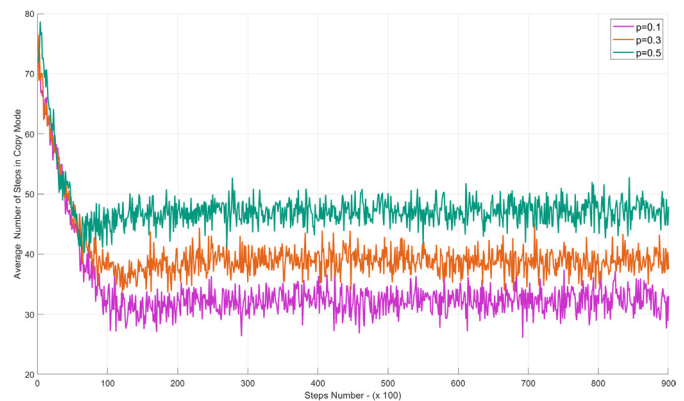


Figure 19. Copy mode statistics for all possibilities of IEAC in the 20x20 obstacle maze

IEAC: Imitation enhanced actor-critic

DISCUSSION

In this study, the IEAC method is based on observing the positive actions of the first agent that led to the goal, without direct information exchange between agents, and then having the second agent imitate these actions under specific conditions. Thus, the learning-by-imitation method, combined with the AC method, is compared with the standard AC algorithm in terms of learning speed, stability, and success in reaching the optimum result.

In an obstacle-free maze environment, although the probabilistic imitation method and the IEAC method are more successful in early learning, it is observed that both AC and IEAC reach the optimum result in the long term. In a fixed obstacle environment, it is clear that the IEAC method provides much better performance than the simple AC method. While the standard AC agent gets stuck because it cannot risk the immediate negative rewards necessary to circumvent the obstacle, the IEAC agent overcomes this problem because it performs the imitation process probabilistically, and comes very close to the optimum solution in reaching the goal. Similar to human behaviour models in the literature, in situations with high exploration costs, the social learning (imitation) mechanism provides a safe learning process by reducing the cost of trial and error. These findings show that in dynamic environments with high uncertainty, choosing an action (through imitation), even if suboptimal, instead of remaining inactive or experiencing indecision, accelerates the learning process.

The literature review revealed that learning through imitation is applied in conjunction with some RL methods. Therefore, it was predicted that positive results could be obtained by combining the AC RL method with the imitation method. To provide insight into whether the proposed IEAC method is useful in learning, the fundamental problem of “finding the shortest path in a maze” was considered. In this regard, experiments were conducted in various 20×20 maze environments, and the learning speed and stability in reaching the optimum result of the method were compared only in a computer environment.

The obstacle maze scenario also highlights a structural limitation of the IEAC model: if the leader agent gets trapped in a local optimum, the follower agent, especially at high imitation probabilities, tends to reproduce this suboptimal strategy. This presents a significant development opportunity for future studies. Instead of a fixed imitation probability (p), developing adaptive mechanisms where p is dynamically adjusted based on TD-error fluctuations, success rates, or stagnation detection would enable the follower to break free from the leader's mistakes and discover its own optimal policy.

Limitations

The experimental results of this study were simulated in a 2D environment with only two agents. Future research could validate the real-world applicability of the IEAC approach by applying it to physical robots and scaling it to multi-agent systems. Furthermore, the study could be extended to more complex and continuous-field problems: instead of using fixed probability values in the imitation method, more systematic methods that can make and implement their own decisions regarding imitation could be developed, or a method created by combining the AC method using deep neural networks with social learning could shed light on the solution of many problems.

CONCLUSION

As a result, the IEAC method achieves faster and more stable learning than the classical AC approach. Its earlier convergence to the optimal solution, particularly in the obstacle environment, confirms the effectiveness of imitation-based RL in complex settings. The proposed framework is particularly suitable for environments where exploration is costly or risky, making it a promising candidate for real-world autonomous navigation problems.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: NK, MDE; Design: NK, MDE; Control: NK, MDE; Resources: NK, MDE; Materials: NK, MDE; Data Collection and/or Processing: NK, MDE; Analysis and/or Interpretation: NK, MDE; Literature Review: NK, MDE; Writing the Article: NK, MDE; Critical Review: NK, MDE.

REFERENCES

- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469-483. <https://doi.org/10.1016/j.robot.2008.10.024>
- Bentivegna, D., Cheng, G., & Atkeson, C. (2005). Learning from observation and from practice using behavioral primitives. *Springer Tracts in Advanced Robotics*, 15, 551-560. https://doi.org/10.1007/11008941_59
- Block, A., Foster, D. J., Krishnamurthy, A., Simchowitz, M., & Zhang, C. (2024). Butterfly effects of SGD noise: Error amplification in behavior cloning and autoregression. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Brederche, N., & Fontbonne, N. (2022). Social learning in swarm robotics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 377, 1843. <https://doi.org/10.1098/rstb.2020.0309>
- Christianos, F., Schäfer, L., & Albrecht, S. V. (2021). Shared experience actor-critic for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 10707-10717. <http://arxiv.org/abs/2006.07169>
- Codevilla, F., Müller, M., López, A., Koltun, V. & Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE Press, 1-9. <https://doi.org/10.1109/ICRA.2018.8460487>
- Erbas, M. D., Winfield, A. F., & Bull, L. (2014). Embodied imitation-enhanced reinforcement learning in multi-agent systems. *Adaptive Behavior*, 22(1), 31-50. <https://doi.org/10.1177/1059712313500503>
- Guenther, F., & Billard, A. G. (2007). Using reinforcement learning to adapt an imitation task. *IEEE International Conference on Intelligent Robots and Systems*, 1022-1027. <https://doi.org/10.1109/IROS.2007.4399449>
- Guo, X., Chang, S., Yu, M., Tesauro, G., & Campbell, M. (2019). Hybrid Reinforcement Learning with Expert State Sequences. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 3739-3746. <https://doi.org/10.1609/aaai.v33i01.33013739>
- Henrich, J. (2016). *The Secret of Our Success: How Culture is Driving Human Evolution, Domesticating Our Species, and Making Us Smarter*. Princeton University Press. <https://doi.org/10.2307/j.ctvc77f0d>
- Mathewson, K. W., & Pilarski, P. M. (2017). Actor-critic reinforcement learning with simultaneous human control and feedback. *arXiv*. <http://arxiv.org/abs/1703.01274>
- Ndousse, K., Eck, D., Levine, S., & Jaques, N. (2021). Emergent social learning via multi-agent reinforcement learning. *arXiv*. <https://doi.org/10.48550/arXiv.2010.00581>
- Nguyen, S. M. (2024). The intrinsic motivation of reinforcement and imitation learning for sequential tasks. *arXiv*. <https://doi.org/10.48550/arXiv.2412.20573>
- Rao, R. P. N., Shon, A. P., & Meltzoff, A. N. (2007). A Bayesian model of imitation in infants and robots. In C. L. Nehaniv & K. Dautenhahn (Eds), *Imitation and Social Learning in Robots, Humans and Animals* (1st ed., pp. 217-248). Cambridge University Press. <https://doi.org/10.1017/CBO9780511489808.016>
- Sallab, A. El, Abdou, M., Perot, E., & Yogamani, S. (2017). Deep Reinforcement Learning framework for Autonomous Driving. *Proceedings of the IS&T International Symposium on Electronic Imaging Science and Technology*, 70-76. <https://doi.org/10.2352/issn.2470-1173.2017.19.avm-023>
- Sharma, A., Kaur, H., & Prashar, D. (2023). Maze Solver: A Federated Reinforcement Learning Approach. 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp.1-5. <https://doi.org/10.1109/ICCCNT56998.2023.10307671>
- Sutton, R. S., & Barto, A. G. (2014). *Reinforcement learning: an introduction* (2nd ed.). MIT Press.
- Uwano, F., Tatebe, N., Tajima, Y., Nakata, M., Kovacs, T., & Takadama, K. (2018). Multi-agent cooperation based on reinforcement learning with internal reward in maze problem. *Journal of Control, Measurement, and System Integration*, 11(4), 321-330. <https://doi.org/10.9746/jcmsi.11.321>
- Yang, X., Gao, J., Wang, P., Long, W., & Fu, C. (2024). Social Learning with Actor-Critic for dynamic grasping of underwater robots via digital twins. *Ocean Engineering*, 306. <https://doi.org/10.1016/j.oceaneng.2024.118070>

Android application for multilingual hate speech detection: using XLM-RoBERTa and ONNX optimized

 Salam Thabet Doghmash^{*1},  İbrahim Alper Doğru²,  Kazım Kılıç³

¹Department of Computer Sciences, Institute of Informatics, Gazi University, Ankara, Türkiye

²Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

³Department of Artificial Intelligence and Data Engineering, Faculty of Engineering and Natural Sciences, Samsun University, Samsun, Türkiye

Cite this article as: Doghmash, S. T., Doğru, İ. A., & Kılıç, K. (2026). Android application for multilingual hate speech detection: using XLM-RoBERTa and ONNX optimized. *J Comp Electr Electron Eng Sci*, 4(2), 72-76. <https://doi.org/10.51271/JCEES-0039>

Received: 01.02.2026

Accepted: 07.05.2026

Published: 13.10.2026

ABSTRACT

Aims: Social media and digital platforms face increasing challenges in dealing with multilingual abusive content, which calls for the development of effective solutions for the automatic detection of hate speech. This research presents a practical system for monitoring hate speech in three languages (Arabic, Turkish, and English) based on the improved XLM-RoBERTa model, while exploring ensemble techniques as a baseline for comparison.

Methods: FrancophonIA data, consisting of 219,508 texts across the three languages, was used after processing. Baseline models were developed combining logistic regression, support vector machines, and random forest using soft voting with TF-IDF feature extraction. The main model was based on XLM-RoBERTa with graphics processing unit (GPU)-accelerated training and an early stopping strategy to improve efficiency. The final models were converted to ONNX for compatibility, and a mobile application was created for Android.

Results: The baseline model achieved a macro F1-score of 0.7796, outperforming the individual classifiers. In contrast, the optimized XLM-RoBERTa achieved a substantial improvement of 84.93% accuracy and a macro F1-score of 0.8493 across different languages. Optimization via open neural network exchange (ONNX) enabled efficient real-time inference in both the API and the mobile application.

Conclusion: The research proves the feasibility of employing transformer-based multilingual models in building practical content monitoring systems, with the potential for actual deployment on mobile devices to achieve real-time monitoring of hate speech in real-world environments.

Keywords: Multilingual hate speech detection, XLM-RoBERTa, ensemble baseline, ONNX, Android deployment

INTRODUCTION

Globally, the way people communicate and share information has changed dramatically as a result of the digital communication platforms' explosive rise. But this growth has also accelerated the dissemination of damaging content, such as hate and offensive speech, which has a detrimental impact on people's well-being and social cohesiveness (Vogels, 2021). A Pew Research Center survey found that 41% of American adults have personally been the victim of internet harassment, with 25% reporting more severe or protracted instances. Furthermore, recent reports from the EU highlight substantial challenges in content moderation, particularly the ineffectiveness of automated systems and human reviewers in detecting nuanced hateful speech across languages and dialects (Online Content Moderation-Current Challenges in Detecting Hate Speech | European Union Agency for Fundamental Rights, 2023).

Despite increasing attention, current hate speech detection approaches face several limitations. Studies show that approximately 85% of existing systems are designed

exclusively for English, with less than 15% addressing multilingual contexts, leaving a considerable gap for widely spoken languages such as Arabic and Turkish (Mnassri et al., 2024). Moreover, many models remain computationally expensive, resource-intensive, and difficult to deploy in real-world scenarios, particularly on mobile devices where content is frequently generated and consumed (Mahajan et al., 2024).

New developments in natural language processing (NLP) have shown promising cross-lingual performance, especially transformer-based architectures like XLM-RoBERTa. However, practical integration of these models into deployable, real-time applications remains limited due to infrastructure complexity and high resource demands (Usman et al., 2025). A recent systematic review by (Ahmad et al., 2024) compared over one hundred studies concluded that multilingual transformer models consistently outperform traditional methods, though they raise challenges in terms of computational cost and fairness across languages. (Conneau et al., 2020) expanded on multilingual capabilities by introducing XLM-R, showing

*Corresponding Author: Salam Thabet Doghmash, salam.th.d@gmail.com



that zero-shot cross-lingual transfer is much enhanced by scaling multilingual models with a variety of language data. For morphologically rich languages like Arabic and Turkish, (Rust et al., 2021) investigated language-specific fine-tuning techniques in more detail, addressing dialectal variances and code-switching that are common in social media contexts. For under-resourced languages, recent work has emphasized Arabic and Turkish (Ahmad et al., 2024) investigated Arabic hate speech detection, noting the difficulties posed by dialectal variation and informal writing styles, and highlighted the need for tailored preprocessing.

Addressing these challenges requires solutions that not only achieve high detection accuracy but also ensure accessibility, scalability, and efficiency for end-users. This paper presents a complete methodology for detecting hate speech in Arabic, Turkish, and English. The system combines an improved transformer model, XLM-RoBERTa, that has been fine-tuned on GPU with balanced sampling, with the traditional ensemble baselines of Random Forest, Support Vector Machines, and Logistic Regression coupled through soft-voting. In addition to performance, the models were deployed in a practical environment that included an enhanced open neural network exchange (ONNX) API for real-time inference and an Android mobile app for easy access by end users. This combination of improved accuracy and scalability demonstrates the feasibility of translating advanced multilingual NLP research into practical content management solutions.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Study Design

This study was designed as an experimental analysis concentrating on building a system for detecting hate speech in multiple languages. Our methodology covered six stages: dataset preparation, preprocessing, baseline construction, transformer-based modeling, evaluation, and deployment as shown in **Figure 1**.

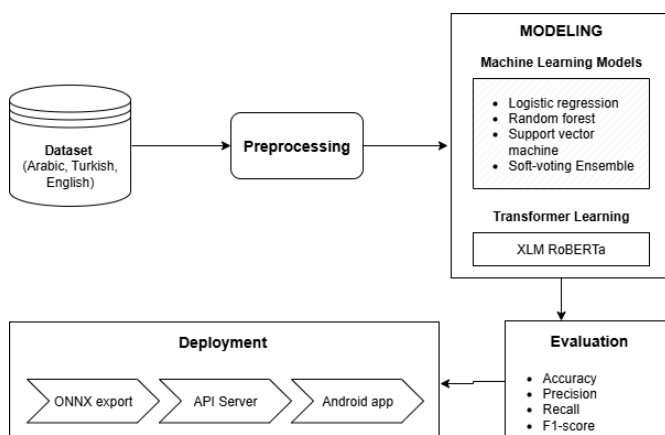


Figure 1. The methodology steps

Dataset: The FrancophonIA multilingual hate speech dataset was used, this dataset originally contained approximately 219,508 training samples in Arabic, English, and Turkish. The dataset shows a moderate imbalance in classes, with

hate speech comprising 38.5% (84,418 samples) and non-hate speech 61.5% (135,090 samples). To address this imbalance, under sampling was applied, randomly selecting 84,418 samples from the larger group. This resulted in a balanced dataset of 168,836 samples, with each group representing 50%. Subsequently, stratified sampling was used to divide the balanced dataset into three groups: a training set (138,122 samples), a validation set (15,349 samples), and a test set (15,365 samples). The 50% representation of each group was maintained across all groups. The final dataset distribution is summarized in **Table 1**. In addition, the dataset provides formal test sets dedicated to each language, used to assess performance separately for each language, including: Arabic (463 samples), Turkish (5,397 samples), and English (5,906 samples), as shown in **Table 2**.

Table 1. The dataset distribution

Subset	Non-hate samples	Hate samples	Total
Training	69,061	69,061	138,122
Validation	7,674	7,675	15,349
Test	7,683	7,682	15,365

Table 2. Official language test sets

Language	Total
Arabic	463
Turkish	5,397
English	5,906

Preprocessing: Preprocessing was applied consistently across the three languages using a five-stage pipeline. In the first stage, null records were identified and deleted, reducing the dataset from 219,981 to 219,508 samples (473 samples deleted). In the second stage, all text was converted to string format, removing extra spaces at the beginning and end of the text. In the third stage, URLs were removed using regular expressions. In the fourth stage, spacing was standardized by combining spaces and consecutive lines into a single space. In the fifth stage, character-level filtering was applied to retain only Unicode ranges associated with the target languages. These ranges include Arabic characters (U+0600-U+06FF), expanded Latin characters covering Turkish characters such as ç, ğ, ı, ö, ş, and ü (U+00C0-U+024F), and standard English letters and numbers, while all other special characters and symbols were removed. The average token length remained relatively constant before and after processing, at 18.66 tokens before processing and 18.97 tokens after processing per sample, with a maximum of 2,909 tokens and a minimum of 1 token. Finally, the labels were standardized into a binary format, so that the value 0 represents texts not classified as hate speech, and the value 1 represents hate speech, resulting in a data distribution of 135,090 non-hate samples (61.54%) and 84,418 hate samples (38.46%).

Baseline models: Classical machine learning models were implemented as a benchmark. TF-IDF vectorization was used to represent text, followed by classification with logistic regression, support vector machines (SVM), and random forest. Their outputs were combined through a soft-voting ensemble. The ensemble designed was primarily as a baseline for comparison with the transformer approach.

Transformer models: For the main structure, the XLM-RoBERTa transformer was fine-tuned on the balanced dataset. Training was conducted on an NVIDIA A100 GPU with optimized hyperparameters to prevent overfitting while maintaining generalization capability. **Table 3** summarizes the key training configurations and deployment specifications. The trained model was exported in ONNX format, ensuring portability and efficient inference across various deployment environments, including resource-constrained mobile devices.

Table 3. The key training configurations and deployment specifications

Parameter	Value/description
Model	XLM-RoBERTa-base
GPU	NVIDIA A100
Learning rate	1e-5
Max_length	128
Batch size	16
Epochs	8
Scheduler	linear_warmup_decay
Weight_decay	0.01
Early_stopping	patience=3 on val_loss
Random_seed	42
Num_labels	2
Dropout	0.1
Optimization	AdamW optimizer
Deployment target	Cross-platform mobile applications

XLM-RoBERTa: Cross-lingual language model-robustly optimized BERT approach, BERT: Bidirectional encoder representations from transformers, GPU: Graphics processing unit

Deployment: To ensure real-world applicability, deployment was emphasized as a central contribution. The ONNX-optimized transformer model enabled efficient inference with reduced latency, achieving real-time predictions in 22 ms on CPU. A lightweight API was developed for integration into applications, and an Android mobile app was built in Kotlin with a modern interface, confidence visualization, and network status indicators. This dual deployment API for scalability and Android for mobile accessibility demonstrates the feasibility of bridging state-of-the-art multilingual NLP research with practical solutions for multilingual content moderation.

RESULTS

The experimental evaluation was performed on the balanced internal test set of 15,365 samples. **Table 4** summarizes the results. The ensemble baseline achieved a macro F1-score of 0.7796, slightly outperforming individual classifiers. The multilingual BERT baseline (mBERT) achieved an F1-score of 0.8356 and accuracy of 83.57%. These results demonstrate the advantage of contextual embeddings in capturing multilingual hate speech semantics compared to TF-IDF-based approaches. The fine-tuned XLM-RoBERTa model achieved the highest performance with an F1-score of 0.8493 and accuracy of 84.93%, confirming its superiority for multilingual hate speech detection. Early stopping was triggered at epoch 6 for mBERT, while XLM-RoBERTa continued to improve through the full training cycle.

Table 4. The comparative performance of baseline models and the transformer model

Model	F1 macro	Accuracy	Precision	Recall
Ensemble baseline (LR+SVC+RF)	0.7796	0.7797	0.7801	0.7797
mBERT	0.8356	0.8357	0.8367	0.8357
XLM-RoBERTa (ours)	0.8493	0.8493	0.8493	0.8493

mBERT: Multilingual BERT, XLM-RoBERTa: Cross-lingual language model-robustly optimized BERT approach, BERT: Bidirectional encoder representations from transformers

To provide a more detailed evaluation, the transformer model was tested separately on the language-specific test sets provided by the FrancophonIA dataset, which includes Arabic (463 samples), Turkish (5,397 samples), and English (5,906 samples), totaling 11,766 samples. **Table 5** presents the results for each language individually. In English, the model achieved an F1 score of 0.9395 and an accuracy of 93.97%, reflecting strong performance in the language with the most readily available data. In Turkish, the F1 score was 0.8205 and the accuracy was 89.12%. Arabic performed relatively less well, with an F1 score of 0.7628 and an accuracy of 76.89%, attributed to the small size of the test group (463 samples) and the complexity associated with the diversity of Arabic dialects.

Table 5. Performance of the fine-tuned XLM-RoBERTa across languages

Language	Precision	Recall	F1 macro	Accuracy
Arabic	0.7612	0.7761	0.7628	0.7689
Turkish	0.8389	0.8055	0.8205	0.8912
English	0.9388	0.9405	0.9395	0.9397

XLM-RoBERTa: Cross-lingual language model-robustly optimized BERT approach, BERT: Bidirectional encoder representations from transformers

Confusion matrices for each language are shown in **Figure 2**, demonstrating that the model maintains a high capacity for recognizing non-hate speech texts across all languages, although its sensitivity to hate speech varies, particularly in Arabic and Turkish.

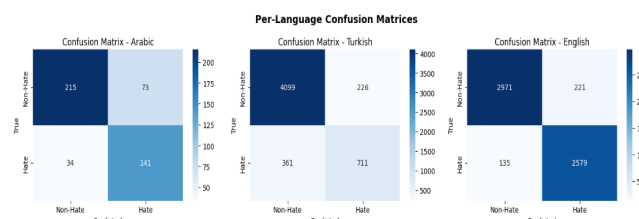


Figure 2. The confusion matrices for per-language using XLM-RoBERTa model

XLM-RoBERTa: Cross-lingual language model-robustly optimized BERT approach, BERT: Bidirectional encoder representations from transformers

In addition to the quantitative evaluation, the deployment aspect was validated through real-world testing. The fine-tuned XLM-RoBERTa model achieved an average inference latency of 9.26 ms per sample (± 7.14 ms) on GPU using PyTorch. Following ONNX export, the model was served through a lightweight API server, achieving real-time inference with response times below one second. The ONNX exported model has a file size of approximately 1.061 MB, comparable to the PyTorch model, while enabling CPU-only inference without requiring a GPU. Inference latency measurements, as shown in **Table 6**, revealed that the PyTorch model achieved a latency of 2.62 ms per sample when running on the A100 GPU, while the ONNX model achieved a latency of 22.41 ms per sample

when running on the CPU without requiring GPU input. Despite the different operating environments, the ONNX model maintains real-time inference capability (less than 25 ms), highlighting its suitability for applications in resource-constrained environments such as mobile devices. To further demonstrate usability, an Android mobile application was developed with a simple interface for multilingual text input and real-time classification. **Figure 3** shows sample screenshots of the application, where the system successfully identifies hate speech across English, Turkish, and Arabic examples while displaying confidence scores.

Table 6. PyTorch vs ONNX: latency and model size comparison

Format	Hardware	Avg latency	Model size
PyTorch	GPU (A100)	2.62 ms	1,100 MB
ONNX	CPU	22.41 ms	1,061 MB

ONNX: Open neural network exchange, GPU: Graphics processing unit, CPU: Central processing unit, Avg: Average, ms: Milliseconds, MB: Megabytes

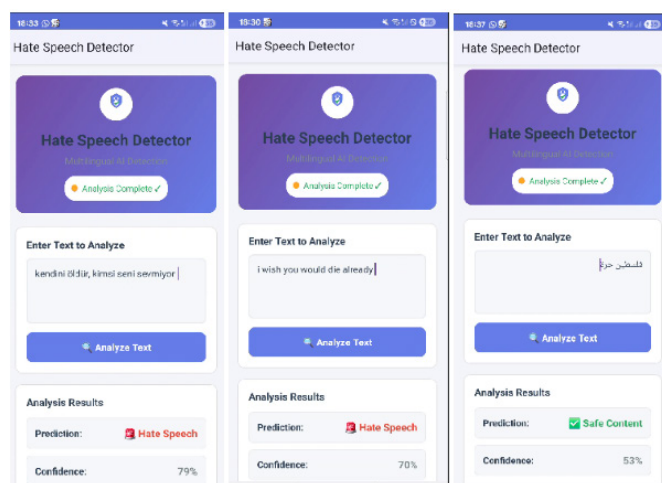


Figure 3. Sample screenshots from the application

Together, these results confirm that while ensemble baselines provide a reasonable benchmark, the transformer-based solution offers superior accuracy and practical deployability in both cloud and mobile environments.

DISCUSSION

The experimental results highlight the superiority of transformer-based models compared to classical machine learning approaches. While the ensemble baseline achieved moderate performance with a macro F1-score of 0.7796, the fine-tuned XLM-RoBERTa model substantially improved the results, achieving a macro F1-score of 0.8493 and an accuracy of 84.93%. This highlights how accurately contextual embeddings capture the complexity of multilingual hate speech. For comparison, the mBERT model achieved an overall F1 score of 0.8356, confirming that the broader multilingual pre-training of the XLM-RoBERTa model provides a significant advantage over the standard multilingual BERT model.

The evaluation for each language revealed a marked variation in performance among the three target languages. English achieved the best results with an F1 score of 0.9395 and an accuracy of 93.97%, while Turkish showed average performance F1=0.8205, accuracy 89.12%. Arabic showed relatively lower results F1=0.7628, accuracy=76.89%, which

can be attributed to two factors: the small size of the formal Arabic language test group (463 samples) and the inherent complexity of the diversity of Arabic dialects, for which the model was not specifically trained.

The strength of the study lies in its coverage of three languages (Arabic, Turkish, and English) within a unified model, with a significant reduction in false negatives thanks to data balancing and preprocessing. Furthermore, the successful ONNX export enabled real-time inference of 22.41 ms per sample on the CPU. This confirms its practical deployment on resource constrained devices without relying on the GPU. The Android application also underscores the system's overall feasibility for managing multilingual content in the real world.

Limitations

However, some limitations remain. The dataset does not adequately reflect the diversity of Arabic dialects, as Egyptian, Levantine, and Gulf dialects differ significantly from the patterns observed in the training data. Furthermore, the system is limited to text only and does not support multimedia speech that combines images or audio with text, which is increasingly common on social media platforms. Social media language also evolves over time, meaning the model's performance may decline without periodic retraining.

Overall, the results demonstrate that transformer-based architectures constitute a robust and deployable solution for multilingual hate speech detection, providing a solid foundation for future developments in practical content moderation systems.

CONCLUSION

This study presents a multilingual Android-based system for detecting hate speech, covering Arabic, Turkish, and English. Based on experimental results, the modified XLM-RoBERTa transformer performed significantly better than traditional baselines, achieving an overall F1-score of 0.8493 and an accuracy of 84.93%, while the baseline achieved an overall F1-score of 0.7796. These results confirm the effectiveness of contextual embeddings in dealing with multilingual hate speech detection compared to traditional feature-based methods. In addition to the model's performance, the system was successfully deployed with ONNX export and Android app, ensuring instant inference with lightweight deployment. This highlights the feasibility of linking advanced natural language processing research with deployable solutions for online content management.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: STD, İAD, KK; Design: STD, İAD, KK; Control: STD, İAD, KK; Resources: STD, İAD, KK; Materials: STD, İAD, KK; Data Collection and/or Processing: STD, İAD, KK; Analysis and/or Interpretation: STD, İAD, KK; Literature Review: STD, İAD, KK; Writing the Article: STD, İAD, KK; Critical Review: STD, İAD, KK.

REFERENCES

- Ahmad, A., Azzeh, M., Alnagi, E., Abu Al-Haija, Q., Halabi, D., ..., & AbuHour, Y. (2024). Hate speech detection in the Arabic language: corpus design, construction, and evaluation. *Frontiers in Artificial Intelligence*, 7, 1345445. <https://doi.org/10.3389/frai.2024.1345445>
- Barkhordar, E., Topçu, I. S., & Hürriyetoglu, A. (2024). Team Curie at HSD-2Lang 2024: hate speech detection in Turkish and Arabic tweets using BERT-based models. In A. Hürriyetoglu, H. Tanev, S. Thapa, & G. Uludoğan (Eds.), *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)* (pp. 215-220). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.case-1.30>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ..., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schuster, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440-8451). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.747>
- European Union Agency for Fundamental Rights. (2023). Online content moderation: current challenges in detecting hate speech. Publications Office of the European Union. <https://doi.org/10.2811/923316>
- Mahajan, E., Mahajan, H., & Kumar, S. (2024). EnsMulHateCyb: multilingual hate speech and cyberbully detection in online social media. *Expert Systems with Applications*, 236, 121228. <https://doi.org/10.1016/j.eswa.2023.121228>
- Mnassri, K., Farahbakhsh, R., Chalehchaleh, R., Rajapaksha, P., Jafari, A. R., ..., & Crespi, N. (2024). A survey on multi-lingual offensive language detection. *PeerJ Computer Science*, 10, e1934. <https://doi.org/10.7717/peerj-cs.1934>
- Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., & Gurevych, I. (2021). How good is your tokenizer? On the monolingual performance of multilingual language models. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 3118-3135). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.acl-long.243>
- Usman, M., Ahmad, M., Tash, M. S., Gelbukh, I., Tellez, R. Q., & Sidorov, G. (2025). Multilingual hate speech detection in social media using translation-based approaches with large language models. arXiv. <https://doi.org/10.48550/arXiv.2506.08147>
- Vogels, E. A. (2021, January 13). The state of online harassment. Pew Research Center. <https://www.pewresearch.org/internet/2021/01/13/the-state-of-online-harassment/>

Explainable brain tumor classification using an attention-augmented CNN and Grad-CAM

 Maral Amer Mustafa^{*1,2},  Osman Ayhan Erdem³,  Esra Söğüt³

¹Department of Computer Engineering, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Türkiye

²Department of Production Mechanics Techniques, Kirkuk Technical Institute, Northern Technical University, Kirkuk, Iraq

³Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye

Cite this article as: Mustafa, M. A., Erdem, O. A., & Söğüt, E. (2026). Explainable brain tumor classification using an attention-augmented CNN and Grad-CAM. *J Comp Electr Electron Eng Sci*, 4(2), 77-82. <https://doi.org/10.51271/JCEES-0040>

Received: 12.01.2026

Accepted: 21.05.2026

Published: 13.10.2026

ABSTRACT

Aims: This paper explores clinical explainability of deep neural network architectures for classifying brain neoplasms using magnetic resonance imaging (MRI) scans.

Methods: We evaluated multiple architectures, including a baseline convolutional neural network (CNN), a novel attention-augmented CNN, and pre-trained transfer learning models (MobileNetV2, InceptionV3, and Xception). Models were trained using the Adam optimizer on a quaternary-class MRI dataset comprising glioma, meningioma, pituitary adenoma, and healthy cases. Performance was assessed using quantitative metrics (accuracy, precision, sensitivity, and F1-score), while interpretability was evaluated via gradient-weighted class activation mapping (Grad-CAM) visualizations.

Results: MobileNetV2 demonstrated superior performance, achieving a classification accuracy of 91.7%, with highly consistent precision, sensitivity, and F1-scores. The attention-augmented CNN achieved competitive efficacy, while Grad-CAM heatmaps consistently localized pathologically significant regions, thereby confirming the diagnostic reliability and transparency of the models' decision-making processes.

Conclusion: Computationally efficient pre-trained architectures like MobileNetV2 are highly effective for MRI-based neoplasm classification. Furthermore, integrating attention mechanisms with Grad-CAM visualization significantly enhances both diagnostic accuracy and model transparency, highlighting their strong potential for adoption in real-world clinical decision support systems.

Keywords: MRI, deep learning, brain tumor, transfer learning, explainable AI

INTRODUCTION

Deep neural network (DNN) models have proven highly effective for brain tumor classification using magnetic resonance imaging (MRI) and have substantially advanced diagnostic imaging research (Ebiele et al., 2020). Despite these improvements, their clinical adoption remains limited because of persistent concerns regarding model precision, reliability, and transparency. Deep learning systems are often criticized for their inherent opacity, which is commonly described as a black-box problem. This lack of transparency reduces clinicians' confidence in algorithmic predictions and underscores the need for frameworks that offer not only strong predictive performance but also interpretable diagnostic explanations (Brima et al., 2021).

Explainable Artificial Intelligence (XAI) methods are model-agnostic interpretability frameworks designed to provide localized explanations for the predictions of complex machine learning models (Atagün et al., 2026). Recent studies have further emphasized that XAI has become a critical requirement for the reliable adoption of deep learning in medical imaging, particularly in clinically sensitive tasks. Current literature

highlights that transparency in medical image analysis is necessary not only to improve clinician trust, but also to support accountability, interpretability, and safer decision-making. Studies have also shown that explainability in medical imaging is not limited to saliency-based visualizations, but may also include textual justifications, and comparative evaluation of explanation quality across methods. In addition, different XAI techniques may vary substantially with some methods providing stronger confidence increase and more clinically meaningful localization of abnormal regions than others. These findings indicate that explainability should be considered a central design objective in AI-based medical imaging systems rather than an optional post hoc add-on (Chaddad et al., 2025; Houssein et al., 2025; Haupt et al., 2025).

Recent studies on brain tumor MRI analysis have increasingly emphasized the joint use of high-performing deep learning architectures and XAI techniques to improve both predictive accuracy and clinical transparency. Recent multiclass classification studies have integrated hybrid deep learning frameworks with Grad-CAM, LIME, and SHAP to

*Corresponding Author: Maral Amer Mustafa, 24833304401@gazi.edu.tr



generate interpretable explanations while achieving strong performance on public brain MRI datasets. Other works have proposed unified diagnostic pipelines combining classification, preprocessing enhancement, and XAI-based visual interpretation, showing that explainability can support the clinical usability of accurate MRI-based tumor detection systems. In addition, explainable deep learning models have also been extended to tumor localization and segmentation tasks, where Grad-CAM and backpropagation-based visualizations were used to improve transparency and support expert interpretation. Collectively, these studies indicate that current brain tumor MRI research increasingly prioritizes not only classification success, but also interpretable and clinically meaningful model behavior (Hasan et al., 2026; Lamrani et al., 2026; Kumar et al., 2026).

Different pre-trained deep learning models have been tested for brain tumor MRI classification using gradient-based attribution methods such as gradient-weighted class activation mapping++ (Grad-CAM++) (Chattopadhyay et al., 2018), Grad-CAM (Selvaraju et al., 2017), and Integrated Gradients (Sundararajan et al., 2017). The Vanilla Gradient method, which relies on gradient propagation for attribution visualization, is susceptible to signal noise and gradient vanishing (Zeineldin et al., 2022). Comparative analyses in NeuroXAI-related studies have shown that Vanilla Gradient often produces noisy and less discriminative saliency maps than alternative techniques (Martí-Juan et al., 2022; Philbrick et al., 2018; Zeineldin et al., 2022). In vascular segmentation, the TorchEsegeta evaluation framework reported that Vanilla Gradient and DeconvNET produced more understandable attributions than DeepLIFT and Grad-CAM++ (Chatterjee et al., 2022). Guided Backpropagation variants were later introduced to reduce signal noise, although sometimes at the cost of oversimplifying feature representations (Wood et al., 2022). layer-wise relevance propagation (LRP) addresses gradient saturation by propagating relevance scores instead of raw gradients, thereby offering an alternative explanation strategy (Saleem et al., 2021). LRP has also shown value in generating class-discriminative activations in computed tomography-based aneurysm identification studies (Montavon et al., 2019).

Recent studies have also explored hybrid and inherently interpretable architectures for brain MRI analysis in order to improve both predictive performance and clinical usability. For example, hybrid frameworks have combined deep feature extraction with machine learning and XAI tools such as Grad-CAM, Grad-CAM++, and LIME to support multiclass brain tumor classification with more transparent decision-making. Other recent work has proposed interpretable MRI pipelines that integrate attention mechanisms, transformer-based components, and retrieval-oriented designs. They improve segmentation, classification, and clinical decision support under heterogeneous imaging conditions. In addition, recent efficient classification architectures have shown that high diagnostic accuracy can be pursued together with built-in interpretability. And these studies suggesting that transparency is increasingly being treated as a core design objective rather than a secondary visualization step (Pacal et al., 2026; Jenifer et al., 2026; Hussain et al., 2026).

Motivated by recent developments, this study investigates the performance of transfer learning architectures and an attention-augmented CNN for multiclass brain tumor

classification. Grad-CAM is employed to examine whether model predictions rely on diagnostically meaningful regions in MRI scans. The proposed approach focuses on both classification performance and interpretability.

The key novelty of this study is the integrated evaluation of predictive performance and explainability within a unified framework. Unlike prior studies that mainly focus on accuracy or provide limited qualitative interpretation, this work combines an attention-enhanced CNN with Grad-CAM visualization and evaluates explanation reliability using quantitative metrics. In addition, a comprehensive comparison is conducted across a baseline CNN, an attention-augmented CNN, and state-of-the-art pre-trained models, enabling a more holistic assessment of both performance and interpretability.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Study Design

The study provides a detailed procedure on how to classify cerebral neoplasm using MRI sequences with several DNN structures. The reviewed frameworks include CNN architecture with reference, various pre-trained transfer models and an attention-augmented implementation with diagnostic transparency enabled with Grad-CAM interpretability analysis.

The publicly available brain tumor MRI Dataset edited by Msoud Nickparvar (2021) was used as experimental procedures. It contains T1 weighted magnetic resonance sequences stratified in four categories of diagnosis- glioma, meningioma, pituitary adenoma and healthy tissue specimens. The entire data consisted of 5,617 MRI scans that were obtained on 5,617 different individuals. To avoid subject-level data leakage, patient-level independence was rigorously enforced during dataset partitioning. The dataset was divided into training (4,493 sequences from 4,493 individuals), validation (562 sequences from 562 individuals), and testing (562 sequences from 562 individuals), while preserving approximately balanced class distributions across all partitions (Table 1).

Table 1. Class distributions per split

Split	Class	Count	Percentage (%)
Training	Pituitary	1,165	25.93
	No tumor	1,200	26.71
	Meningioma	1,071	23.84
	Glioma	1,057	23.53
Validation	Pituitary	146	25.98
	No tumor	150	26.69
	Meningioma	134	23.84
	Glioma	132	23.49
Test	Pituitary	146	25.98
	No tumor	150	26.69
	Meningioma	134	23.84
	Glioma	132	23.49

Statistical check: $\chi^2=0.0015$, $p=1.0000$: No significant difference across splits, indicating distributional homogeneity

The Chi-square statistical test was used to identify the lack of significant distributional variance in diagnostic categories between the training, validation and testing cohorts ($\chi^2=0.0015$, $p=1.0000$), hence achieving statistical equivalence and balanced representation. The independence at the level of patients was strictly maintained with each individual identifier being represented only in one data partition.

Reference convolutional neural network architecture was developed to be used as a baseline. In this structure, three convolutional blocks were stacked in a series with 64, 128, and 256 kernels respectively followed by activation functions of Rectified Linear Units and maximum pooling respectively. A highly connected network of 512 neural units, regularized by a dropout probability of 0.5, dealt with the problem of overfitting, and a final softmax layer produced probabilistic observations in four categories of neoplasms. The process of training was done over 10 epochs using Adam as the optimization algorithm (learning rate=10, -3) and using a validation set at the same time.

The domain-specific adaptation of three pre-trained architectural models (MobileNetV2, InceptionV3, and Xception) to identify neoplasms was performed to increase the ability to extract features and the ability of the model to generalize. Training modules based on original classification on ImageNet were replaced by a global average pooling mechanism, dropout regularization, and a quaternary-class output layer.

The inputs of MobileNetV2 were normalized to the range of -1, +1 as a pre-training requirement. Early training held frozen base layers during 10 epochs then followed by biased unfreezing and refinement of the top 30 layers with a reduction learning rate (10^{-4}), Inception V3 and Xception designs used a bifurcated design: initial training of classifier heads was followed by progressive unfreezing of additional convolutional layers, and extensive fine-tuning was done over the last 20 epochs. This hierarchical model compromised between maintenance of previously learned low-level and intermediate-level representations and domain-specific adaptation to the properties of MRI imaging.

Moreover, the convolutional architecture was enhanced using a convolutional block attention module (CBAM)-inspired attention mechanism to improve both predictive performance and interpretability. Unlike a standalone attention block, the proposed attention modules are directly integrated into the convolutional pipeline to refine intermediate feature representations.

Specifically, the attention mechanism is embedded after convolutional blocks, allowing the network to adaptively recalibrate feature maps during feature extraction. The attention process is applied sequentially, consisting of channel attention followed by spatial attention. In the channel attention stage, global average pooling is used to aggregate channel-wise statistics, which are then passed through fully connected layers to model inter-channel dependencies. This enables the network to emphasize the most informative feature channels.

Subsequently, the spatial attention module focuses on identifying important spatial regions within the feature maps, allowing the model to localize diagnostically relevant tumor structures more effectively. This sequential channel-spatial

design improves feature discrimination while introducing minimal computational overhead.

For training, the attention-enhanced CNN was optimized using the Adam optimizer with a learning rate of 0.001, selected due to its stable convergence and adaptive learning capability observed during preliminary experiments. The batch size was set to 32 to balance computational efficiency and generalization performance under hardware constraints. All input images are resized to 128×128 and normalized to the (0,1) range. The model was trained for 15 epochs, which was empirically determined to achieve convergence while avoiding overfitting.

Extensive comparisons of models across all five models (MobileNetV2, InceptionV3, Xception, reference CNN, and attention-augmented CNN) used F1-score, classification accuracy, precision values and sensitivity values. Confusion matrices provided visual representation of category-specific prediction patterns. MobileNetV2 demonstrated superior accuracy, substantiating the efficacy of computationally efficient architectures in medical imaging applications. The attention-enhanced CNN achieved competitive secondary performance, validating the contribution of attention mechanisms toward refined spatial feature localization.

Grad-CAM was used to visualize and interpret model predictions to produce heatmaps, which indicate areas that were most significant to the classification choices. The heatmaps overlaid on MRI images with a JET colormap showed that the most successful models, in this case the attention CNN, paid attention to the tumor-affected regions, which is in accordance with clinical principles and increases the confidence in the work of automated diagnostic systems.

Figure 1 represents the general workflow, comprising of the MRI preprocessing, CNN-based classification, decision evaluation, Grad-CAM visualization, and interpretability analysis.

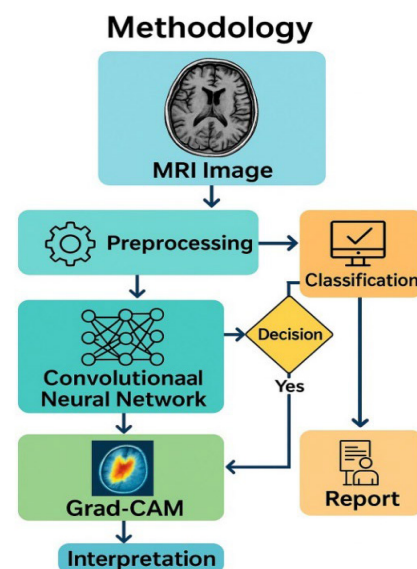


Figure 1. Flow diagram illustrating the deep learning methodology of categorizing tumors of the brain based on MRI images

Grad-CAM: Gradient-weighted class activation mapping, MRI: Magnetic resonance imaging

RESULTS

Table 2 delineates the predictive efficacy of five evaluated architectures on the reserved testing cohort: reference CNN, attention-augmented CNN, MobileNetV2, InceptionV3, and

Xception. All reported metrics represent aggregate values with corresponding standard deviations derived from three independent training iterations, accompanied by 95% confidence intervals (CIs). Across all examined frameworks, MobileNetV2 exhibited superior and most consistent performance characteristics (accuracy=0.917±0.007, F1=0.914±0.008, PR-AUC=0.972±0.002), demonstrating robust discriminative capacity and reliable generalization across experimental replicates. Such high-quality performance can be linked to its highly efficient depthwise separable convolutional computations and learned hierarchies of features that are able to capture the neoplasm-specific morphological features of MRI series. Attention-enhanced CNN architecture obtained second place ranking (accuracy=0.876±0.018), which is around 6.8 percentage point higher than the reference CNN, therefore vindicating the fact that the channel-spatial attention mechanisms increase the localization of diagnostically relevant tumor structures. The same can be said of InceptionV3 (accuracy=0.874±0.001) and Xception (accuracy=0.909±0.011), which also showed high and consistent performance, demonstrating that the modern factorized and computationally efficient backbone architectures could be transferred to the field of medical imaging. The accuracy of the reference CNN (0.808±0.099) was worse with high variability, which indicates the low optimization stability as compared to pre-trained frameworks. Paired t-tests and Wilcoxon signed-rank tests were statistically validated that MobileNetV2 is of great superiority to InceptionV3 and attention-augmented CNN in terms of accuracy and F1-score measures ($p < 0.05$). Further Xception enjoyed expected calibration error (ECE) of 0.032±0.012 which represents the best probabilistic prediction reliability. All of these results support the conclusion that paradigms of transfer learning and attention-mediated mechanisms significantly increase model stability, predictive accuracy, and calibration fidelity in cerebral neoplasm MRI category tasks.

The quantitative analysis of model transparency showed that the Grad-CAM attribution visualizations were highly reliant on the parameters of trained networks, thus establishing them as model-dependent visualization systems of explaining. The validation of integrity provided an average correlation coefficient of -0.0815 ± 0.1137 between the original and parameter randomization activation maps, which implied that saliency visualizations indeed changed greatly during the course of training rather than being a manifestation of inherent structural prejudices. Causal perturbation tests also supported the explanatory fidelity, where deletion area over the perturbation curve (AOPC) scored 94.32 and insertion AOPC scored 87.34, which is consistent with expected loss of confidence with deletion of salient regions and recovery of the confidence in the case of improvement of the diagnostically relevant regions. All these cumulative results justify the soundness and accuracy of Grad-CAM-based explanations, which can be combined in the future with clinical interpretation systems and diagnostic reliability standards.

DISCUSSION

The results show that deep learning models, particularly pre-trained architectures, are effective for brain tumor classification from MRI, with MobileNetV2 achieving the highest accuracy of 91.7%. The attention-augmented CNN also improved interpretability by emphasizing anatomically relevant regions during classification.

Nevertheless, the clinical applicability of these models depends on their robustness to MRI noise, data variability, and potential domain shifts across different scanner settings. Although the integrated attention mechanism and pre-trained backbones may reduce some variability by focusing on diagnostically relevant regions, further evaluation on external datasets and through cross-validation would provide stronger evidence of generalizability.

Table 2. Performance of multi-run tests

Architecture	Precision	F1-score	Classification accuracy	Sensitivity	ECE	PR-AUC
Xception	91.1±1.4% (87.7, 94.6)	90.6±1.3% (87.3, 93.8)	90.9±1.1% (88.2, 93.7)	90.6±1.2% (87.8, 93.5)	3.2±1.2%	96.8±0.2% (96.3, 97.3)
MobileNetV2	91.5±0.8% (89.6, 93.4)	91.4±0.8% (89.4, 93.3)	91.7±0.7% (89.9, 93.5)	91.4±0.7% (89.6, 93.2)	3.6±0.5%	97.2±0.2% (96.6, 97.8)
Attention-augmented CNN	88.2±2.6% (81.6, 94.7)	87.3±2.0% (82.4, 92.2)	87.6±1.8% (83.2, 92.0)	87.2±1.9% (82.6, 91.8)	4.0±2.2%	93.8±2.0% (88.8, 98.7)
InceptionV3	87.7±0.7% (86.0, 89.3)	87.1±0.2% (86.5, 87.7)	87.4±0.1% (87.2, 87.7)	87.1±0.2% (86.7, 87.5)	4.3±1.6%	94.7±0.4% (93.7, 95.7)
Reference CNN	86.6±3.6% (77.6, 95.6)	80.7±10.1% (55.7, 105.7)	80.8±9.9% (56.3, 105.3)	80.6±9.9% (56.0, 105.1)	6.6±0.3%	90.3±2.8% (83.3, 97.2)

ECE: Expected calibration error, PR-AUC: Precision-recall area under the curve, CNN: Convolutional neural network

Figure 2 indicates the visualization of a meningioma MRI by Grad-Cam as the regions of a tumor are drawn where the model is sensitive to the diagnostically significant information, and, therefore, learning behavior can be interpreted and relied on.

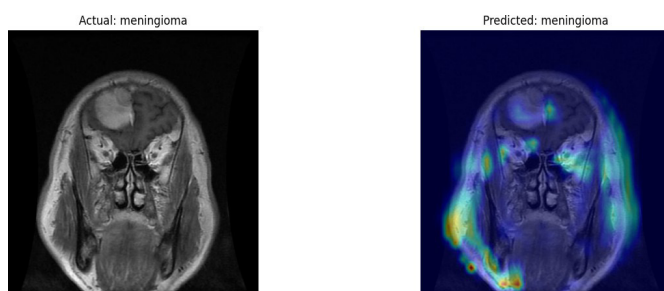


Figure 2. Grad-CAM dating of model focus on tumor areas during the MRI classification of meningioma

Grad-CAM: Gradient-weighted class activation mapping, MRI: Magnetic resonance imaging

In addition, the reliability of the Grad-CAM explanations was assessed beyond visual inspection using parameter randomization and causal perturbation tests, supporting the clinical relevance of the generated heatmaps. Despite these strengths, the study remains limited by the use of a single public dataset, the lack of external validation, and the restricted spatial resolution of Grad-CAM explanations.

Even so, the proposed framework offers practical advantages for clinical deployment, including strong predictive performance, computational efficiency, and improved transparency for MRI-based brain tumor assessment.

CONCLUSION

It was in this study that predictive performance evaluation was combined with transparency analysis to determine the efficacy

of different DNN designs to categorize cerebral neoplasms using MRI sequences. One of the most notable contributions is the serious comparative evaluation of the reference CNN, attention-enhanced CNN, and various pre-trained models in a single experimental framework that has provided the way that computationally efficient models of MobileNetV2 can be used to attain higher classification accuracy at low computational costs. The study also demonstrates that in addition to determining optimal predictive performance, combination of Grad-CAM visual attribution techniques explains the spatial concentration of algorithmic attention on tumor-specific anatomical areas, hence creating meaningful networks between computational predictions and diagnostic utility. These results are indicative of the novelty of combining the comparative efficacy analysis with interpretability mechanisms in a unified framework which confirms the potential of clear deep learning systems as a clinically feasible radiology tool.

Three strategic directions for future research are as follows: developing interpretable clinical decision support interfaces that combine calibrated uncertainty quantification with visual attribution overlays to boost radiologist confidence and enable practical use in real clinical settings; implementing ensemble techniques that learn to combine heterogeneous architectures to enhance predictive stability and probability calibration; and adding multimodal MRI contrasts to improve their ability to distinguish tumor subtypes.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: MAM, OAE, ES; Design: MAM, OAE, ES; Control: MAM, OAE, ES; Resources: MAM, OAE, ES; Materials: MAM, OAE, ES; Data Collection and/or Processing: MAM, OAE, ES; Analysis and/or Interpretation: MAM, OAE, ES; Literature Review: MAM, OAE, ES; Writing the Article: MAM, OAE, ES; Critical Review: MAM, OAE, ES.

REFERENCES

- Atagün, E., Temür, G., & Biroğul, S. (2026). Graph-based interpretable dialogue sentiment analysis: a HybridBERT-LSTM framework with semantic interaction explainer. *Computer Standards & Interfaces*, 97, 104086. <https://doi.org/10.1016/j.csi.2025.104086>
- Brima, Y., Atemkeng, M., Tankio Djiokap, S., Ebiele, J., & Tchakounté, F. (2021). Transfer learning for the detection and diagnosis of types of pneumonia including pneumonia induced by COVID-19 from chest X-Ray images. *Diagnostics*, 11(8), 1480. <https://doi.org/10.3390/diagnostics11081480>
- Chaddad, A., Hu, Y., Wu, Y., Wen, B., & Kateb, R. (2025). Generalizable and explainable deep learning for medical image computing: an overview. *Current Opinion in Biomedical Engineering*, 33, 100567. <https://doi.org/10.1016/j.cobme.2024.100567>
- Chatterjee, S., Das, A., Mandal, C., Mukhopadhyay, B., Vipinraj, M., Shukla, A., ..., & Nürnberger, A. (2022). TorchEsegeta: framework for interpretability and explainability of image-based deep learning models. *Applied Sciences*, 12(4), 1834. <https://doi.org/10.3390/app12041834>
- Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 839-847). IEEE. <https://doi.org/10.1109/WACV.2018.00097>
- Ebiele, J., Ansah-Narh, T., Djiokap, S., Proven-Adzri, E., & Atemkeng, M. (2020). Conventional machine learning based on feature engineering for detecting pneumonia from chest X-Rays. In Proceedings of the 2020 Conference of the South African Institute of Computer Scientists and Information Technologists (pp. 149-155). Association for Computing Machinery. <https://doi.org/10.1145/3410886.3410898>
- Hasan, M. R., Rudra, S. M., Karmoker, N., Yousuf, M. A., Akhter, J., Al-Moisheer, A. S., ..., & Moni, M. A. (2026). 3D MRI reconstruction and brain tumor diagnosis using deep learning with explainable AI. *Expert Systems with Applications*, 315, 131513. <https://doi.org/10.1016/j.eswa.2026.131513>
- Haupt, M., Maurer, M. H., & Thomas, R. P. (2025). Explainable Artificial Intelligence in radiological cardiovascular imaging-a systematic review. *Diagnostics*, 15(11), 1399. <https://doi.org/10.3390/diagnostics15111399>
- Houssein, E. H., Gamal, A. M., Younis, E. M. G., & Mohamed, E. (2025). Explainable Artificial Intelligence for medical imaging systems using deep learning: a comprehensive review. *Cluster Computing*, 28(7), 469. <https://doi.org/10.1007/s10586-025-05281-5>
- Hussain, M. I., Chowdhury, S. H., Hossain, M. M., & Mamun, M. (2026). NeuroBlend-3: hybrid deep and machine learning framework with explainable AI for multi-class brain tumor detection using MRI scans. *Medinformatics*, 3(1), 56-66. <https://doi.org/10.47852/bonviewMEDIN52026540>
- Jenifer, G. G., & Rajakumar, G. (2026). HyBiUnet-TransMask: a clinical decision system with segmentation and classification for interpretable brain tumor diagnosis in multimodal MRI. *Biomedical Signal Processing and Control*, 115, 109393. <https://doi.org/10.1016/j.bspc.2025.109393>
- Kumar, V. D. R., & Singh, P. (2026). HCAC: trustworthy hybrid attentive framework tackling brain tumor identification with XAI. *IEEE Access*, 14, 16158-16175. <https://doi.org/10.1109/ACCESS.2026.3656212>
- Lamrani, D., Mahjoubi, M. A., Shawki, S., Moutaouakil, W., Hamida, S., ..., & Bahatti, L. (2026). Enhanced multiclass brain tumor segmentation using MRI images and explainable AI techniques. *Multimedia Tools and Applications*, 85(1), 28. <https://doi.org/10.1007/s11042-026-21180-2>
- Martí-Juan, G., Frías, M., García-Vidal, A., Vidal-Jordana, A., Alberich, M., Calderon, W., ..., & Pareto, D. (2022). Detection of lesions in the optic nerve with magnetic resonance imaging using a 3D convolutional neural network. *NeuroImage: Clinical*, 36, 103187. <https://doi.org/10.1016/j.nicl.2022.103187>
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, & Müller, K. R. (Eds.), *Explainable AI: Interpreting, explaining and visualizing deep learning* (Lecture Notes in Computer Science, Vol. 11700, pp. 193-209). Springer. https://doi.org/10.1007/978-3-030-28954-6_10
- Nickparvar, M. (2021). Brain tumor MRI dataset. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/2645886>
- Pacal, I., & Banerjee, T. (2026). Towards accurate and interpretable brain tumor diagnosis: T-FSPANNet with tri-attribute and pyramidal attention-based feature fusion. *Biomedical Signal Processing and Control*, 113, 108852. <https://doi.org/10.1016/j.bspc.2025.108852>
- Philbrick, K. A., Yoshida, K., Inoue, D., Akkus, Z., Kline, T. L., Weston, A. D., ..., & Erickson, B. J. (2018). What does deep learning see? Insights from a classifier trained to predict contrast enhancement phase from CT images. *American Journal of Roentgenology*, 211(6), 1184-1193. <https://doi.org/10.2214/AJR.18.20331>
- Saleem, H., Shahid, A. R., & Raza, B. (2021). Visual interpretability in 3D brain tumor segmentation network. *Computers in Biology and Medicine*, 133, 104410. <https://doi.org/10.1016/j.compbiomed.2021.104410>
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: visual explanations from deep networks via gradient-based localization. In 2017 IEEE International Conference on Computer Vision (ICCV) (pp. 618-626). IEEE. <https://doi.org/10.1109/ICCV.2017.74>

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In D. Precup & Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 3319-3328). Proceedings of Machine Learning Research. <https://proceedings.mlr.press/v70/sundararajan17a.html>

Wood, D. A., Kafiabadi, S., Al Busaidi, A., Guilhem, E., Montvila, A., Lynch, J., ..., & Booth, T. C. (2022). Deep learning models for triaging hospital head MRI examinations. *Medical Image Analysis*, 78, 102391. <https://doi.org/10.1016/j.media.2022.102391>

Zeineldin, R. A., Karar, M. E., Elshaer, Z., Coburger, J., Wirtz, C. R., ..., & Mathis-Ullrich, F. (2022). Explainability of deep neural networks for MRI analysis of brain tumors. *International Journal of Computer Assisted Radiology and Surgery*, 17(9), 1673-1683. <https://doi.org/10.1007/s11548-022-02619-x>

Binary text classification using machine learning and transformer models
for small and imbalanced datasetsİsmail Şenyurt^{*1}, Sinan Toklu²¹Graduate Student, Department of Computer Engineering, Graduate School of Natural and Applied Sciences,
Gazi University, Ankara, Türkiye²Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Türkiye**Cite this article as:** Şenyurt, İ., & Toklu, S. (2026). Binary text classification using machine learning and transformer models for small and imbalanced datasets. *J Comp Electr Electron Eng Sci*, 4(2), 83-91. <https://doi.org/10.51271/JCEEES-0041>

Received: 14.06.2026

Accepted: 22.07.2026

Published: 13.10.2026

ABSTRACT

Aims: The increasing speed with which platforms providing digital content have entered our lives has made the classification of the content on these platforms important. In this study, the binary classification problem of Netflix content as a movie or show was addressed, the effect of metadata features and text descriptions on an accurate classification process was examined and a new hybrid approach was presented.**Methods:** Two Netflix datasets from open sources (Kaggle and Hugging-Face) were used in the study. Furthermore, so as to eliminate the imbalance in the datasets, machine learning (ML) and deep learning (DL) models with natural language processing (NLP) were used separately and hybridly on the two datasets downloaded for shows and movies, both individually and combined. In this way, the dataset was examined separately as imbalanced and balanced, and the imbalance was eliminated by merging two datasets.**Results:** The proposed hybrid framework achieved the results of 0.970940 accuracy, 0.982801 precision, 0.936768 recall and 0.959233 F1-score.**Conclusion:** It has been demonstrated through experimental results that the obtained results are better than those of other models. In the experiments, various machine learning methods and transformer-based language models were evaluated; it was observed that the metadata-based models showed high performance in terms of accuracy, whereas BERT and similar models were particularly successful in detecting the 'show' class. The study reveals that the structured metadata and textual data are complementary in nature, and it is shown that hybrid approaches combining machine learning with transformer models would produce more balanced and robust results.**Keywords:** Text classification, natural language processing, ensemble learning, hybrid learning, ELECTRA

INTRODUCTION

In addition to the existence of many artificial intelligence applications containing text data, big data is required for the model to learn meaningful representations effectively. More consistent and accurate results can be obtained by utilizing the positive aspects of one or more datasets. Recent studies have demonstrated that the use of natural language processing and machine learning techniques is increasing day by day to improve text classification performance. The dataset of this study was previously examined by Salsabila (2024), who evaluated text classification using Decision Tree and Random Forest algorithms, providing a baseline for further investigation. In addition to the proposed hybrid model, several machine learning and transformer models were evaluated under the same experimental conditions to provide a comprehensive performance comparison. Furthermore, unlike the study by Salsabila (2024), model performance was assessed using not only accuracy but also recall, precision and F1-score metrics to provide a more comprehensive evaluation.

In this study, binary text classification experiments were conducted on a small and initially imbalanced dataset. To mitigate the effects of class imbalance, the original dataset was integrated with a second dataset. The study further investigates the effect of combining transformer-based text representations with both traditional machine learning and hybrid approaches under a unified experimental framework. In addition, the performance of these approaches was examined on both the original and the merged datasets to assess the impact of dataset balancing.

Despite the remarkable advances in transformer-based natural language processing, hybrid and ensemble learning for small and imbalanced binary text classification datasets remain limited. To address this gap, this study presents a unified experimental framework that systematically compares these approaches under identical preprocessing, training and evaluation conditions.

^{*}Corresponding Author: İsmail Şenyurt, 24802002006@gazi.edu.tr

The main contributions of this study are summarized as follows;

A comprehensive framework is presented to compare machine learning, transformer-based, hybrid and ensemble learning approaches for binary text classification on small and initially imbalanced datasets.

The effectiveness of integrating efficiently learning an encoder that classifies token replacements accurately (ELECTRA)-based contextual text representations with conventional machine learning algorithms is investigated through a hybrid framework, providing a detailed comparison with standalone transformer models.

The impact of balanced datasets is analyzed systematically to show its influence on the classification performance of different learning approaches.

An ensemble learning strategy is evaluated to investigate whether combining multiple classifiers can further improve the predictive performance of transformer-based hybrid models.

The framework is established by reporting both overall and class-wise performance metrics (accuracy, precision, recall, F1-score), providing a more reliable assessment of classifier behaviour, particularly for small and imbalanced datasets.

RELATED WORKS

One of the studies conducted in this field is the work of Shafiee (2024). Although this study examines in detail how machine learning techniques are used in recommendation systems, it does not provide systematic experimental comparisons regarding the text-based classification performance of transformer-based large language models (Table 1).

Additionally, in another study, Motamedi et al. (2024) applied a ML method based on metadata, audio and visual features to predict movie scores. In their study, the problem was treated as two separate binary classifications (high/low E and H) and classical ML algorithms such as logistic regression (LR), support vector machines (SVM), decision trees (DT), k-nearest neighbours and random forests (RF) were evaluated using the nested cross-validation method. The results showed that metadata features, particularly those containing genre information, provided the highest performance. DL models were also tested, but they did not yield better results than classical ML methods due to the relatively small size of the dataset (709 films). In another important study in this field, Shaikh and Singh (2025) used the cancer-risk-factors.csv dataset, which includes demographic, lifestyle, genetic and environmental factors to comparatively evaluate SVM, LR, RF and XGBoost algorithms for the purpose of predicting

Table 1. Similar studies in the literature

Study	Preferred dataset	Models	Positive ways	Limitations	Accuracy-precision-recall-F1-score
Motamedi et al. (2024)	IMDb	LR, SVC, DT, RF,	Using Cross validation technique	Dataset is limited, contains only 709 films	0.900-0.894-0.894-0.895
Salsabila (2024)	Netflix dataset	RF, DT	Feature extraction methods were added by using DT and RF algorithms	Study is limited with one dataset. Evaluations were done with limited features	0.968, -, -, -
Proposed study	Two different datasets from Kaggle and hugging face	LR, SVM, RF, DT, BERT, RoBERTa, DeBERTa, DistillBERT, ELECTRA	Utilizing both ML and DL methods	Datasets are not enough for DL models	0.970-0.982-0.936-0.939
Shaikh and Singh (2025)	cancer-risk-factors.csv	SVM, LR, RF, XGBoost	Multi-label prediction and evaluation of all metrics	Class imbalance	0.900-0.894-0.900-0.896
Dritsas and Trigka (2025)	NYC Taxi Trip Duration, Netflix Prize, Higgs Boson	LinR, LR, RF, GBT, SVR, KNN	Implementing more than one dataset, evaluation and monitoring of CPU/RAM computations	Limited with only ML models	0.900-0.911-0.892-0.901
Galke et al. (2025)	20NG, r8, r52, OHSUMED, MR	RNN, SGM, LSTM, BERT, GPT, RoBERTa, DistillBERT, DeBERTa	Using several ways such as LLM and SLM and graph based approaches	Only comparison, not available any new method	0.872, -, -, -
Nilla and Setiawan (2024)	IMDb data from Twitter	CNN, RoBERTa	Merging RoBERTa and CNN1d and using SMOTE	Only accuracy evaluation	0.864, -, -, -
Chaudhary (2024)	Short Jokes(kaggle), non humorous dataset (Reuters)	SBERT, LR, SVC, NB, BERT, DistillBERT, RoBERTa	Using both ML and DL methods	Results obtained from DL models are lower than ML models, that's why has limited generalization power	0.784, -, -, 0.781
Ammar and Rania (2021)	Arabic Corpus, AJGT, IMDB review, SemEval, COVID19-Fake, ArSarcasm	GB, SVM, NB, LR, RF, LSTM, GRU, CNN, LSTM	Using ensemble method with six datasets	Only accuracy evaluation	0.781, -, -, -
Wang et al. (2024)	COVID-19-related Tweets, Economiz Texts, E-Commerce Texts, SMS Spam Collection	MNB, LR, RF, DT, KNN, GRU, LSTM, RNN, DeBERTa, GPT 3.5, GPT 4, Gemini-pro, Llama 3, Qwen 14b	Using both ML and DL methods	The success of GRU model depends on fine tuning and computation capacity	0.938, -, -, 0.938

IMDb: Internet movie database, LR: Logistic regression, SVC: Support vector classifier, DT: Decision tree, RF: Random forest, SVM: Support vector machine, BERT: Bidirectional encoder representations from transformers, RoBERTa: Robustly optimized BERT pretraining approach, DeBERTa: Decoding-enhanced BERT with disentangled attention, DistillBERT: Distilled BERT, ELECTRA: Efficiently learning an encoder that classifies token replacements accurately, XGBoost: Extreme gradient boosting, NYC: New York city, LinR: Linear regression, GBT: Gradient-boosted trees, SVR: Support vector regressor, KNN: K-nearest neighbors, CPU: Central processing unit, RAM: Random access memory, 20NG: 20 newsgroups, R8: Reuters-8, R52: Reuters-52, OHSUMED: OHSUMED medical literature dataset, MR: Movie reviews, RNN: Recurrent neural network, SGM: Sequence generation model, LSTM: Long short-term memory, GPT: Generative pre-trained transformer, CNN: Convolutional neural network, SMOTE: Synthetic minority over-sampling technique, SBERT: Sentence-BERT, NB: Naive bayes, GB: Gradient boosting, GRU: Gated recurrent unit, MNB: Multinomial naive bayes, LLM: Large language model, SLM: Small language model, DL: Deep learning, ML: Machine learning

cancer risk levels and the problem was treated as a multi-class classification problem. According to the results obtained, the LR model showed the highest performance with an accuracy rate of 90%, followed by the SVM, XGBoost and RF models. Precision, recall and F1-score metrics were also reported and model performance was detailed through confusion matrix analysis. However, the study primarily focused on the accuracy metric, did not comprehensively address the class imbalance problem, and did not evaluate DL based methods.

In their study on the Apache Spark ecosystem, Dritsas and Trigka (2025) addressed a multi-class classification problem aimed at categorizing movie ratings using a large-scale movie dataset. In the study, a classical ML approach was adopted and by using the RF algorithm, values of 0.90 accuracy, 0.91 precision, 0.892 recall and 0.901 F1-score were obtained. These results demonstrated that ensemble-based methods can be effective in multi-classification problems within big data environments and DL algorithms and transformer-based architectures were not included in the study. Additionally, in another study, Galke et al. (2025) systematically examined the progress in the field of text classification, comparing a wide family of methods ranging from BoW-based classical ML models to transformer-based language models and graph-based approaches. The study revealed that fine-tuned SLMs in particular, exhibited more stable and higher performance compared to in-context LLMs. It was reported that graph-based methods failed to surpass strong baselines in most cases.

In a study similar to these topics conducted by Nilla and Setiawan (2024), a hybrid movie recommendation system was developed by combining the content-based filtering method with a RoBERTa-based word embedding approach and 1D convolutional neural network (CNN) classification. In the study, 845 movies and 34,086 Twitter comments were used and data imbalance was addressed using the SMOTE method. As a result of a three-stage experimental design, the CNN model achieved 86.41% accuracy with SGD optimization and an appropriate learning rate. Although this study presents a structure that integrated deep learning classification with content-based filtering, it did not directly fine-tune transformer models and did not report detailed performance metrics such as precision, recall and F1-score.

Another study in the literature is the one conducted by Chaudhary (2024). In this study, classical ML methods and DL models were compared for the classification of social media texts. Traditional methods such as Naive Bayes, SVM and LR as well as CNN, LSTM and BiLSTM architectures, were conducted in the study. Furthermore, an accuracy of 0.7767 and F1-score of 0.7819 were obtained with DistilBERT; an accuracy of 0.7823 and F1-score of 0.7772 with the BERT model; an accuracy of 0.7841 and F1-score of 0.7778 with the RoBERTa model and these values lagged behind the results obtained from ML methods. In another study, Ammar and Rania (2021) applied both ML and DL techniques using the IMDB dataset, one of their utilized datasets and achieved accuracy values of 0.7815 with LSTM1 and 0.7841 with GRU1. In the study, individual models were compared with an ensemble structure and it was demonstrated that the ensemble model exhibited higher and more stable performance in terms of accuracy and F1-score, particularly on the IMDB dataset. Additionally, Wang et al. (2023) comparatively evaluated traditional ML algorithms, DL models and LLMs in the

problem of classifying e-commerce product texts. In the study, the GRU model showed the highest performance among models without fine-tuning, achieving 0.9387 accuracy and 0.9383 F1-score; this revealed that GRU's sequence and context modeling capacity is effective in product text classification.

Although previous studies have demonstrated the effectiveness of both conventional machine learning and transformer-based approaches for text classification, most studies focus on a single learning paradigm. Furthermore, few studies systematically compare machine learning, transformer-based, hybrid, and ensemble learning approaches under identical experimental conditions while simultaneously addressing class imbalance through dataset enrichment and reporting class-wise performance metrics. These limitations constitute the primary motivation for the proposed framework.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Dataset Description

This study aims to provide a comprehensive comparison of traditional machine learning and transformer-based models for binary text classification. Two publicly available Netflix datasets were employed in this study. The primary dataset was obtained from Kaggle (<https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies?select=titles.csv>), while a second dataset was collected from Hugging Face (<https://huggingface.co/datasets/hugginglearners/netflix-shows>) to support the dataset balancing process described in section 3.2.

The datasets contain both numerical and textual information describing Netflix content. Each row includes attributes such as content title, description, release year and content type (show or movie). The combination of heterogeneous feature types enables the evaluation of both machine learning algorithms and transformer-based language models within the same experimental framework.

The primary Kaggle dataset consists of 5,849 instances represented by 15 features. The target feature is the content type, where each instance is classified as either a movie or a show. The original dataset has an imbalanced distribution with nearly 64% of the rows belonging to the movie class and 36% to the show class. The detailed preprocessing and balancing procedures are presented in section 3.2.

Data Preprocessing and Dataset Balancing

Prior to model development, the datasets underwent a comprehensive preprocessing procedure to improve data quality and minimize bias. Missing parts of the dataset were examined first and replaced with the value 'unknown' to preserve the available samples. In addition, the 'runtime' and 'seasons' features were excluded because they might point the target class directly. Duplicate records were identified and removed in order to improve the generalization power of the models. Furthermore, textual descriptions containing clue words such as 'movie, show, series, television and mini-series' were eliminated to prevent models from making biased

predictions. Finally, after the dataset was divided into training and test subsets an additional duplicate check was performed to prevent potential data leakage.

The original dataset (DF1) consisted of 3.744 movie instances and 2.106 show instances, resulting in an imbalanced class distribution. To investigate the most appropriate balancing strategy, DF1 was merged with another Netflix dataset (DF2), which contained 6.131 movie instances and 2.676 show instances. The dataset consisted of 9875 movie samples and 4782 show samples after combining. Four approaches were considered: (1) increasing the number of show class to 9875 by oversampling, (2) reducing the movie class to 4782 by downsampling, (3) using the merged dataset without doing anything. (4) combining show samples (reaching 4782 rows) from two datasets and adding 1038 movie samples of second dataset to first dataset (containing 3744 movie sample) to balance each class. The relatively small size of the dataset was the main reason why oversampling and downsampling approaches did not provide the expected improvements. During oversampling, the minority class had to be augmented with nearly its original size, while downsampling reduced limited training data. Therefore, the fourth strategy, which preserved minimal sampling process was selected for the experiments. The comparative performance of these ways is presented in the result section.

Experimental Workflow

Figure 1 illustrates the whole experimental workflow adopted in this study. Data collection, preprocessing and balancing parts were described in previous sections. The dataset was divided into test and training sets with an 80:20 ratio. In addition, 5-fold-cross-validation was employed during the model development phase to reduce the risk of overfitting. The experimental framework consists of four consecutive stages. In the first stage, traditional machine learning algorithms were trained and in the second stage, several transformer-based models were fine-tuned on the same dataset. In the third stage, transformer models were integrated with machine learning classifiers to construct a hybrid model and ensemble techniques were employed for further performance. Finally, all models were evaluated using the same experimental processes and detailed descriptions of model architectures are presented in the following sections.

Machine Learning Models

Machine learning algorithms were employed as the first stage of model development to set a baseline performance. Owing to their relatively low computational complexity, short training time and effectiveness of using small datasets, conventional machine learning models remain a suitable choice for text classification tasks. All machine learning models were trained on the same preprocessed dataset using an 80:20 ratio. This protocol is widely adopted in the literature for text classification cases. For example, Agrawal et al. (2025) reported that an 80:20 train-test split produced the highest classification accuracy, supporting the selection of the same partitioning strategy in the present study. Several machine learning algorithms were evaluated and these models include comprehensive comparison strategies. Their performance serves as a foundation for the hybrid and ensemble learning framework introduced in the subsequent sections.

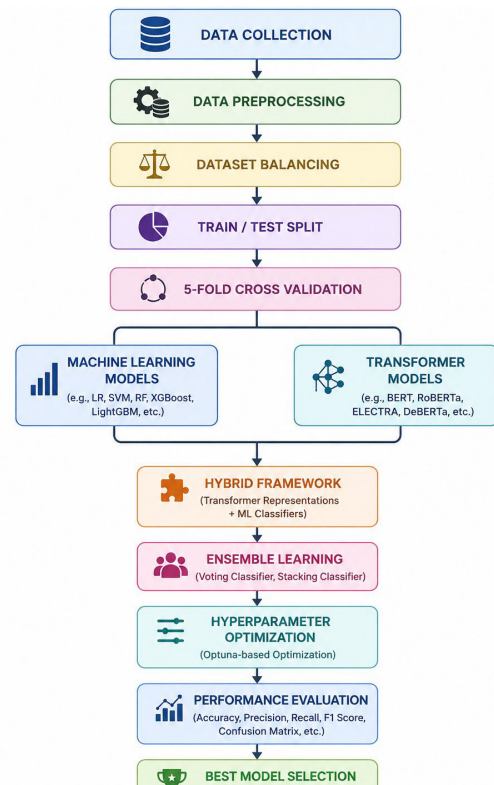


Figure 1. Workflow of the study

Transformer-Based Models

Transformer-based models were employed in the second stage of the experimental framework to investigate the contribution of contextual representations to binary text classification. Unlike conventional machine learning algorithms, transformer architectures are capable of capturing semantic relations and long-range text dependencies within the data by using attention mechanisms. Since the ‘description’ feature contains more semantic information defining each Netflix title, transformer models were expected to provide more information than traditional approaches. To provide a comprehensive comparison, six publicly available pretrained transformer models were evaluated, namely BERT-base-uncased, DistilBERT, RoBERTa, DeBERTa, ALBERT, ELECTRA. These models have different architectures such as computational complexity and pretraining strategy. Among them, ELECTRA was selected as the primary model because it achieved the most competitive overall performance during experiments and also it was used to construct a hybrid model in section 3.6. This hybrid design enables to evaluate a standalone classification and to integrate as a component within hybrid and ensemble approaches.

Hybrid and Ensemble Framework

Instead of relying solely transformer Electra model, a hybrid design was enabled to combine efficient ways of machine learning approaches. Within the proposed framework, the textual descriptions were first processed by the ELECTRA model to generate contextual embedding vectors. These feature representations were subsequently used as input for several machine learning classifiers including logistic regression, naive bayes, support vector machine, random forest, XGBoost and LightGBM. To further enhance predictive performance and improve robustness, ensemble learning techniques (voting and stacking classifiers) were incorporated into the proposed framework. This two-stage framework enables a systematic

comparison between transformer models, hybrid models and ensemble approaches under identical experimental conditions.

Hyperparameter Optimization

Hyperparameter optimization was performed for the ELECTRA model using the Optuna framework. Optuna was selected due to enable automatic exploration of the hyperparameter space while reducing unnecessary evaluations. A total of 20 trials were conducted, and during each trial, the model was evaluated on a dedicated validation set.

The optimization process included the tuning of several key hyperparameters, including the learning rate, batch size, number of training epochs, maximum sequence length, weight decay, warm-up ratio and dropout rate. The weighted F1-score obtained on the validation set to provide a more reliable performance score than accuracy. The highest validation score was selected as the final ELECTRA model and implemented subsequent experiments.

Performance Evaluation

The performance of all models was evaluated using a comprehensive set of metrics to ensure a reliable comparison. Therefore, in addition to accuracy, precision, recall and F1-score were adopted in order to assess the classification performance of models. To provide a more detailed analysis of class behaviours, class-wise performance metrics were also reported for both show and movie classes. Class-specific performance metrics enable the effectiveness of each model to be assessed individually and also help to understand the effect of dataset balancing and different learning strategies.

RESULTS

Baseline Comparison and Reproduction

The experimental configuration proposed by Salsabila (2024) was first reproduced using the same dataset and machine learning algorithms to establish a reliable baseline. The original study employed four features (release year, age certification, genres and production countries) and evaluated the performance of decision tree and random forest algorithms using different numbers of decision trees. The reproduced baseline results are presented in Table 2.

Table 2. Results obtained in the reference study conducted by Salsabila (2024)

	Accuracy	Correct prediction	Wrong prediction
Decision tree	0.9581	1121	49
Random forest (n=10)	0.9675	1132	38
Random forest (n=50)	0.9684	1133	37
Random forest (n=100)	0.9675	1132	38
Random forest (n=200)	0.9667	1131	39

To investigate the influence of categorical feature encoding on classification performance, the label encoder used in the reference study by Salsabila (2024) was replaced with one-hot encoding with same four features. One-hot encoding

was preferred because most of the features in the dataset are categorical and do not possess a natural ordinal relationship. One-hot encoding represents each category independently, providing a more suitable feature representation for classifiers. The comparative results are presented in Table 3.

Table 3. Comparative results derived from differences during the preprocessing stages

Model	Encoding	Accuracy	Precision	Recall	F1-score
Decision tree	Label encoder	0.9581	0.9582	0.9581	0.9582
	One-hot encoder	0.9615	0.9548	0.9391	0.9469
Random forest (n=10)	Label encoder	0.9675	0.9676	0.9675	0.9674
	One-hot encoder	0.9658	0.9801	0.9251	0.9518
Random forest (n=50)	Label encoder	0.9684	0.9685	0.9684	0.9682
	One-hot encoder	0.9709	0.9828	0.9368	0.9592
Random forest (n=100)	Label encoder	0.9675	0.9676	0.9675	0.9674
	One-hot encoder	0.9701	0.9804	0.9368	0.9581
Random forest (n=200)	Label encoder	0.9667	0.9668	0.9667	0.9665
	One-hot encoder	0.9692	0.9803	0.9344	0.9568

As shown in Table 3, replacing the encoding technique led to an increase in accuracy and precision, whereas recall decreased slightly. These results suggest that representing categorical variables independently provides a more informative feature representation, resulting in improved overall classification performance.

While classifying the ‘description’ feature was incorporated into the four metadata features in the Salsabila’s (2024) study to investigate the contribution of textual information. The proposed configuration additionally exploited semantic information contained in textual descriptions in addition to baseline model. The comparative results are presented in Table 4.

As shown in Table 4, incorporating the ‘description’ feature significantly improved the recall and F1-score, while only a slight reduction in accuracy and precision was observed. These results indicate that the semantic information contained in textual descriptions improves the model’s ability of classifying.

Performance of Machine Learning Models

The performance of the Naïve Bayes classifier was strongly influenced by the selected feature set among the other machine learning algorithms. Although the additional features improved the performance of the Naïve Bayes classifier, other algorithms, particularly random forest, achieved superior overall results. This observation suggests that while Naïve Bayes provides a competitive baseline for binary text classification, models capable of capturing more complex relationships between textual and structured features are better suited for this classification task.

Table 4. Effect of using different features in data preprocessing on model performance

Case	Accuracy	Precision	Recall	F1-score
The results of Salsabila’s (2024) study with 4 columns (release year, age certification, genres, production countries)	0.9684	-	-	-
The results of 5 features (4 features of Salsabila study + description column) (proposed method)	0.9605	0.9446	0.9985	0.9708

Performance of Deep Learning Models

Conventional machine learning models could achieve competitive performance, but they rely primarily on special feature representations and have limited capability in learning deep contextual relationships within textual data. Therefore, transformer-based deep learning models were further preferred to exploit their stronger representation learning capability and higher computational capacity. Before model training, the 'runtime' and 'seasons' features were excluded because they directly revealed the target class and could introduce data leakage.

Zero-shot learning, few-shot learning, full fine-tuning of transformer architectures and embedding-based hybrid approaches were investigated to elaborately evaluate different deep learning strategies. Six pretrained transformer models, namely BERT-base, DistilBERT, RoBERTa, DeBERTa, ALBERT, and ELECTRA, were evaluated under identical experimental settings to provide a fair comparison.

As presented in Figure 2, zero-shot learning produced the lowest performance as expected, because no task-specific adaptation was performed. Fine-tuning with a limited number of labeled samples substantially improved the results in the few-shot setting. Among the fully fine-tuned transformer architectures, ELECTRA achieved the highest weighted F1-score.

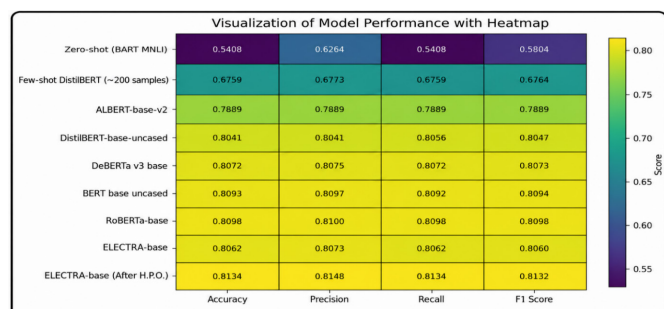


Figure 2. Comparison of values obtained with deep learning models

Class-wise performance results for ELECTRA are presented in Table 5 to provide a detailed evaluation. While the Show class obtained higher recall (0.8370), the movie class achieved higher precision and F1-score (0.8120), indicating that the proposed framework successfully identified both classes without bias toward either category. These findings demonstrate that balancing and preprocessing strategies contributed to improving generalization power of models.

Table 5. Class-wise performance of ELECTRA model

Class	Precision	Recall	F1-score	Support
Movie	0.8263	0.7753	0.8000	957
Show	0.7884	0.8370	0.8120	957
Macro average	0.8073	0.8062	0.8060	1914
Weighted average	0.8073	0.8062	0.8060	1914

ELECTRA: Efficiently learning an encoder that classifies token replacements accurately

Figure 3 Comparison of values obtained with deep learning models presents the training and validation performance of the ELECTRA model throughout the training process. While the training loss continuously decreased with each epoch, the validation loss began to increase after the third epoch,

indicating a sign of overfitting. A similar trend was observed for the accuracy curves, where training accuracy continued to improve whereas validation accuracy plateaued. To preserve the model's generalization capability, dropout regularization and early stopping were employed, preventing further performance degradation on the validation set.

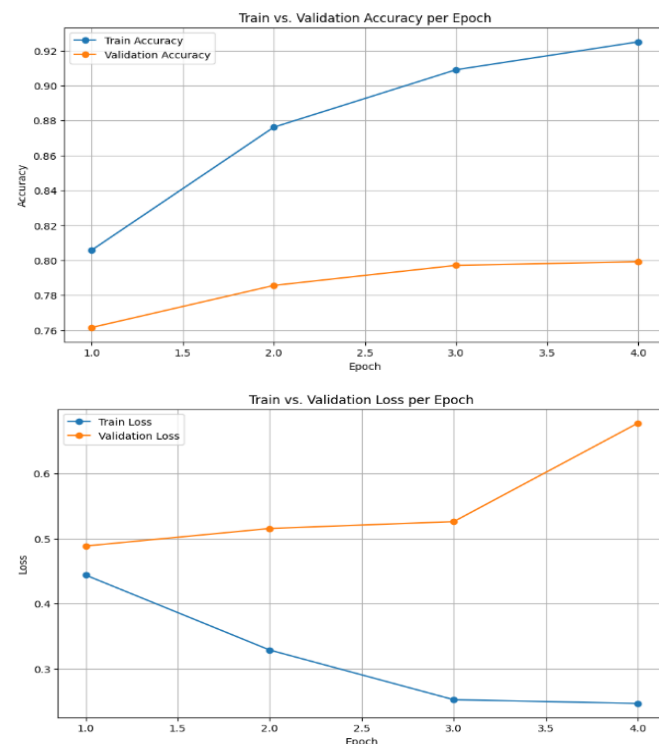


Figure 3. Performance during validation process

Performance of Hybrid and Ensemble Models

Following the comparative evaluation of the transformer architectures, ELECTRA was selected for the proposed hybrid framework because it achieved the highest overall performance among the transformer models. The contextual embeddings generated by ELECTRA were subsequently combined with conventional machine learning classifiers to investigate whether transformer-based feature representations could further improve classification performance.

To evaluate the influence of dataset composition, the proposed hybrid framework was trained separately using DF1, DF2, and the merged DF3 dataset. The corresponding results are presented in Table 6-8. Among the evaluated datasets, the merged dataset (DF3) consistently achieved the highest classification performance, indicating that the enriched dataset provided a more suitable foundation for learning robust feature representations. Consequently, all subsequent hybrid and ensemble experiments were conducted using DF3.

Table 6. Results of DF1 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8259	0.8063	0.8581	0.8314
Multinomial NB	0.8039	0.7597	0.8889	0.8192
Logistic regression	0.7871	0.7705	0.8179	0.7935
Random forest	0.7845	0.8305	0.7149	0.7683
LightGBM	0.7811	0.7933	0.7604	0.7765
VotingClassifier	0.8293	0.8263	0.8340	0.8301
StackingClassifier	0.8400	0.8396	0.8407	0.8401

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

Table 7. Results of DF2 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8256	0.8108	0.8493	0.8296
Multinomial NB	0.8038	0.7704	0.8657	0.8153
Logistic regression	0.7957	0.7940	0.7985	0.7962
Random forest	0.8391	0.9500	0.7158	0.8164
LightGBM	0.8112	0.8661	0.7363	0.7959
VotingClassifier	0.8542	0.8771	0.8239	0.8497
StackingClassifier	0.8726	0.8957	0.8436	0.8688

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

Table 8. Results of DF3 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8463	0.8277	0.8747	0.8505
Multinomial NB	0.8187	0.7864	0.8752	0.8284
Logistic regression	0.8131	0.8087	0.8201	0.8144
Random forest	0.8266	0.8848	0.7509	0.8124
LightGBM	0.8039	0.8509	0.7369	0.7898
VotingClassifier	0.8587	0.8640	0.8514	0.8577
StackingClassifier	0.8803	0.8915	0.8660	0.8786

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

The comparative performance of the transformer, hybrid, and ensemble approaches is summarized in [Table 9](#).

Table 9. Results of DF3 dataset with deep learning, hybrid and ensemble methods

Model	Accuracy	Precision	Recall	F1-score
ELECTRA (full fine-tune)	0.8177	0.8191	0.8177	0.8174
ELECTRA+ML (best RF)	0.8223	0.8223	0.8223	0.8223
ELECTRA+ML (stacking)	0.8228	0.8229	0.8228	0.8228

ELECTRA: Efficiently learning an encoder that classifies token replacements accurately, ML: Machine learning, RF: Random forest

ELECTRA+random forest achieved the highest performance among the hybrid models. Furthermore, the Stacking ensemble slightly outperformed the best hybrid model, increasing the weighted F1-score from 82.23% to 82.28%. Although the numerical improvement is modest, the result indicates that ensemble learning could provide a performance gain.

Hyperparameter Optimization Results

The influence of hyperparameter optimization on the ELECTRA model is summarized in [Table 10](#). Compared with the default configuration, the optimized parameter setting resulted in improved classification performance and provided the final configuration used throughout the hybrid and ensemble experiments. The optimization process identified suitable values for the learning rate, weight decay, warm-up ratio, batch size, maximum sequence length, and dropout probabilities, enabling the model to achieve a better balance between fitting capability and generalization performance.

In this study, when the computational costs of experiments conducted on virtual servers (Google Colab) were compared—ranging from data preprocessing to classical ML, DL and hybrid/ensemble architectures. It was found that the preprocessing stages had a negligible duration, while each transformer-based model incurred a computational cost of approximately 5 hours for training. Additionally, the hyperparameter optimization process performed on the

Table 10. Hyperparameter tuning values

Hyperparameter	Search space	Type	Optimal value (selected)
Learning rate	1×10^{-6} – 5×10^{-5} (log-uniform)	Continuous	1.7048×10^{-5}
Batch size	{8, 16, 32}	Categorical	16
Number of epochs	{3, 4, 5}	Categorical	4
Max sequence length	{128, 192, 256}	Categorical	256
Weight decay	0–0.1 (log-uniform)	Continuous	0.0611
Warm-up ratio	0–0.1 (uniform)	Continuous	0.074
Hidden dropout	0–0.3 (uniform)	Continuous	0.039
Attention dropout	0–0.3 (uniform)	Continuous	0.152
Optimizer	AdamW	Fixed	AdamW
Loss function	Binary cross-entropy	Fixed	Binary cross-entropy
Evaluation metric	Weighted F1-score	Fixed	Weighted F1-score

ELECTRA model was the most costly stage and required approximately 6 to 8 hours of additional processing time (experiments with other models were not included into these calculations because of failure/missing results). On the other hand, while the CLS embedding extraction and experiments conducted with five different ML models in the Electra-based hybrid approach on the combined dataset were completed in nearly 1 hour, testing the ML methods as a hybrid took 10 minutes and the stacking ensemble structure required an additional 30 minutes of computation time. Overall, results indicate that while transformer-based models provide high accuracy, they require significant computational cost; conversely, hybrid and classical methods can offer competitive performance with much lower resource consumption.

DISCUSSION

This study presented a comprehensive comparison of conventional machine learning algorithms, transformer-based language models, and hybrid and ensemble learning frameworks for binary text classification on small and initially imbalanced datasets. Unlike previous studies that relied primarily on metadata features or evaluated a limited number of models, this work systematically investigated the contribution of data preprocessing, dataset balancing, transformer-based representation learning, and hybrid learning strategies within a unified experimental framework. The results demonstrate that each stage of the proposed workflow contributed to improving the robustness and reliability of the classification models.

The experimental findings show that preprocessing decisions substantially influenced classification performance. Replacing Label Encoding with One-Hot Encoding provided a more appropriate representation for categorical variables. Furthermore, incorporating the description feature significantly improved the model's ability to capture semantic information contained in textual data, resulting in higher recall and weighted F1-score values. These findings confirm that combining structured metadata with textual information produces more informative feature representations than relying on metadata alone.

The comparative evaluation of transformer architectures demonstrated that pretrained language models can effectively

exploit contextual semantic information for binary text classification. Among the evaluated architectures, ELECTRA consistently achieved the best overall performance and was therefore selected as the backbone of the proposed hybrid framework. Rather than using the conventional transformer classification head alone, the contextual embeddings generated by ELECTRA were successfully integrated with conventional machine learning classifiers. The results further demonstrated that the ELECTRA+random forest hybrid model outperformed the standalone transformer model, while the Stacking ensemble achieved the highest overall performance, indicating that combining complementary learning strategies provides a consistent improvement over individual model.

Another important finding concerns dataset balancing. Instead of relying solely on oversampling or downsampling techniques, a second publicly available dataset with similar characteristics was incorporated to enrich the training data and reduce class imbalance. The merged dataset (DF3) consistently produced better classification performance than the individual datasets and improved model generalization. In addition, reporting class-wise precision, recall, and F1-score enabled a more comprehensive evaluation of classifier behaviour for both the majority and minority classes, providing a more reliable assessment than overall accuracy alone.

Limitations

Although the proposed framework achieved competitive classification performance, several limitations should be acknowledged. First, despite dataset enrichment, the overall training corpus remains relatively small compared with the large-scale datasets typically used to fully exploit transformer architectures. Second, only English-language Netflix content was considered, which may limit the generalizability of the findings to multilingual or domain-specific datasets. Finally, the computational cost of transformer fine-tuning remains substantially higher than that of conventional machine learning models, although the proposed hybrid framework considerably reduces the additional computational overhead after feature extraction.

CONCLUSION

Future research may extend this work by evaluating the proposed framework on larger and multilingual text collections, investigating more recent large language models and encoder architectures. Furthermore, adaptive ensemble strategies and multimodal learning approaches that jointly utilize textual and non-textual information represent promising directions for improving binary text classification performance.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: İŞ, ST; Design: İŞ, ST; Control: İŞ, ST; Resources: İŞ, ST; Materials: İŞ, ST; Data Collection and/or Processing: İŞ, ST; Analysis and/or Interpretation: İŞ, ST; Literature Review: İŞ, ST; Writing the Article: İŞ, ST; Critical Review: İŞ, ST.

REFERENCES

- Aburass, S., Dorgham, O., & Abu Rumman, M. (2024). An ensemble approach to question classification: integrating ELECTRA transformer, GloVe, and LSTM. *International Journal of Advanced Computer Science and Applications*, 15(1), 507-514. <https://doi.org/10.14569/IJACSA.2024.0150148>
- Agrawal, R., Gupta, T., Gupta, S., Chauhan, S., Patel, P., & Hamdare, S. (2025). Fostering trust and interpretability: integrating explainable AI (XAI) with machine learning for enhanced disease prediction and decision transparency. *Diagnostic Pathology*, 20, 105. <https://doi.org/10.1186/s13000-025-01686-3>
- Behmanesh, S., Talebpour, A., Shamsfard, M., & Jafari, M. M. (2023). Improved relation span detection in question answering systems over extracted knowledge bases. *Expert Systems with Applications*, 224, 119973. <https://doi.org/10.1016/j.eswa.2023.119973>
- Biswas, D., Byun, T. Y., & Gil, J. M. (2025). Research paper classification based on CNN and BiLSTM models utilizing word embedding methods. *Human-Centric Computing and Information Sciences*, 15, 24. <https://doi.org/10.22967/HICIS.2025.15.024>
- Chaudhary, T. (2024). Humorous and professional: a dive into social media text classification. Master's thesis, International Institute of Information Technology Hyderabad.
- Das, S. D., Vadi, Y. A., & Yadav, K. (2024). Transformer-based joint modelling for automatic essay scoring and off-topic detection. In Calzolari, N., Kan, M. Y., Hoste, V., Lenci, A., Sakti, S., & Xue, N. (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 16751-16761).
- Filali, A., Shorfuzzaman, M., Abdellaoui Alaoui, E. A., Merras, M., Es-Sabry, M., Berrajaa, A., ..., & Yousef, A. (2026). Cross-lingual SMS spam detection using GAN-based augmentation for imbalanced datasets. *Scientific Reports*, 16, 7128. <https://doi.org/10.1038/s41598-026-37769-4>
- Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: from text to predictions. *Information*, 13(2), 83. <https://doi.org/10.3390/info13020083>
- Krishnan, A. (2023). Exploring machine learning and transformer-based approaches for deceptive text classification: a comparative analysis. *Inteligencia Artificial*, 20(59), 123-127.
- Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ..., & He, L. (2022). A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2), 1-41. <https://doi.org/10.1145/3495162>
- Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Medical Research Methodology*, 22, 181. <https://doi.org/10.1186/s12874-022-01665-y>
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: a comprehensive review. *ACM Computing Surveys*, 54(3), 1-40. <https://doi.org/10.1145/3439726>
- Mohammed, A., & Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 8825-8837. <https://doi.org/10.1016/j.jksuci.2021.11.001>
- Mu, Y., Jin, M., Song, X., & Aletras, N. (2024). Enhancing data quality through simple de-duplication: navigating responsible computational social science research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 12477-12492). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.emnlp-main.694>

- Nilla, A., & Setiawan, E. B. (2024). Film recommendation system using content-based filtering and the convolutional neural network (CNN) classification methods. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 10(1), 17-29. <https://doi.org/10.26555/jiteki.v10i1.28113>
- Pham, P., Nguyen, L. T. T., Pedrycz, W., & Vo, B. (2023). Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artificial Intelligence Review*, 56(6), 4893-4927. <https://doi.org/10.1007/s10462-022-10265-7>
- Rizvi, S. M. H., Imran, R., & Mahmood, A. (2025). Text classification using graph convolutional networks: a comprehensive survey. *ACM Computing Surveys*, 57(8), 1-38. <https://doi.org/10.1145/3714456>
- Shaukat, F., Ejaz, N., Ashraf, Z., Alnfai, M. M., Alotaibi, N. N., & Alnefaie, S. M. M. (2025). An interpretable multi-transformer ensemble for text-based movie genre classification. *PeerJ Computer Science*, 11, e2945. <https://doi.org/10.7717/peerj-cs.2945>
- Shaukat, F., Ejaz, N., Kamal, R., Alkhalifah, T., Aslam, S., & Mu, M. (2025). Multi-label movie genre classification with attention mechanism on movie plots. *Computers, Materials & Continua*, 83(3), 5595-5622. <https://doi.org/10.32604/cmc.2025.061702>
- Sy, C. Y., Maceda, L. L., Canon, M. J. P., & Flores, N. M. (2024). Beyond BERT: exploring the efficacy of RoBERTa and ALBERT in supervised multiclass text classification. *International Journal of Advanced Computer Science and Applications*, 15(3), 223-233. <https://doi.org/10.14569/IJACSA.2024.0150323>
- Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024). A comprehensive survey of text classification techniques and their research applications: observational and experimental insights. *Computer Science Review*, 54, 100664. <https://doi.org/10.1016/j.cosrev.2024.100664>
- Tahir, T., Maqbool, K. Q., Zafar, S., Hasan, M. Z., Hussain, M. Z., & Yaqub, M. A. (2025). A global analysis of Netflix content production: unveiling dominant countries and industry trends. *Dialogue Social Science Review*, 2(3), 519-534.
- Timoneda, J. C., & Vallejo Vera, S. (2025). BERT, RoBERTa, or DeBERTa? Comparing performance across transformers models in political science text. *The Journal of Politics*, 87(1), 347-364. <https://doi.org/10.1086/730737>
- Tuteja, M., & González Juclà, D. (2023). Long text classification using transformers with paragraph selection strategies. In *Proceedings of the Natural Legal Language Processing Workshop 2023* (pp. 17-24). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.nllp-1.3>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ..., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
- Wang, Z., Pang, Y., Lin, Y., & Zhu, X. (2024). Adaptable and reliable text classification using large language models. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 67-74). IEEE. <https://doi.org/10.1109/ICDMW65004.2024.00015>
- Yao, X., Du, W., Sun, L., & Hu, B. (2025). Automatic trailer generation for movies using convolutional neural network. *Scientific Reports*, 15, 7819. <https://doi.org/10.1038/s41598-025-91084-y>
- Yildirim, S. (2024). Fine-tuning transformer-based encoder for Turkish language understanding tasks. *arXiv*. <https://doi.org/10.48550/arXiv.2401.17396>
- Yin, L., Wang, L., Cai, Z., Lu, S., Wang, R., AlSanad, A., ..., & Zheng, W. (2024). DPAL-BERT: a faster and lighter question answering model. *Computer Modeling in Engineering & Sciences*, 141(1), 771-786. <https://doi.org/10.32604/cmescs.2024.052622>
- Zeidi, F., Amasyali, M. F., & Erol, Ç. (2024). LegalTurk: optimized BERT for multi-label text classification and NER. *arXiv*. <https://doi.org/10.48550/arXiv.2407.00648>
- Zhang, M.-L., & Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10), 1338-1351. <https://doi.org/10.1109/TKDE.2006.162>
- Zhang, T., Patil, S. G., Jain, N., Shen, S., Zaharia, M., ..., & Gonzalez, J. E. (2024). RAFT: adapting language model to domain specific RAG. *arXiv*. <https://doi.org/10.48550/arXiv.2403.10131>
- Zheng, W., Lu, S., Cai, Z., Wang, R., Wang, L., & Yin, L. (2024). PAL-BERT: an improved question answering model. *Computer Modeling in Engineering & Sciences*, 139(3), 2729-2745. <https://doi.org/10.32604/cmescs.2023.046692>
- Zhu, W., Qiu, R., & Fu, Y. (2024). Comparative study on the performance of categorical variable encoders in classification and regression tasks. *arXiv*. <https://doi.org/10.48550/arXiv.2401.09682>
- Ziyaden, A., Yelenov, A., Hajiyev, F., Rustamov, S., & Pak, A. (2024). Text data augmentation and pre-trained language model for enhancing text classification of low-resource languages. *PeerJ Computer Science*, 10, e1974. <https://doi.org/10.7717/peerj-cs.1974>