RESEARCH ARTICLE

Enhancing Social Media Bot Detection with Cross-Feature Gating and Residual Learning

Abdullah Khan[1], **Arooj Fatima**[1], **Ridda Jamil**[1], **Hassan Ahmed**[2,*] and **Aini Saba**[1]

[1] Department of Computer Science, University of Wah, Wah Cantt 47040, Pakistan
[2] Department of Computer Science, FAST—National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan

Abstract

The growing presence of malicious bot accounts on social media poses a threat to the authenticity of online communities, as they amplify misinformation, spread spam, and manipulate engagement. Reliable detection of these accounts is therefore essential to protect the integrity of platforms such as Instagram. This study introduces a deep learning–based detection framework built on the CrossGatedTabular (CGT) architecture, designed to learn complex patterns in user activity. To strengthen evaluation, two publicly available datasets of Instagram accounts were merged into a comprehensive benchmark representing diverse user behaviors. Natural language processing (NLP) was applied to refine textual content and metadata, enhancing the quality of feature representation. For classification, a CrossGatedTabular neural architecture is employed, which integrates cross-feature interactions, gated multilayer perceptron (MLP) layers, and regularization mechanisms to effectively capture complex patterns within the dataset. Experimental results demonstrate that the proposed approach achieves 0.9340 accuracy along with high precision, recall, and F1-scores, consistently outperforming baseline classifiers. These findings highlight the effectiveness of deep neural architectures in malicious account detection and provide a scalable solution for enhancing the trustworthiness of social media platforms.

Keywords: malicious bot detection, social media security, NLP, CrossGatedTabularV4, Instagram account classification.

1 Introduction

Automated social media accounts, or bots, have become a growing problem in today's digital environment. Their design usually imitates some human behavior but manages malicious intentions such as false dissemination of information, spamming, manipulating online trends, and distortion of public opinion [1]. Social media networks such as X (formerly Twitter), Instagram, and Reddit can be attacked due to their size, the types of information shared in real time, and the easy content-sharing environments. Bots are degrading user experiences by

Submitted: 22 July 2025
Accepted: 23 November 2025
Published: 02 January 2026

Vol. 3, **No.** 1, 2026.
10.62762/TETAI.2025.791029

***Corresponding author:**
✉ Hassan Ahmed
hassan.ahmed@nu.edu.pk

Citation
Khan, A., Fatima, A., Jamil, R., Ahmed, H., & Saba, A. (2026). Enhancing Social Media Bot Detection with Cross-Feature Gating and Residual Learning. *ICCK Transactions on Emerging Topics in Artificial Intelligence*, 3(1), 20–32.

affecting democratic processes and cybersecurity, in addition to safety in digital aspects in terms of trust, making bot detection an important research focus [2].

Early detection efforts were largely reliant on classical machine learning (ML) algorithms, mainly for handcrafted features like follower/following ratios, posting frequency, and several lexical statistics of the posted contents [3]. These techniques have shown some moderate success, but they relied heavily on manual feature engineering, limiting their adaptability to evolving attack strategies. Furthermore, the performance of many of these models is not consistent with other platforms, such that their performance degrades when applied to heterogeneous datasets [4].

Detection mechanisms for bot behavior should be self-adaptive and feature-extracting through automated processes without any kind of human intervention because of the dynamic and adversarial nature of bot behavior. Deep learning (DL) is changing the whole game by automatically learning new features from the analysis of big, high-dimensional, and complex data sources for bot detection features. Convolutional Neural Networks (CNNs) are set to find spatial and contextual patterns within texts and images, while using Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs) in modeling and capturing sequential user activities over time. Researchers have used LSTMs, and they showed promising results in remembering the interdependencies in the network's traffic [5, 6]. These architectures allow for the inclusion of multimodal data sources in the detection system, like metadata, activity logs, or textual content, thus strengthening and scaling the detection system. Additionally, the adaptability of deep learning techniques to novel attack strategies reinforces the continued detection performance in an adversary setting [7]. Transformer-based architectures have recently demonstrated outstanding performance in Natural Language Processing (NLP) tasks [8]. They provide strong contextual understanding and scale effectively to analyze large volumes of user-generated content. On the other side, Graph Neural Networks (GNNs) pose the most formidable competition for utilizing the relational structures of social networks in the identification of bot clusters, which are typically beyond the detection reach of traditional ways [9].

Moreover, despite advances in detecting bots using DL, there are still many critical challenges that need to be addressed. The presence of very small datasets in large-scale, well-organized, and labeled sources constrains the training of highly precise models. High levels of class imbalance and minority classes work in a detection bias toward the existing majority classes. In addition, the semantics of language generation are increasingly human-like, and the uses of such patterns consist of a representation of activity time distributions where abnormal activity events blend with legitimate interaction patterns, greatly reducing accuracy in detection. To meet the challenges mentioned above, detection frameworks should be developed that are accurate, accessible, and computationally efficient for real-time scalability [10].

However, there are still challenges. The behavior of bots widely differs between social media platforms. Furthermore, the availability of multi-platform annotated datasets is scarce, limiting research into cross-platform detection methods. This study fills these gaps by proposing an integrative deep learning-based bot detection framework that relies on datasets sourced from Instagram. The system analyzes account metadata (profile picture, description length, follower and following counts, posting activity, and ratios such as followers–following and engagement) and behavioral indicators (follow activity score and interaction patterns) to classify accounts as human or bot. The aim is to improve detection accuracy and generalizability while minimizing biases toward individual platforms.

The contributions of this study are as follows:

- Two publicly available Instagram datasets are merged into a unified and annotated corpus of genuine and malicious accounts, providing a reliable benchmark for bot detection.
- A systematic preprocessing pipeline with normalization, missing value handling, and feature engineering is applied, while class imbalance is mitigated through oversampling to ensure fair and effective model training.
- A CrossGatedTabular deep learning model is implemented to capture gated feature interactions, achieving robust and scalable performance in distinguishing bots from real users.

Following an introduction to the general problem, Section 2 reviews the relevant literature around bot detection, emphasizing the evolution from traditional methods to the modern deep learning approach. Section 3 describes the proposed methodology that incorporates dataset preprocessing, feature extraction, and model architecture. Section 4 presents the results and analyses of the experiments. The section 5 presents the conclusions and future work of this study.

2 Related Work

Ellaky et al. [11] proposed a hybrid DL architecture combining GloVe word embeddings with BiGRU and LSTM networks was proposed for social media bot detection. Trained on the Cresci-2017 and TwiBot-20 datasets, the model achieved exceptional performance, particularly on TwiBot-20, with 100% precision, 99.73% accuracy, 99.56% recall, and 99.63% F1-score, outperforming existing methods. The study also addressed issues such as class imbalance and generalizability, suggesting directions for multilingual detection and incorporating features beyond text content. Terumalasetti et al. [12], developed a comprehensive framework named Visual Graph Temporal Network (VGTN), which integrates visual, temporal, and network features for detecting fake accounts. Evaluated on the Cresci-2017 dataset, it outperformed conventional methods across key metrics and discussed practical deployment challenges, including bias mitigation and privacy concerns.

Sallah et al. [13] introduced a transformer-based approach for Twitter bot detection, leveraging pre-trained language models such as BERT, RoBERTa, and GPT-3 to generate rich text embeddings for a deep neural network classifier. Tested on TweepFake and fox8-23 datasets, it achieved state-of-the-art F1-scores of 90.29% and 93%, respectively, and incorporated explainable AI techniques like SHAP and LLM-based prompting for model interpretability. Zeng et al. [14] presented ESA-BotRGCN. This emoji-driven multimodal detection framework combined emoji-text mapping, sentiment analysis, and relational graph convolutional networks to capture semantic, emotional, and topological features. On the TwiBot-20 and Cresci-15 datasets, ESA-BotRGCN achieved an accuracy of 87.46%, though it was limited to text-based features, with future work suggesting the integration of image and audio modalities.

Swathi et al. [15] proposed RL-RNN as a hybrid of Reinforcement Learning and Recurrent Neural Networks, focusing on URL-based and NLP-derived features for malicious bot detection on Twitter. The RL framework optimized feature selection, while the RNN component identified behavioral patterns, outperforming SVM in accuracy, precision, and recall, though lacking reported quantitative results and limitation analysis. Mohammadi et al. [16] introduced a mobile botnet detection system using machine learning algorithms alongside a golden ratio evolutionary algorithm for feature selection and dimensionality reduction. Evaluated on NSL-KDD, Drebin, and ISCX datasets, the method achieved the highest accuracy with random forest, with decision tree and k-nearest neighbor also performing well. Feature selection reduced training time, although generalization to unseen data was not assessed.

Arranz-Escudero et al. [17] proposed a multimodal detection framework, TMTM, integrating user profile features, text analysis, and graph-based techniques. Using the TwiBot-22 dataset, it improved detection accuracy by 5.48% over existing models and identified key parameters for optimization. Limitations included computational resource demands and restricted exploration of graph relationships. Guyan et al. [18] developed the PEGNN framework to exploit both central and peripheral network information for bot detection by reframing the task as central node classification and introducing three loss functions to fuse multi-network features. Tested on TwiBot-20 and TwiBot-22, it achieved notable accuracy and F1-score improvements with minimal computational overhead.

Zhou et al. [19] proposed the LGB framework, integrating language models with graph neural networks to detect social bots, particularly those with isolated or sparsely linked nodes. Evaluated on TwiBot-22 and TwiBot-20, it outperformed state-of-the-art baselines and incorporated an online smart feedback system for continuous adaptation to new bot behaviors. Key contributions included its multimodal design, comprehensive experimental validation, and mechanisms to address the evolving nature of social bots. Finally, Ghosh et al. [20] proposed a machine learning-based approach for detecting inorganic accounts by extracting temporal and semantic features, such as periodicity, ARIMA-based metrics, lexical diversity, and sentiment. Using 470 organic and 373 inorganic users from the Bot Repository, the study showed that SVM outperformed K-Means in accuracy, specificity, and sensitivity. The work also applied Split Conformal Prediction for uncertainty quantification, noting limitations in relying on historical data and the need for adaptive real-time detection.

Lopez-Joya et al. [21] have emphasized the importance of integrating both account-based and content-based attributes for more accurate bot identification. They achieved better performance from an extensive collection of newly suggested and already existing features through Random Forest classifier-based feature selection and integration across several datasets. On Cresci-15, their approach achieved 0.996 accuracy, 0.993 precision, 0.995 recall, and 0.994 F1-score; on Cresci-17, they achieved 0.994 accuracy, 0.987 precision, 0.990 recall, and 0.998 F1-score; and on TwiBot-20, they achieved 0.85 accuracy, 0.832 precision, 0.936 recall, and 0.870 F1-score.

Using the CRESCI-2017 dataset, Wei et al. [22] used a model developed using deep learning called BiLSTM (Bi-directional long, short-term memory) to classify the bot accounts and human accounts, With 96% accuracy, 96.3% F-measure, and 92% MCC for Test 1 and 92.9% accuracy, 92.6% F-measure, and 85.7% for Test 2. Mounika et al. [23] utilized Random Forest classification to examine various factors, namely account age, posting frequency, follower-to-following ratio, and content patterns. The approach also combined explainable techniques with privacy-preserving measures to ensure availability and compliance with the rules. Based on simulations, it outperforms Decision Tree and SVM with 96.8% accuracy, 95.7% precision, 96.2% recall, and a 95.9% F1-score, providing a very efficient and instant method for identifying fraudulent accounts on social media.

Thavasimani et al. [24] proposed that the Custom Genetic Algorithm (CGA) finds the optimal settings for deep learning models, thereby drastically reducing search time by tweaking hyperparameters. The tuned models, through CGA, outperform the existing methods and easily adapt to different classification datasets. They proved the flexibility and efficiency of the method with a 97.8% accuracy, 99.4% precision, 98.2% recall, and 98.8% F1-score on the Cresci-17 dataset. Deshmukh et al. [25] have made a good contribution towards formulating a bot identification platform by proposing GraphSAGE-BERT with SVM, which encompasses the deep contextual knowledge derived from BERT and the machine learning capabilities of GraphSAGE. The algorithm detects complex text and behavioral patterns of account information by merging embeddings from each model into one single feature vector. And the final categorization is done by an SVM classifier. The algorithm achieves 74.62% on the TwiBot-22 and 98.68% on the Cresci-17 dataset.

The social bot detection influence of word embeddings with neural network algorithms was investigated using the Cresci-17 dataset and response attributes exclusively by Ellaky et al. [26]. The techniques used to test the various embeddings, BOW, TF-IDF, Doc2Vec, BERT, Word2Vec, and FastText, included Logistic Regression, Decision Trees, K-NN, SVM, Random Forest, Naive Bayes, Adaboost, XGBoost, and MLP. Doc2Vec has given a perfect score of 100% precision, while Decision Tree and Random Forest have done quite well in terms of producing high-level precision, i.e., 99.96% with the use of BERT. Javed et al. [27] proposed an interpretable machine learning framework for enhancing social network bot identification via multiple methods with cross-validation tuning. The model has an expressive result interpretation of categorization and indicates which factors interfere with SHAP and LIME in the predictions. Tests on the Cresci-15 and Cresci-17 datasets produced F1 scores of 99.3% and accuracies of 99.1% and 99.0%, respectively.

Di Paolo et al. [28] proposed BotHash, which is a training-free technique that employs estimated nearest-neighbour searching and reduced user representations to identify social bots. Even in cases when postings are produced by sophisticated LLMs, this eliminates the difficulties of deep learning training and the compilation of big datasets while still differentiating between bots and people. BotHash's F1-score and accuracy on benchmark datasets were 0.96 and 0.95 on Cresci-15, 0.98 and 0.97 on Cresci-17 and Cresci-18, and 0.64 and 0.61 on Twibot-22.

3 Methodology

A malicious bot detection methodology for social media is proposed, which consists of several sequential stages, namely dataset integration, preprocessing, feature engineering, data balancing, and model training. Two publicly available Instagram datasets are merged into a unified Dataset containing annotated user accounts categorized as bots or genuine users, which serves as the foundation for subsequent analysis [29]. The textual content and associated metadata are subjected to NLP preprocessing techniques, including tokenization, stop-word elimination, and vectorization, to ensure a consistent and semantically meaningful representation of the data [17]. Furthermore, feature engineering is conducted to extract discriminative attributes such as posting frequency, followers–following ratio, engagement metrics, and content-based characteristics, thereby enhancing the learning capability of the detection model [30].

To address the inherent issue of class imbalance

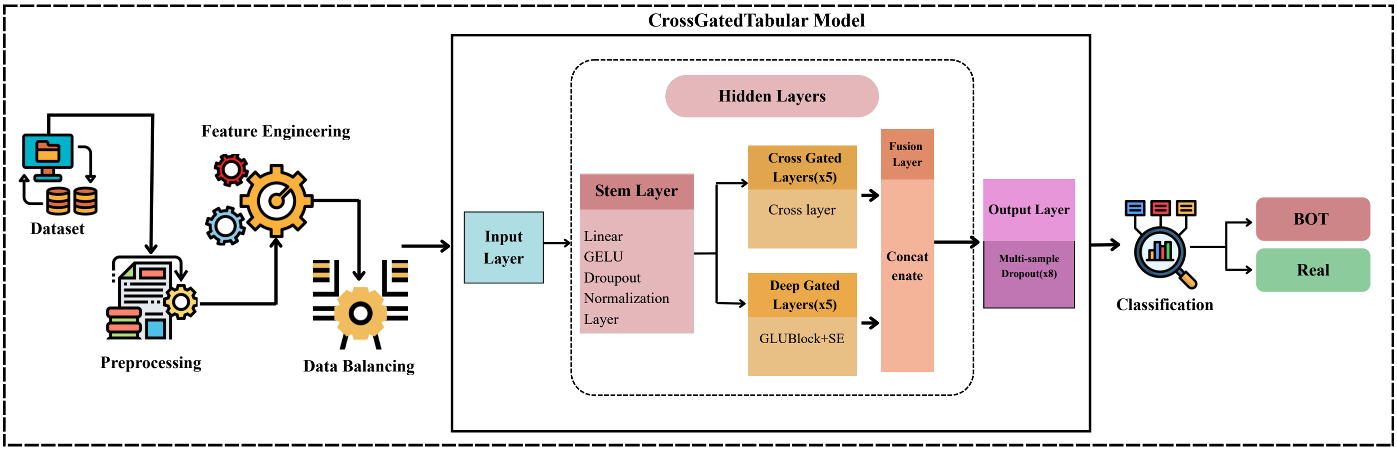

Figure 1. Architecture of the proposed methodology illustrating the sequential steps of the bot detection framework.

in bot detection datasets, the Synthetic Minority Oversampling Technique (SMOTE) is employed to generate synthetic samples of the minority class, ensuring a balanced distribution of instances and mitigating bias during training [31]. Subsequently, multiple DL algorithms are implemented to classify accounts as either malicious bots or legitimate users. This hybrid pipeline leverages both data-centric preprocessing and model-driven classification to achieve robust and scalable detection. The overall framework, as illustrated in Figure 1, outlines the integration of preprocessing, feature extraction, balancing, and classification, providing a systematic approach for detecting malicious bots on Instagram.

3.1 Preprocessing

Preprocessing is the first step to ensure consistency across the heterogeneous Instagram dataset, which contains diverse features such as follower counts, posting frequency, and engagement ratios. Without preprocessing, features with larger ranges can dominate smaller-scale attributes, which may lead to biased learning and unstable convergence.

To address this, normalization and standardization are applied. Standardization ensures each feature has zero mean and unit variance as shown in Equation 1. Normalization rescales features into the $[0, 1]$ range, using min-max scaling, as shown in Equation 2.

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{1}$$

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

These transformations harmonize the feature space, reduce the effect of outliers, and stabilize training. In malicious account detection, this is crucial to prevent skewed learning from extreme cases such as celebrity accounts (very high followers) or bots (very low engagement).

3.2 Feature Engineering

After preprocessing, feature engineering is applied to enhance the discriminative power of the dataset. The approach includes constructing second-order polynomial interaction features, which involve generating all pairwise products of the base features. However, generating all interactions can drastically increase dimensionality. To avoid noise and redundancy, the interaction features are evaluated using point-biserial correlation with the class labels. Only the top-ranked interactions with the highest discriminative strength are retained. Finally, these selected interaction terms are concatenated with the original features, producing an enriched dataset that captures both base-level signals and interaction-based patterns. This balance enhances expressiveness without unnecessary computational overhead, strengthening the foundation for malicious account classification.

3.3 Data Balancing

One of the key challenges in bot detection is the class imbalance problem, where malicious accounts are far fewer than legitimate ones. Training on such imbalanced data biases the model toward the majority class, leading to poor detection of minority (bot) accounts. To mitigate the class imbalance problem, SMOTE is applied. Unlike simple oversampling, which duplicates existing samples, SMOTE generates synthetic examples by interpolating between real minority instances in the feature space using the Equation 3.

$$x_{\text{new}} = x_i + \lambda \cdot (x_j - x_i), \quad \lambda \in [0, 1] \tag{3}$$

This creates new, realistic samples that preserve the underlying distribution of malicious accounts. By balancing the dataset after preprocessing and feature engineering, SMOTE ensures that the classifier receives equal exposure to both Real and Bot accounts, improving generalization and robustness. Only balancing was applied to the training subset, while the validation and test subsets were left in their natural distributions. This approach prevents bias in evaluation and guarantees that model performance is assessed under realistic conditions.

3.4 CrossGatedTabular Model Architecture

The CrossGatedTabular model is a specialized DL architecture designed to effectively detect malicious accounts in tabular social media datasets. Traditional classifiers, such as standard Multi-Layer Perceptrons, often struggle to capture the diverse and subtle patterns present in user behavior. These patterns include both direct relationships between features, such as follower counts and engagement ratios, and complex nonlinear dependencies, such as interactions between posting frequency, content type, and follower engagement. To address these challenges, the CrossGatedTabular model integrates cross-feature interaction modeling with gated residual learning, enabling it to capture both explicit multiplicative dependencies and deep hierarchical representations while suppressing redundant signals through selective gating mechanisms. This dual approach makes the model particularly suited for detecting subtle behavioral anomalies indicative of fake or bot accounts, even in highly imbalanced and noisy datasets.

3.4.1 Input Layer

The network begins with an input layer that receives the preprocessed feature vector $\mathbf{x} \in \mathbb{R}^d$, where d represents the number of input features. Each feature is projected into a higher-dimensional latent space using a linear transformation followed by a non-linear GELU activation function which is defined in Equation 4.

$$\mathbf{h}_0 = \text{GELU}(\mathbf{W}_0\mathbf{x} + \mathbf{b}_0) \tag{4}$$

This initial projection allows the network to operate in a richer latent space where both linear and nonlinear interactions can be more effectively modeled.

3.4.2 Hidden Layers

The model contains two parallel processing streams: a *cross tower* and a *deep tower*.

Cross Tower: The cross tower explicitly models feature interactions by updating each layer based on both the original input embedding and the current layer representation as defined in Equation 5. Each layer combines the original input with a learned transformation of the previous layer, enabling the network to capture higher-order feature interactions efficiently.

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \text{DropPath}\Big(\text{LayerScale}\big(\mathbf{W}_2\,\text{GELU}(\mathbf{W}_1(\mathbf{x}_0 \odot (\mathbf{W}_0\mathbf{x}_l)) + \mathbf{b})\big)\Big) \tag{5}$$

where $\odot$ denotes element-wise multiplication, which enables multiplicative feature interactions. LayerScale stabilizes deeper networks by rescaling outputs per channel, and DropPath introduces stochastic depth to regularize the network and prevent overfitting.

Deep Tower: In parallel, the deep tower processes the original embedding $\mathbf{x}_0$ through a sequence of Gated Linear Unit (GLU) residual blocks with Squeeze-and-Excitation (SE) mechanisms as defined in Equations 6 and 7.

$$\mathbf{y} = \mathbf{a} \odot \sigma(\mathbf{g}), \quad [\mathbf{a}, \mathbf{g}] = \text{Linear}(\text{LayerNorm}(\mathbf{x})) \tag{6}$$

$$\mathbf{x}_{\text{deep}} = \mathbf{x} + \text{DropPath}(\text{LayerScale}(\text{SE}(\mathbf{y}))) \tag{7}$$

The SE block reweights feature channels based on importance, while the gating function $\sigma(\cdot)$ ensures that only the most relevant information is propagated forward.

3.4.3 Fusion Layer

After processing through the cross and deep towers, the outputs are fused into a joint representation as shown in Equation 8, combining complementary information from both streams.

$$\mathbf{z} = \text{Fusion}([\mathbf{x}_{\text{cross}}, \mathbf{x}_{\text{deep}}]) = \text{GELU}(\mathbf{W}_f[\mathbf{x}_{\text{cross}}; \mathbf{x}_{\text{deep}}] + \mathbf{b}_f) \tag{8}$$

This fusion layer leverages complementary information from both streams, enhancing the representational capacity while maintaining computational efficiency.

3.4.4 Output Layer

The classification head applies a multi-sample dropout strategy for improved generalization. Multiple dropout masks are applied to the fused representation, producing several logits that are averaged, as shown in the Equation 9.

$$\hat{y} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{W}_{\text{out}}\,\text{Dropout}_m(\mathbf{z}) + \mathbf{b}_{\text{out}}\right)\right) \tag{9}$$

where M is the number of dropout samples, and σ is the sigmoid activation function for binary classification. This technique reduces overfitting and ensures more robust predictions, particularly in noisy or imbalanced datasets. The CrossGatedTabular model integrates cross-feature interactions, gated residual learning, and multi-sample dropout to effectively capture complex patterns in tabular social media data. Its dual-tower, fused architecture enables the model to effectively capture both linear and nonlinear feature interactions, improving robustness and generalization in detecting malicious accounts.

4 Results & Discussion

The proposed hybrid framework was evaluated on the classification of malicious bot detection using the unified Instagram dataset. The dataset was categorized into binary classes, real users, and bots. The dataset is carefully partitioned into training, validation, and testing subsets to ensure unbiased performance assessment. Experimental findings demonstrate that the model consistently achieves strong classification performance, even under challenging conditions of class imbalance, thereby validating the robustness and reliability of the approach.

4.1 Experimental Setup

The experiments were primarily executed in the Kaggle environment with GPU acceleration provided by an NVIDIA Tesla T4 (16 GB VRAM). Preliminary preprocessing and dataset preparation were carried out on a local machine equipped with an Intel® Core™ i5-9600K CPU and 16 GB RAM. The CrossGatedTabular model was implemented in PyTorch under stratified evaluation, with oversampling applied only to the training data, while validation and test sets were preserved in their original distribution.

4.2 Data Acquisition

The dataset employed in this study was obtained by merging two publicly available Instagram datasets, InstaFake [32] from GitHub and IFSG [33] from the Kaggle repository. These datasets were curated to address the detection of real and bot accounts on social media. After resolving redundancy and overlap, the final merged dataset comprised 1,890 entities, including 1,342 real accounts and 548 bot accounts, thereby supporting the binary classification task.

The class distribution of the dataset is illustrated in Figure 2, which highlights the inherent imbalance between real and bot accounts.

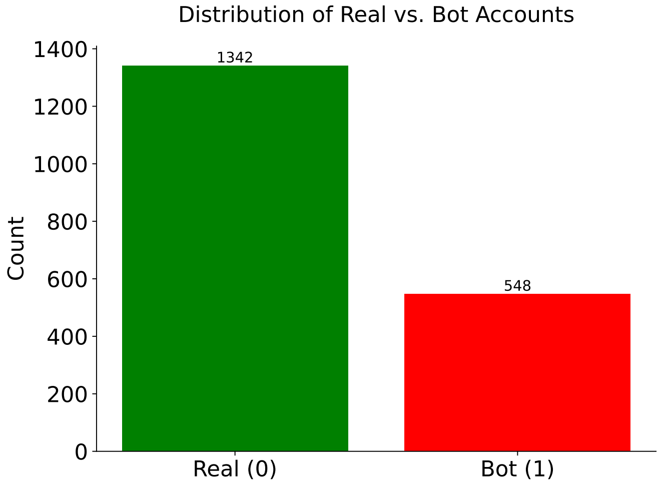

Figure 2. Distribution of samples across real and bot classes.

Table 1. Dataset distribution before and after balancing.

Split	Pre-balance			Post-balance		
	Real (0)	Bot (1)	Total	Real (0)	Bot (1)	Total
Training	872	356	1228	872	872	1744
Validation	201	82	283	201	82	283
Test	269	110	379	269	110	379
All samples	**1342**	**548**	**1890**	**1342**	**1064**	**2406**

The dataset was organized into a structured tabular format with features such as follower and following counts, posting frequency, and engagement statistics. The pre-balancing and post-balancing split distributions are detailed in Table 1. This distribution provided a reliable foundation for subsequent preprocessing and model evaluation.

4.3 Proposed Model Results

In this section, the evaluation of the proposed model is presented using classification report, confusion matrix, accuracy, and loss metrics. Classification performance was measured using precision, recall, F1-score, and accuracy. The model achieved an overall accuracy of 0.9340 on the test set. , as summarized in Table 2.

For Real users (class 0), the model obtained a precision of 0.9552, a recall of 0.9517, and an F1-score of

Table 2. Classification report on Test Set.

Class	Precision	Recall	F1-score	Support
0 (Real)	0.9552	0.9517	0.9534	269
1 (Bot)	0.8829	0.8909	0.8869	110
Accuracy			0.9340	379
Macro Avg	0.9191	0.9213	0.9202	379
Weighted Avg	0.9342	0.9340	0.9341	379

0.9534, indicating reliable classification performance. For bots (class 1), the model reached a precision of 0.8829, a recall of 0.8909, and an F1-score of 0.8869. While slightly lower than the performance on class 0, these values demonstrate the model's ability to detect malicious activity effectively despite class imbalance. The macro-averaged F1-score of 0.9202 and the weighted F1-score of 0.9341 further confirm the robustness of the proposed approach across both classes.

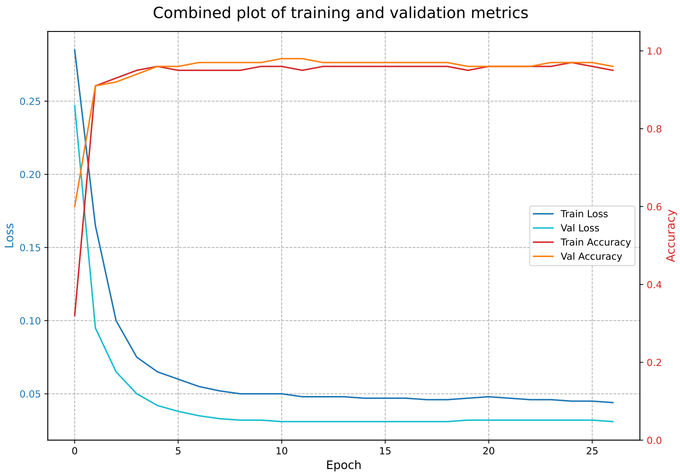

Figure 3. Convergence of model training, plotting loss (left axis) against accuracy (right axis) for both training and validation datasets.

Figure 3 illustrates the model's combined performance, plotting the training and validation accuracy curves (right y-axis) against the corresponding loss curves (left y-axis) over 27 epochs. The results indicate that training accuracy steadily increases over epochs, and validation accuracy follows a similar trend, stabilizing near 0.9647, which reflects strong generalization capability. Both training and validation losses decrease steadily, confirming that the model converges toward an optimal solution without signs of overfitting.

In addition to accuracy and loss trends, the correlation heatmap Figure 4 shows that most features have low pairwise correlation, with only moderate correlations between raw and logarithmic transformations. This indicates diverse, non-redundant attributes that support model generalization and reinforce its robustness.

In Figure 5, the confusion matrix evaluates the classification performance of the proposed model, showing a clear comparison between predicted and actual labels. The model achieves high accuracy across both classes, with most samples correctly classified as true negatives and true positives. Only a small fraction fall into false positives and false negatives, demonstrating strong generalization and reliable detection.

In Figure 6, the receiver operating characteristic (ROC) curve is presented, which plots the true positive rate against the false positive rate across different classification thresholds. The ROC provides an effective visualization of the model's ability to discriminate between the two classes. The area under the curve (AUC) achieved is 0.9830, indicating that the model possesses a very strong discrimination capability. This high AUC demonstrates that the proposed architecture is able to maintain both high sensitivity and specificity, confirming its robustness and reliability in distinguishing between the categories.

4.4 Ablation Study

Table 3 summarizes the ablation study analyzing the influence of architectural depth and batch size on the CrossGatedTabularV4 model. The baseline configuration (Model 1), with five cross layers, six deep layers, a hidden size of 384, and a batch size of 64, achieved an accuracy of 0.9200. Reducing the layer depth while employing a very large batch size of 512 (Model 2) slightly improved accuracy to 0.9288, though the reduced representational capacity constrained further gains. Expanding the hidden dimension to 512 while maintaining the original depth (Model 3) enhanced feature representation, leading to an accuracy of 0.9314. The proposed model, which preserved the deeper configuration but adopted a balanced batch size of 32, achieved the best performance with an accuracy of 0.9340. These results demonstrate that both depth and batch size significantly impact generalization, and that the proposed configuration provides the most effective trade-off.

Table 3 presents the ablation study on the effect of depth and batch size in the CrossGatedTabularV4 model. The baseline (Model 1) with five cross layers, six deep layers, hidden size 384, and batch size 64 achieved 0.9200 accuracy. Using fewer layers with a large batch size of 512 (Model 2) slightly improved accuracy to 0.9288, though limited by

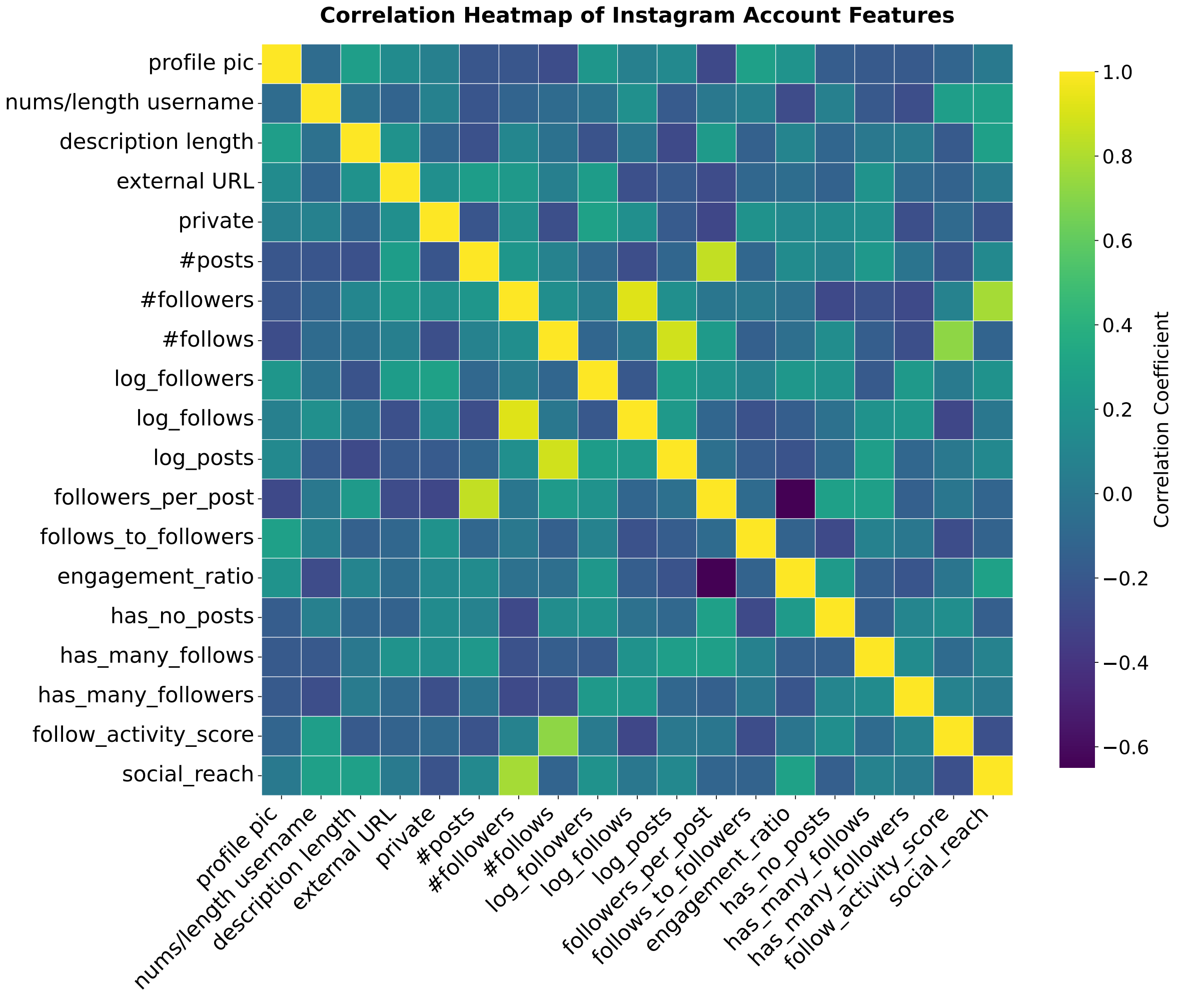

Figure 4. Correlation heatmap visualizing the relationships between features.

reduced representational capacity. Expanding the hidden dimension to 512 (Model 3) improved accuracy to 0.9314. The proposed configuration, with deeper layers and a balanced batch size of 32, achieved the best result of 0.9340. This highlights the critical role of depth and batch size in optimizing generalization.

Table 3. Ablation study on model depth, hidden dimension, and batch size.

Model	n_{cross}	n_{deep}	Hidden Layers	Batch	Results
1	5	6	384	64	0.9200
2	4	5	256	8	0.9288
3	5	6	512	128	0.9314
Proposed	**5**	**6**	**512**	**32**	**0.9340**

4.5 Comparison with Existing Studies

Table 4 highlights the comparative performance of existing methods and the proposed model across benchmark datasets for Instagram fake-account detection. Prior work on the IFSG dataset demonstrates that Gradient Boosting consistently yields strong results, with accuracies of 0.9312 [34] and 0.9240 [35]. In contrast, approaches relying primarily on textual signals, such as the DistilBERT+RF pipeline evaluated on the InstaFake dataset, achieve lower performance, reaching only 0.8384 accuracy [36]. More recently, [37] evaluated Random Forest, KNN, and Linear SVC models on the IFSG dataset, reporting that Random Forest achieved the highest accuracy of 0.925, followed by KNN at 0.875 and Linear SVC at 0.683.

The proposed model, trained on the merged IFSG+InstaFake datasets, achieves 0.9340 accuracy and 0.9830 ROC-AUC, surpassing both IFSG-only and InstaFake baselines while maintaining robustness under dataset integration. This indicates that leveraging heterogeneous benchmarks mitigates dataset-specific bias, thereby improving the model's practical applicability for real-world malicious account detection.

Furthermore, the comparison underscores the advantage of profile-centric, tabular representations

Table 4. Comparison of the proposed model against existing approaches.

Reference	Year	Dataset	Methods	Accuracy
Al-Amin et al. [34]	2024	IFSG	Gradient Boosting	93.12%
Gunawan et al. [35]	2025	IFSG	Gradient Boosting	92.4%
Goyal et al. [36]	2024	InstaFake	DistilBERT+RF	83.84%
Arunprakaash et al. [37]	2024	IFSG	Random Forest	92.5%
Proposed	**2025**	**(InstaFake+IFSG)**	**CrossGatedTabular**	**93.40%**

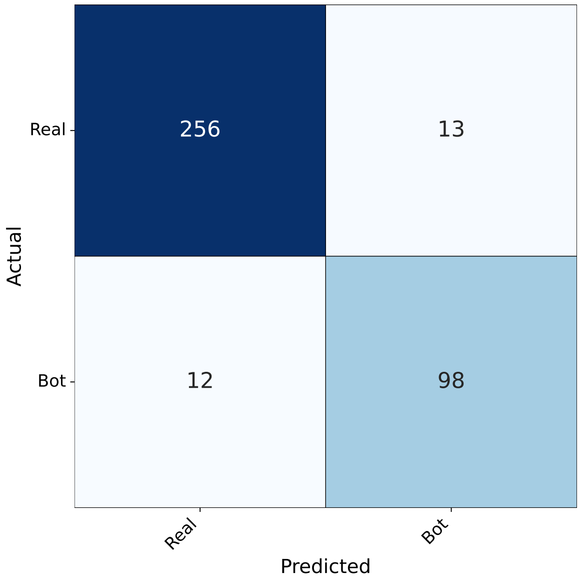

Figure 5. Confusion matrix illustrating the proposed model's performance.

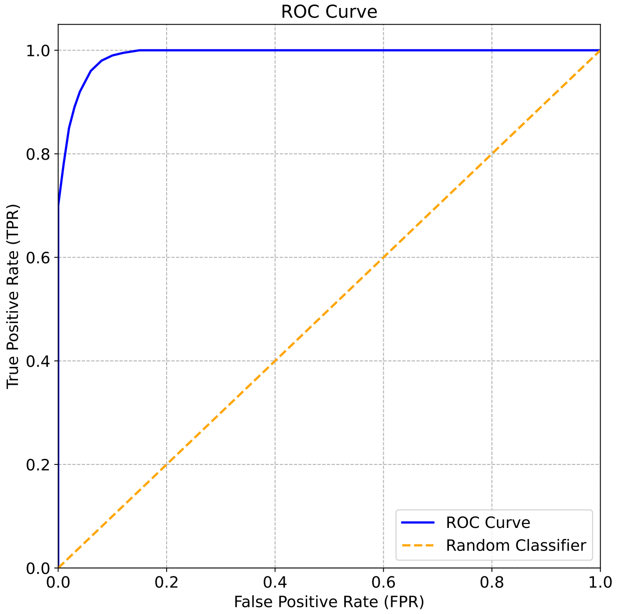

Figure 6. Receiver Operating Characteristic (ROC) curve for the proposed model for real and bot classification.

over text-only approaches for fake-account detection. While text-based pipelines may capture linguistic cues, profile and activity metadata provide stronger, more generalizable signals for discriminating between legitimate and malicious accounts. Prior studies on Instagram fake-account detection show that Gradient Boosting achieves strong performance on the IFSG dataset (0.9312 [34], 0.9240 [35]), while text-focused methods like DistilBERT+RF perform lower on InstaFake (0.8384 [36]). More recent evaluations [37] reported Random Forest at 0.925, KNN at 0.875, and Linear SVC at 0.683 on IFSG. In comparison, the proposed model trained on merged IFSG+InstaFake datasets achieves 0.9340 accuracy and 0.9830 ROC-AUC, demonstrating robustness, and the advantage of profile-centric tabular features over text-only approaches.

4.6 Discussion

The superior performance of the proposed model can be attributed to its specialized architecture, which is uniquely suited to the challenges of bot detection beyond what traditional models or standard deep learning offers. While models like Gradient Boosting are effective at finding optimal decision boundaries, they can struggle to model highly complex, continuous interactions. The proposed model's cross-tower explicitly learns multiplicative feature interactions. For instance, capturing the nuanced relationship where a high follower count is only meaningful in combination with a high engagement rate. Simultaneously, the deep tower's gated mechanisms (GLU and Squeeze-and-Excitation) act as a dynamic, learned feature-selection filter. This allows the model to dynamically suppress redundant or noisy signals on a per-sample basis, a critical advantage when dealing with diverse user profiles where the importance of a feature can vary. Finally, the multi-sample dropout strategy provides robust regularization, making the model less likely to overfit to the synthetic samples generated by SMOTE and enabling it to generalize more effectively to the naturally imbalanced, unseen test data.

5 Conclusion

This research presented a deep learning–based framework for social media bot detection, leveraging cross-feature interactions and gated tabular representations to achieve strong classification performance. The proposed model attained a testing accuracy of 0.9340, with consistent improvements across precision, recall, and F1-score, demonstrating its effectiveness in distinguishing between human and automated accounts. These findings confirm the potential of advanced neural architectures in addressing the challenges of bot identification on modern platforms.

Looking forward, future work will explore the use of feature-rich and cross-platform datasets, integrating behavioral, linguistic, and network-level attributes to further strengthen detection capability. Such advancements are expected to enhance robustness against evolving adversarial strategies and enable the development of scalable, adaptable systems that contribute to safer and more trustworthy online environments.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Subrahmanian, V. S., Azaria, A., Durst, S., Kagan, V., Galstyan, A., Lerman, K., ... & Menczer, F. (2016). The DARPA Twitter bot challenge. *Computer, 49*(6), 38-46. [CrossRef]

[2] Abulaish, M., & Fazil, M. (2020). Socialbots: Impacts, threat-dimensions, and defense challenges. *IEEE Technology and Society Magazine, 39*(3), 52–61. [CrossRef]

[3] Shao, C., Ciampaglia, G. L., Varol, O., Yang, K. C., Flammini, A., & Menczer, F. (2018). The spread of low-credibility content by social bots. *Nature communications, 9*(1), 4787. [CrossRef]

[4] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016, April). Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web* (pp. 273-274). [CrossRef]

[5] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017, April). The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion* (pp. 963-972). [CrossRef]

[6] Arin, E., & Kutlu, M. (2023). Deep learning based social bot detection on twitter. *IEEE Transactions on Information Forensics and Security, 18*, 1763-1772. [CrossRef]

[7] Sayyadiharikandeh, M., Varol, O., Yang, K. C., Flammini, A., & Menczer, F. (2020, October). Detection of novel social bots by ensembles of specialized classifiers. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2725-2732). [CrossRef]

[8] 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*.

[9] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems, 32*(1), 4-24. [CrossRef]

[10] Kudugunta, S., & Ferrara, E. (2018). Deep neural networks for bot detection. *Information Sciences, 467*, 312-322. [CrossRef]

[11] Ellaky, Z., Benabbou, F., Matrane, Y., & Qaqa, S. (2024). A hybrid deep learning architecture for social media bots detection based on BiGRU-LSTM and GloVe word embedding. *IEEE Access*. [CrossRef]

[12] Terumalasetti, S., & Reeja, S. R. (2024). Enhancing social media user's trust: A comprehensive framework for detecting malicious profiles using multi-dimensional analytics. *IEEE Access*. [CrossRef]

[13] Sallah, A., Agoujil, S., Wani, M. A., Hammad, M., Maleh, Y., & Abd El-Latif, A. A. (2024). Fine-tuned understanding: Enhancing social bot detection with transformer-based classification. *IEEE Access, 12*, 118250-118269. [CrossRef]

[14] Zeng, K., Li, Z., & Wang, X. (2025). Emoji-Driven Sentiment Analysis for Social Bot Detection with Relational Graph Convolutional Networks. *Sensors, 25*(13), 4179. [CrossRef]

[15] Swathi, P., Karmakar, M., Banshal, S. K., & Moni, R. (2024, November). Malicious Social Bot Detection: RL-RNN Based Hybrid Approach. In *2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)* (pp. 1-5). IEEE. [CrossRef]

[16] Mohammadi, H., & Hosseini, S. (2025). Mobile botnet attacks detection using supervised learning algorithms. *Security and Privacy, 8*(2), e494. [CrossRef]

[17] Arranz-Escudero, O., Quijano-Sanchez, L., &

Liberatore, F. (2025). Enhancing misinformation countermeasures: a multimodal approach to twitter bot detection. *Social Network Analysis and Mining*, *15*(1), 26. [CrossRef]

[18] Guyan, Q., Liu, Y., Liu, J., & Zhang, P. (2025). PEGNN: Peripheral-Enhanced graph neural network for social bot detection. *Expert Systems with Applications*, *278*, 127294. [CrossRef]

[19] Zhou, M., Zhang, D., Wang, Y., Geng, Y., Dong, Y., & Tang, J. (2025). Lgb: Language model and graph neural network-driven social bot detection. *IEEE Transactions on Knowledge and Data Engineering*. [CrossRef]

[20] Ghosh, D., Boettcher, W., Johnston, R., & Lahiri, S. (2025). Bot Identification in Social Media. *arXiv preprint arXiv:2503.23629*.

[21] Lopez-Joya, S., Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2024). Exploring social bots: a feature-based approach to improve bot detection in social networks. *arXiv preprint arXiv:2411.06626*.

[22] Wei, F., & Nguyen, U. T. (2019). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (pp. 101–109). IEEE. [CrossRef]

[23] Mounika, K., & Reddy, N. R. (2025). An Integrated Machine Learning Framework for Spammer and Fake User Detection in Online Social Networks. *Fringe Multi-Engineering Proceedings (FMEP, ISSN: 3107-6149)*, *1*(3), 12-25. [CrossRef]

[24] Thavasimani, K., & Srinath, N. K. (2022). Optimal hyper-parameter tuning using custom genetic algorithm on deep learning to detect Twitter bots. *Journal of Engineering Science and Technology*, *17*(2), 1532–1549.

[25] Deshmukh, A., Moh, M., & Moh, T. S. (2024, December). Bot Detection in Social Media Using GraphSage and BERT. In *2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (pp. 804-811). IEEE. [CrossRef]

[26] Ellaky, Z., Benabbou, F., Ouahabi, S., & Sael, N. (2021). Word embedding for social bot detection systems. In *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)* (pp. 1–8). IEEE. [CrossRef]

[27] Javed, D., Jhanjhi, N. Z., Khan, N. A., Ray, S. K., Al-Dhaqm, A., & Kebande, V. R. (2025). Identification of Spambots and Fake Followers on Social Network via Interpretable AI-Based Machine Learning. *IEEE Access*. [CrossRef]

[28] Di Paolo, E., De Gaspari, F., & Spognardi, A. (2025). BotHash: Efficient and Training-Free Bot Detection Through Approximate Nearest Neighbor. *arXiv preprint arXiv:2506.20503*.

[29] Duman, A., & Mengutaycı, Ü. (2025). Transformer Based Approach for Instagram Fake Profile Detection. In *Proceedings of the 5th International Conference on Contemporary Academic Research (ICCAR)* (pp. 109-113). Konya, Turkey. [CrossRef]

[30] Chelas, S., Routis, G., & Roussaki, I. (2024). Detection of fake instagram accounts via machine learning techniques. *Computers*, *13*(11), 296. [CrossRef]

[31] Ellaky, Z., Benabbou, F., Bouaine, C., & Matrane, Y. (2024, October). Enhanced Multi-model Approach for Social Media Bots Recognition Systems Using Imbalanced Dataset. In *The Proceedings of the International Conference on Smart City Applications* (pp. 256-266). Cham: Springer Nature Switzerland. [CrossRef]

[32] Akyon, F. C., & Kalfaoglu, M. E. (2019, October). Instagram fake and automated account detection. In *2019 Innovations in intelligent systems and applications conference (ASYU)* (pp. 1-7). IEEE. [CrossRef]

[33] Bakhshandeh, B. (2019). Instagram fake spammer genuine accounts. Kaggle. Retrieved from https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts (accessed on 01 January 2026).

[34] Al-Amin, M., Deb, P., Rintu, I. J., Islam, M. M., Das, D. C., & Khan, S. A. (2024). Genuine or Spammer? Enhanced Fake Profile Detection using Feature Synthesis. In *2024 27th International Conference on Computer and Information Technology (ICCIT)* (pp. 3499–3504). IEEE. [CrossRef]

[35] Gunawan, H., Budhi, G. S., & Gunadi, K. (2025). *Machine Learning-Based Fake Account Detection System: Instagram Case Study* (Doctoral dissertation, Petra Christian University).

[36] Goyal, B., Gill, N. S., & Gulia, P. (2024). Securing social spaces: machine learning techniques for fake profile detection on instagram. *Social Network Analysis and Mining*, *14*(1), 231. [CrossRef]

[37] Arunprakaash, R. R., & Nathiya, R. (2024, August). Leveraging Machine Learning algorithms for Fake Profile Detection on Instagram. In *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)* (Vol. 1, pp. 869-876). IEEE. [CrossRef]

Abdullah Khan received his MS degree in Computer Science in 2025 from the University of Wah, Wah Cantt, Pakistan, and his BS degree in Software Engineering in 2022 from COMSATS University Islamabad, Wah Campus, Pakistan. He has published research on CAN bus intrusion detection and has multiple manuscripts under review in areas such as quantum neural networks and blockchain-based security. He has presented his work at international conferences, including ICETECC 2025. His research interests include AI security, deep learning, explainable AI (XAI), quantum computing, intrusion detection in autonomous vehicles, and medical imaging. (Email: abdullahkhanswati@outlook.com)

Arooj Fatima received her M.S. degree in Computer Science from the University of Wah, Pakistan, in 2025, her M.Sc. degree in Computer Science from the University of the Punjab, Pakistan, in 2021, and her B.Sc. degree in Computer Science from the University of Sargodha, Pakistan, in 2019. Her research focuses on the application of deep learning and machine learning techniques for secure and intelligent systems. Her broader interests include deep learning, explainable artificial intelligence, cybersecurity, and advanced intrusion detection and prevention systems. (Email: fatimaroojhxe@gmail.com)

Ridda Jamil received her M.S. degree in Computer Science from the University of Wah, Wah Cantt, Pakistan, in 2025, and M.Sc. degree in Information Technology from the University of the Punjab, Lahore, Pakistan, in 2021. Her research interests include cybersecurity, deep learning, medical image processing, and artificial intelligence, with a focus on developing intelligent solutions for real-world applications. (Email: riddajameel7@gmail.com)

Hassan Ahmed received his MS degree in Computer Science in 2024 from the University of Wah, Wah Campus, Pakistan, and his BS degree in Software Engineering in 2022 from COMSATS University Islamabad, Wah Campus, Pakistan. He is currently working in the Department of Computer Science at FAST National University of Computer and Emerging Sciences, Pakistan. He has published research in the areas of network security and deep learning, and has served as a reviewer for peer-reviewed international conferences. His research interests include AI, deep learning, XAI, network security, large language models, and computer vision. (Email: hassan.ahemd@nu.edu.pk)

Aini Saba received her M.S. degree in Computer Science from the University of Wah, Wah Cantt, Pakistan, in 2025. Her research interests include deep learning, medical image processing, artificial intelligence, and cybersecurity, with a strong emphasis on developing intelligent and secure systems to address practical challenges in healthcare and technology. (Email: aini786saba@gmail.com)