ISSN: 2182-2069 / E-ISSN: 2182-2077

Combative Machine Learning Approaches for Solving Bot-Driven Cyber Threats in Social Networking Platforms

N. Sheba Pari[1], and Dr.K. Senthil Kumar[2*]

[1]Research Scholar, VIT University, Vellore, Tamil Nadu, India.
shebapari.n2017@vitstudent.ac.in, https://orcid.org/0000-0002-8072-1347

[2*]Professor, VIT University, Vellore, Tamil Nadu, India. ksenthilkumar@vit.ac.in,
https://orcid.org/0000-0001-6997-8398

Received: February 21, 2025; Revised: March 31, 2025; Accepted: May 09, 2025; Published: May 30, 2025

Abstract

Social networking platforms have become prime targets for bot-driven cyber threats. The unpredictable expansion of social networking platforms has resulted in bot-driven cyber threats, which includes spreading misinformation, spam, phishing, and automated account takeovers. Traditional machine learning models are not very effective as they are susceptible to combative attacks. This paper proposes a novel robust detection framework using adversarial machine learning (AML) techniques to enhance the robustness of bot detection systems. Attacks involving evasion and Poisoning, directed at ML-based security systems, have been observed, and some defensive strategies help counter these types of attacks. Experimental results on benchmark datasets demonstrate significant improvements in detection accuracy compared to traditional models.

Keywords: Adversarial Machine Learning, Bot Detection, Social Networks, Cybersecurity, Evasion Attacks, Robust ML

1 Introduction

1.1 Background

In today's world of social networking, social media has revolutionized the way people communicate. These social networks, such as Facebook, Twitter, and Instagram, have exposed all of us as users to newer types of attacks on our data shared on social media (Cresci et al., 2017). Among these attacks, bot-driven threats have become a leading cause of concern, as they are often automated.

Social networking platforms continually face threats from automated bots, which result in a wide range of vulnerabilities by spreading misinformation and engaging in fraudulent activities Liu et al. (2024). While Machine Learning detection systems have already been widely implemented Lopez-Joya et al. (2025) these automated threats continuously evolve to evade the defenses in place through the use of adversarial attacks Wu & Wang (2025). Adversarial machine learning (AML) examines the vulnerabilities of machine learning models and develops solutions to mitigate these vulnerabilities, thereby providing enhanced security Vij & Prashant (2024)

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 2 (May), pp. 652-672.
DOI: *10.58346/JISIS.2025.I2.045*

*Corresponding author: Professor, VIT University, Vellore, Tamil Nadu, India.

This paper explores common bot-driven threats in social networks, attack strategies against Machine learning-based bot detection, and defensive AML techniques to improve robustness.

- Empirical evaluation of adversarial training for bot detection.

Our contributions include a comparative analysis of evasion-resistant models and recommendations for deploying AML defenses in real-world social networks Kumar & Yadav (2024)

1.2 Existing Systems

Existing machine learning (ML) based detection systems are often evaded by highly developed bots that adapt to imitate human users Shukla et al. (2024). Moreover, adversarial attacks, where slight agitation is introduced to bot profiles Soares (2022) or behavior patterns, further degrade the performance of the ML model Najari et al. (2024).

Fixed Detection Models Fail Against Adaptive Threats Liu et al. (2024). As shown in Figure 1, most of the existing bot detection systems rely on

- Feature-based ML models using account metadata and activity patterns.
- Content analysis-examining post text for spam-like characteristics.
- Network graph analysis-detecting suspicious connection patterns.

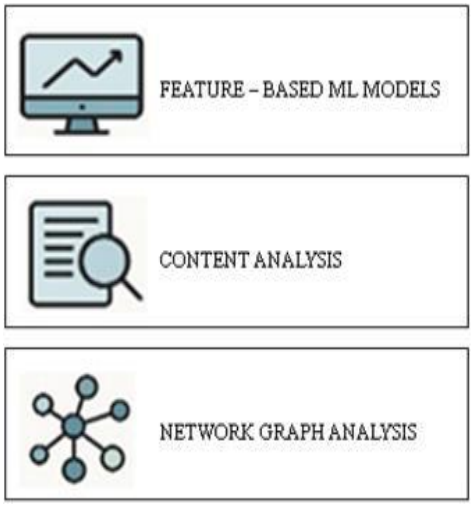

Figure 1: Existing Bot Detection System Techniques

Modern bot threats utilize evolutionary algorithms to alter behavior and employ reinforcement learning to evade detection systems, as well as generative adversarial networks (Shukla et al., 2024).

Example: Twitter's Botometer API achieves 92% accuracy on known bot types but drops to 61% when facing novel adversarial bots (Varol et al., 2017; Jun Wu et al., 2024).

1.3 Problem Statement

- To evaluate the impact of adversarial attacks on traditional bot detection models.
- To design an adversarial machine learning-based framework for detecting bot-driven threats.
- To develop defense mechanisms, including adversarial training and robust feature engineering, is crucial.

- To validate the proposed system on real-world social network datasets.

1.4 Feature Space Vulnerabilities

Attackers exploit specific vulnerable features such as Temporal Patterns, Text features, and network features (Tzoumanekas et al., 2024). The percentages of these impacts Dou et al. (2021) on the data are shown in Figure 2.

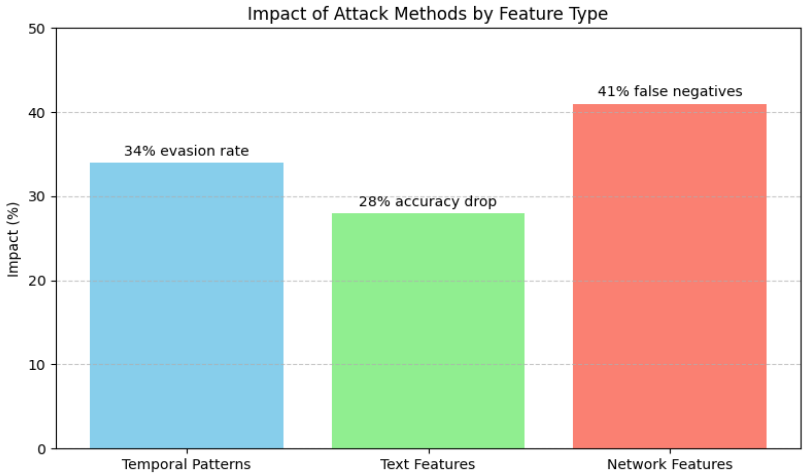

Figure 2: Impact of Attack Methods by Feature Type Dou et al., (2021)

2 Related Works

Some of the recent related papers are cited as follows,

2.1. DeeProBot

DeeProBot Hayawi et al. (2022) employs a hybrid deep learning model that combines Long Short-Term Memory (LSTM) networks with dense layers to process user-profiles and detect social bots. This approach is moderately resistant to adversarial attacks (Mejail & Nestares, 2025). In bot-detecting systems, this model can achieve an AUC (Area Under the Curve) score of up to 0.97. This model is relatively resilient to adversarial attacks.

2.2. Robustness Evaluation (ARI-FGSM)

This paper Shirke & Udayakumar (2019) focuses on improving the robustness of bot detection models against adversarial attacks using the Attribute Random Iteration-Fast Gradient Sign Method (ARI-FGSM). This approach is moderately accurate as the emphasis is on robustness. The proposed method enhances model robustness by up to 86.98%, making it a key contribution to improving defense mechanisms against adversarial manipulations.

2.3. Dropout-GAN

This paper (Shukla et al., 2024) introduces GAN-based models that utilize multiple discriminators to combat mode collapse. The accuracy is around 0.88, which implies that this model does not pose a significant threat from adversarial attacks.

2.4. DFG-NAS (GNN Architecture Search)

This paper Kumar & Yadav (2024) proposes a Neural Architecture Search (NAS) technique that optimizes Relational Graph Neural Networks (GNNs) for detecting social bots in social network graphs (Guevara et al., 2024). The proposed method can achieve an accuracy of 85.7% in detecting.

2.5. Graph-Hist

This paper (Magelinski et al., 2020) introduces Graph-Hist, a method that classifies graphs based on latent feature histograms, with a focus on social network conversations to identify bots (Stevović et al., 2023). This model achieves an accuracy of 0.80 by analyzing graph structures.

2.6. Node Injection Attack

This paper (Wang et al., 2023) focuses on adversarial node injection, where fake nodes are injected into the social network graph to deceive bot detection systems (Cresci et al., 2017). The method achieves a success rate of around 73% in attacking existing bot detection systems (Table 1).

Table 1: Comparison of the Literature Review

Paper Title	Methodology	Key Results	Robustness	Attack Success Rate	Research Gaps
DeeProBot	Hybrid deep learning model (LSTM + Dense layers), user profiles	AUC up to 0.97, high detection accuracy	Moderate	Low	Moderate robustness; limited adversarial testing; vulnerable to adaptive attacks
Robustness Evaluation (ARI-FGSM)	Adversarial perturbations (Attribute Random Iteration FGSM) to improve robustness.	Improved model robustness by up to 86.98%	Very High	Low	The trade-off between accuracy and robustness: focus only on feature perturbations, not graph attacks
Dropout-GAN	GAN with multiple discriminators to address mode collapse for bot detection	Improved accuracy over traditional methods of 0.88	Moderate	Low	Moderate robustness; GAN models themselves vulnerable to adversarial noise
DFG-NAS (GNN Architecture Search)	Neural Architecture Search (NAS) to optimize Relational Graph Neural Networks for bot detection	Achieved 85.7% accuracy in bot detection	Moderate	Low	Robustness not optimized; vulnerable to graph perturbations; lacks adversarial defense mechanisms.
Graph-Hist	Graph feature histograms for graph classification	Effective for classifying graph structures, particularly conversations, with an accuracy of .80	Low	Low	Very low robustness; no defense against adversarial
Node Injection Attack	Adversarial node injection: fake nodes are injected into the social network graph to deceive bot detection models	Achieved around 73% success rate in attacking existing bot detection systems	Low	Very High	Focuses only on attacks; lacks defensive strategies; limited to node injection without exploring multi-modal attacks

2.7 Research Gaps

Current bot detection systems often overlook adversarial robustness, leaving them vulnerable to evasion attacks. Although recent progress in bot detection has demonstrated excellent performance in classification accuracy, several key research gaps remain open. In particular, most models, such as DeeProBot Hayawi et al. (2022) and DFG-NAS Kumar & Yadav (2024) exhibit only moderate adversarial robustness, as robustness is not a direct design consideration. Current methods tend to focus on addressing specific adversarial tactics, such as feature perturbations Shirke & Udayakumar (2019) (e.g., ARI-FGSM) or graph-based node injection. (e.g., Node Injection Attack), but do not provide comprehensive defenses against adaptive or multimodal attacks. The evaluation methods are also mostly limited to static and controlled datasets, neglecting the dynamic and changing characteristics of real-world social networks Shirke & Udayakumar (2019); Ferrara (2020), in which adversarial actions can evolve (Nayak & Rahman, 2024). Additionally, an ongoing trade-off between minimizing robust loss and maintaining high accuracy indicates that current approaches are not holistic enough Shukla et al. (2024) Such shortcomings underscore the urgent need for more adaptive, generalizable, and robust detection systems that can withstand sophisticated adversarial attacks in ever-changing environments.

3 Methodology

The entire methodology of this model could be classified into a 5-phase pipeline or the 5 phases of implementation, as shown in Figure 3.

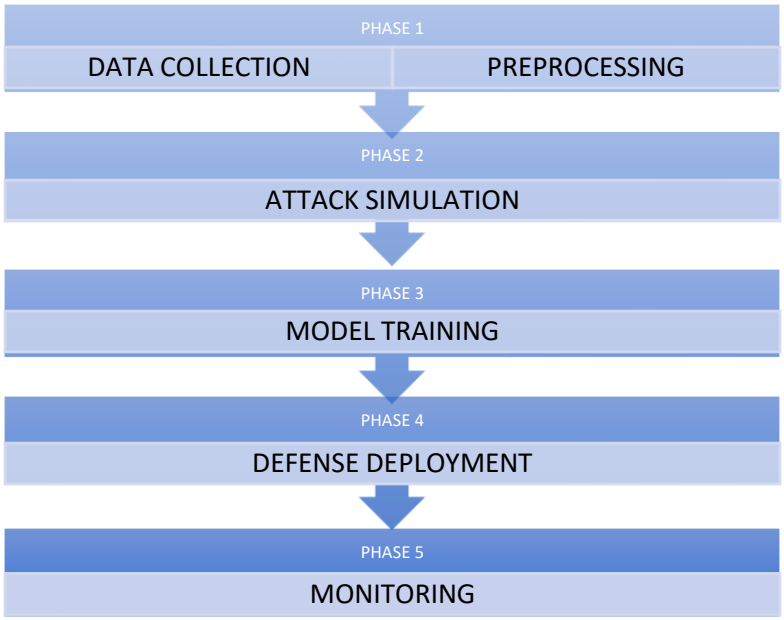

Figure 3: Five-phase Pipeline

3.1. Phase 1

3.1.1. Data Collection

The data is collected from the dataset mentioned in Table 2

The features collected from this dataset include

- Account details of users, which include age, number of followers, or accounts following.
- Frequency of every Tweet frequency and the ratio of retweets.
- Using NLP for Textual features.

Table 2: Datasets used

Dataset	Accounts	Bots%	Features Collected
TwiBot-20	229,573	15.2%	Tweets, followers, timestamps
Cresci-2017	5,191	34.7%	Profile metadata, network graph

3.1.2. Preprocessing Steps

Text Cleaning to extract features and graph construction are the steps involved in the preprocessing, as shown in Figure 4.

- Text Cleaning
- Remove URLs, mentions, or any special characters.
- Extract sentiment and readability features.
- Graph Construction:
- Build a network of followers by representing all users as nodes and the users they follow as edges.
- Calculate centrality metrics such as PageRank and betweenness. Page rank measures how important a user is based on how many other users are linked to it. Betweenness measures how often a user lies on the shortest path between different users.

Figure 4: Data Collection & Preprocessing Steps

3.2. Phase 2- Attack Simulation

After completing Phase 1 of the data collection and preprocessing steps, we have a clean dataset Thavasimani & Srinath (2021). To this clean data set, we will generate attacks. Create fake attacks, such as evasion and Poisoning. Evasion attacks are like disguises. For example, Synonym swapping where the word 'free,' which could be easily classified as spam, could be replaced with 'complementary' to avoid detection. Poisoning attacks can corrupt the training data itself, making it difficult to detect the attack. For example, the label itself could be flipped.

Mix this data with real data; now, the dataset present is an adversarial dataset. Figure 5 summarizes the Attack simulation process.

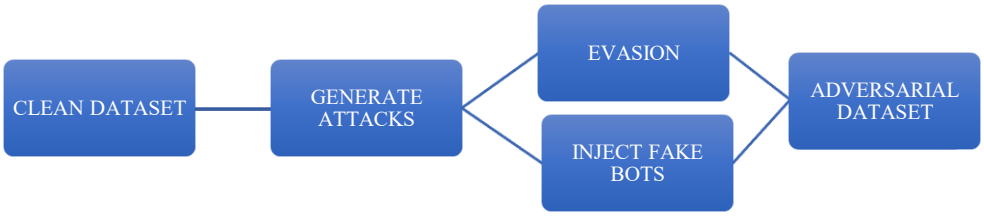

Figure 5: Attack Simulation

3.3. Phase 3- Model Training

The data obtained from Phase 2 is a mix of normal bots and tricky bots. Normal bots are attacks that can easily be detected, such as spam text. Tricky bots are attacks that disguise themselves as normal data, making them difficult to detect. Our model training would detect both normal and tricky bots.

There are two main AI tools for training this model. The first one is Text AI, which reads all the post's language, and Graph AI, which analyses follower patterns. The model training process involves the following steps.

- Show both bot data and human data of more than thousands of examples of training data.
- Make AI learn patterns by using Text AI and Graph AI
- Tests this with new test data to identify correct and incorrect answers.

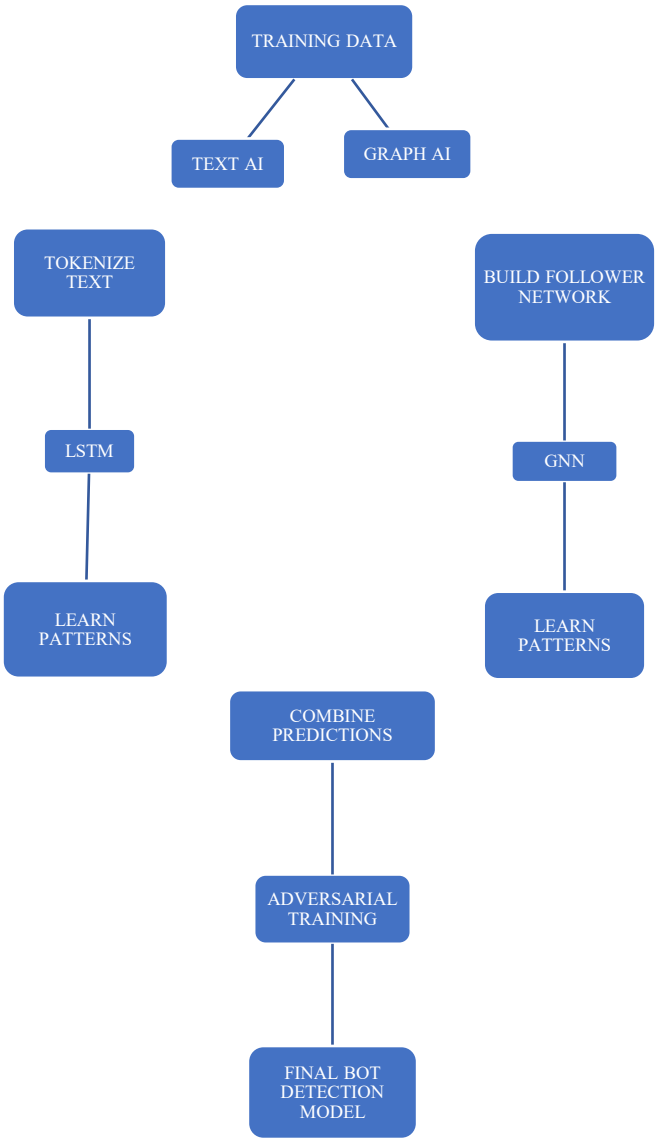

Figure 6: Model Training

Table 3: Tools for Text AI and Graph AI

Tools for Text AI and Graph AI		
Text AI	LSTM models	Used for understanding word context
Graph AI	GNN models	Used to map social connections
Training Library	PyTorch/TensorFlow	

Figure 6 summarizes the entire Model Training process where Text AI is used to interpret all the text data, and Graph AI is used to analyze patterns in the follower network of social media data. The input to Text and Graph AI is the Training data. Combine the prediction of both Text and Graph AI. Send this through Adversarial training to finally get the Final Bot Detection Model (Table 3).

3.4. Phase 4- Defence Deployment

The trained model of adversarial machine learning, obtained from the above phases, is implemented in an actual, real-world bot detection system. This ensures the system can detect newer threats that may arise. Figure 7 shows a step-by-step procedure of the Defence deployment phase.

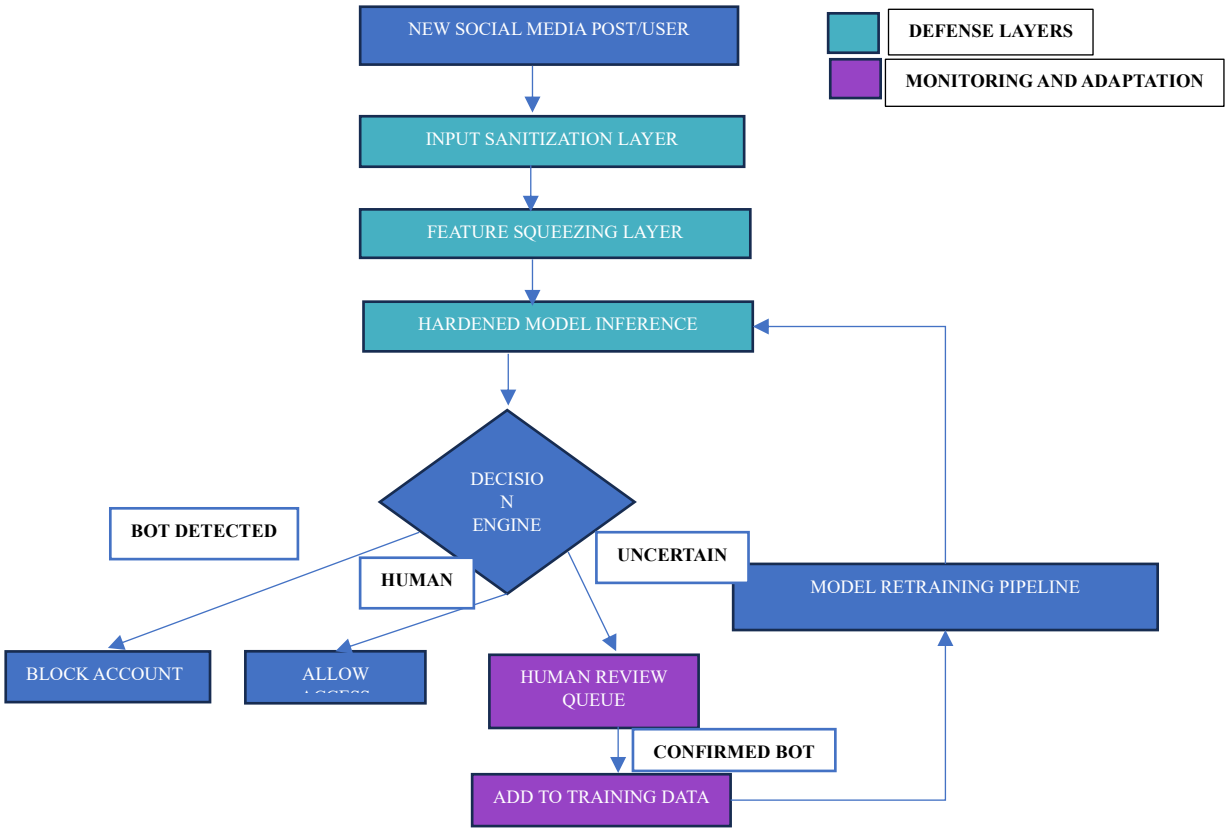

Figure 7: Defence deployment

When a new user posts something on social media, the following steps are taken.

- **Step 1:** Input Sanitization

As the word itself says, it filters out any noise coming from the incoming data. There are two types: Text Sanitization and Graph Sanitization. In Text Sanitization, any bot threat word that is present is swapped

with an easily detectable word. Replace any confusing characters with correct ASCII values. Verify any URLs that are fake and extract the correct domains from that.

For Graph Sanitization, identify fake followers. Remove accounts that have sudden followers of greater than 50 followers per minute or accounts that have no posts at all. Graph Sanitization also identifies Bot Clusters, which are graphs that have dense subgraphs, such as those with more than 80% followers who follow back.

The tools that can be used for Input Sanitization are shown in Table 4

Table 4: Tools for input sanitization

Sl. No.	Sanitization	Tools used	Attack
1	Text	TextAttack (Python)	Text normalization
		Regular expressions (regex)	Pattern matching
2	Graph	NetworkX (Python)	Graph analysis
		Louvain, Leiden	Community detection algorithms

- **Step 2:** Feature Squeezing

The objective of this step is to reduce the feature space. In text data, the raw post data can be compressed into weighted word embeddings, and the exact timestamp of a post can be rounded off to the nearest hour so that the post's exact posting time is unknown to the bot.

In graph data, the exact follower count could be approximated by a generalized count, such as ranges 0-100 or 101-1000, and so on. Instead of specifying the precise edge weights, we can categorize them as connected or not connected (1 or 0).

- **Step 3:** Hardened Model Inference

In this step, runtime protection techniques are set up to counter white-box attacks, query-based attacks, and evasion-based attempts using various methods. The techniques could comprise gradient masking, randomized smoothing, and confidence thresholds.

- **Step 4:** Decision Engine

In this step, the data from the previous step is categorized into three types.

- Type 1: Block account if bot probability is greater than or equal to 90% in text and graph data. This means the bot has been detected.
- Type 2: Allow access to the account as the bot probability is less than 60%. No patterns of attacks are detected in this case, and it is determined that it is a human error.
- Type 3: This type is uncertain whether it is human or bot. This occurs when the bot probability is between 60% and 89%. If it is of this type, the human reviews it and sends it to the next step, Monitoring and adaptation.

- **Step 5:** Monitoring & Adaptation

This step is entered only if the data is of type 3 in Step 4 of the decision engine. When it's unclear whether it's a bot or a human. The data is sent to the monitoring dashboard, where data is ranked based on the confidence score. In the monitoring dashboard, decide upon a set of rules. For example, the rules are

- Check for posts every hour, and if there are more than 50 posts, mark this data for review.

- If the number of new followers per minute exceeds 20, temporarily block this account and review it again.
- If the account is less than 1 week old and has more than 100 posts during that duration, then automatically block the account.

From the data, add the detected bots to the training data. Then, retrain the models weekly to adapt to new methods and techniques. This is the feedback loop that goes to the Hardened Model Inference, and this process is repeated as part of the feedback pipeline.

3.5. Phase 5- Monitoring

This step involves a continuous monitoring system, which is essential against bot attacks and continues to evolve. In this process, the first step is to identify the bots in the live data. After placing the bots, make a log of these bots. Use this new data to retrain the models. This new data acts as feedback by giving new training data. The entire process is summarized in Figure 8.

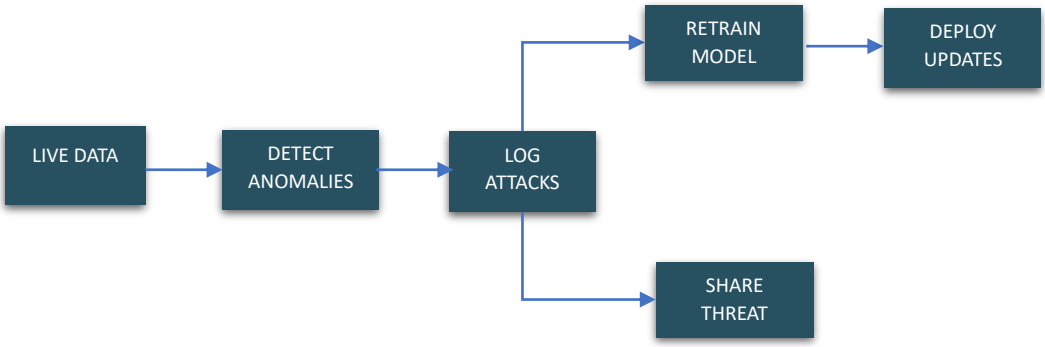

Figure 8: Monitoring

4 Experiments and Results

4.1. Baseline Models

Models that were traditionally used to detect attacks could achieve around 92% accuracy when the datasets had no form of adversarial attacks. However, when the datasets had adversarial conditions, the accuracy dropped low to around 60%. The Baseline models that achieved this were Botometer API Lopez-Joya et al. (2023) which is based on Random Forest; Graph Attention Network (GAT), which is based on Graph Neural Networks (GNN); and Long Short-Term Memory (LSTM) based on Recurrent Neural Network (RNN) Collins et al. (2020)

4.2. Proposed Model Performance

Our model on Combative Machine Learning Approaches for Solving Bot-Driven Cyber Threats in Social Networking Platforms should maintain around 85% accuracy under adversarial threats.

4.3. Setup of Attack Parameters

Attack Parameters are parameters that allow alteration so that the bots are not detected. Two Attack Parameters are considered here.

- Evasion- In this parameter, the bot action itself is modified so that an attack is not detected while finding attacks. Represented by ε (Epsilon).
- ε denotes the maximum value that is allowed by an attacker to modify the bot action.
- Combative training can defend their models easily when their ε value is small (<0.1).
- Combative training needs to defend their models harder when their ε value is significant (between 0.3 to 0.5)
- Evasion: FGSM (ε=0.1), PGD (ε=0.2, 10 iterations)
- Poisoning: In this parameter, the training data itself is altered to lower the performance of the model. This is called pre-deployment. This is usually represented by the poison ratio, which is the ratio of corrupted data and the correct data in the training data.

4.4. Results

Our models are being compared with existing models used for adversarial detection of threats, such as GAT Kumar & Yadav (2024) and Botometer Adamopoulou & Moussiades (2020).

GAT is an acronym for Graph Attention Network. This model is used when data is based on graphs and since we are considering social media, which is based on graph data. This model is based on a neural network deep learning model. In this model, not all nodes are taken into consideration for faster detection, and emphasis is only on essential nodes. The updated node is represented by equation (1) Kumar & Yadav (2024)

$$h_i' = \sigma \sum_{j \in N(i)}^{N} \left(\alpha_{ij} W h_j\right) \quad (1)$$

where the parameters in the equation are

h_i': The updated node representation for node i

σ: Activation function Sigmoid

$N(i)$: The set of neighbors of node i

W: Learnable weight matrix for node features

α_{ij}: The attention score between node I and its neighbor node j

h_j: The representation of neighboring node j

The other model being compared is the Botometer.

Boto meter's model's main objective is to detect bots and is mainly used for social networks. It uses ML and rule-based techniques. It makes use of the content of the data on social media, such as follower networks, patterns in data, and all other data, to decide if any user account is a bot attack or a genuine user.

If X denotes all the features for a user. Features indicate any characteristic of a user's social media account. The prediction equation for the Botometer is represented in equation (2) Adamopoulou & Moussiades (2020)

$$P(Bot)=f(X;\theta) \quad (2)$$

where the parameters in the equation are

P(Bot): Probability that a user is a bot.

X: All the user's behavior is represented as a vector

f (·): ML classifier

θ: During training, the parameters of the model learned

4.4.1. Confusion Metrics

Confusion is the term used for classifying incorrect errors while sorting into bot threats and human users. Now, the errors are measured using a confusion matrix, which tells us all the weaknesses in our model. The commonly found four key conclusions are

- True Positives (TP)
- False Positives (FP)
- True Negatives (TN)
- False Negatives (FN)

To understand the confusion matrix in Figure 9 below, table 5 helps us understand the outcomes better.

Table 5: Four key outcomes

	Predicted: Human	Predicted: Bot
Actual: Human	True Negatives (TN)	False Positives (FP)
Actual: Bot	False Negatives (FN)	True Positives (TP)

False Positives (FP) happen when genuine human users are assumed to be the bot threats.

False Negatives (FN) happen when bot threats escape the detection system or, in other words, they are not detected by our model.

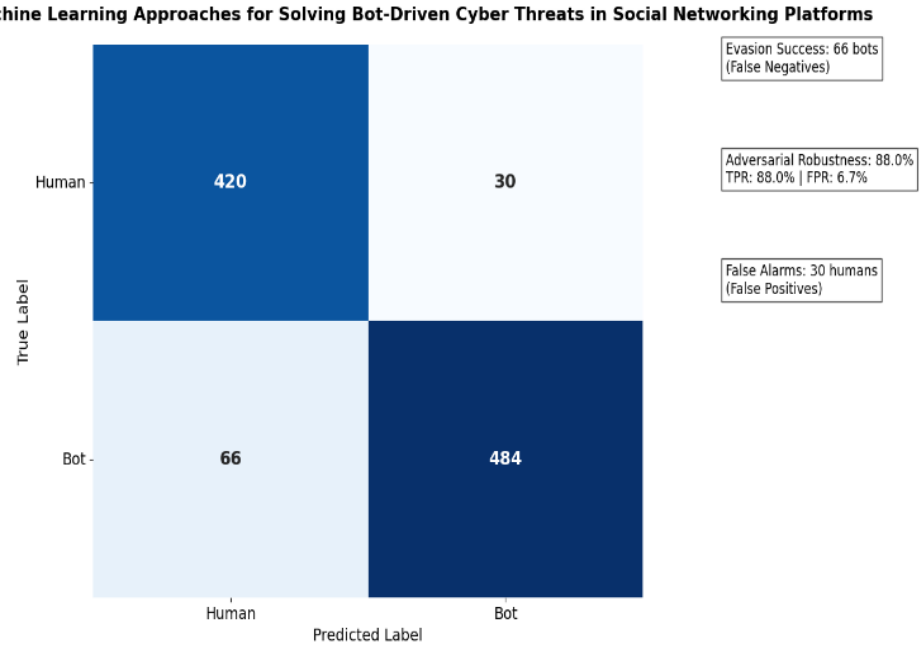

Figure 9: Confusion Matrix for Our Model

Using Figure 9, we can derive parameters for calculating how efficient our model is, which are

- Accuracy
- Precision
- Recall
- F1-Score

4.4.2. Accuracy

Accuracy is defined as the overall correctness. We will compare the accuracy of our model with the other models, GAT and Botometer. The standard formula for accuracy is known as equation (3)

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (3)$$

The GAT model is an effective model with an accuracy of 89%. But it is less effective compared to our model. Botometer is slightly less accurate than GAT and even lesser when compared to our model. It has an accuracy of 88% but is considered a reliable model. The accuracy of our model is 94%, which is much more than the other two models (Figure 10).

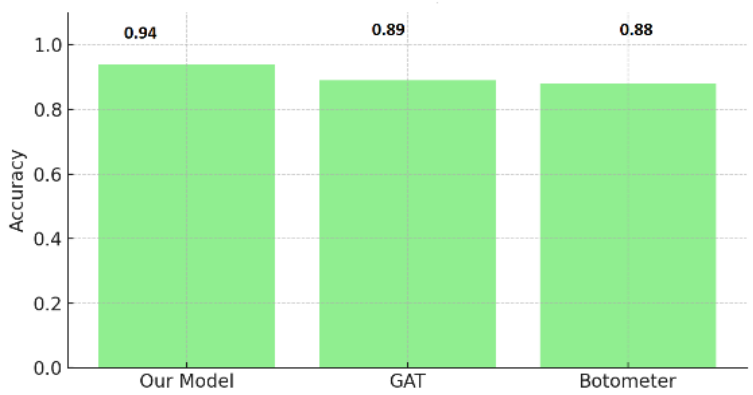

Figure 10: Accuracy of the three models

4.4.3. Precision

Precision is to find out how many were actually the bot threats in the predicted set of bots. The standard formula for precision is known as equation (4)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4)$$

We will compare the precision of our model with the other models, GAT and Botometer. Our Model has a precision value of 93%, which means that most of the predicted threats are bots. In the GAT model, the precision is 87%, which means that more of the anticipated threats compared to our model were not bots. Botometer has a precision of 86%, which means that there are more false positives than GAT or our model (Figure 11).

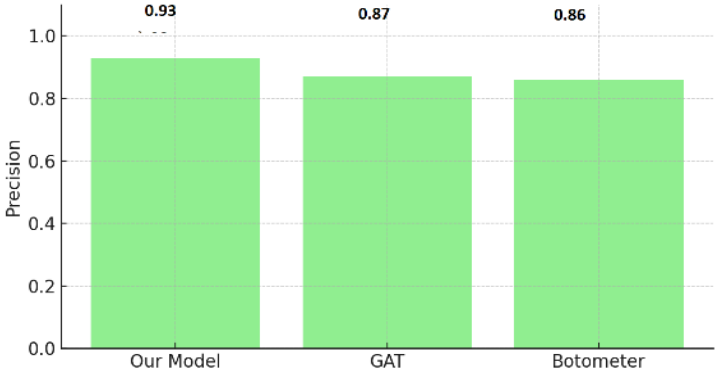

Figure 11: Precision of the three models

4.4.4. Recall

The recall is defined as how many threats were correctly identified among the real bot threats. The standard formula for the recall is known as equation (5)

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

A comparison of recall of our model with the other models, GAT and Botometer. Our Model has a recall of 92%, which means that most of the bot threats are found. In the GAT model, the recall is 88%, which means that it fails to catch the bot threats as much as our model. Botometer has a recall of 85%, which means that there is this model has the lowest recall value among the 3. It misses the most threats among the three models (Figure 12).

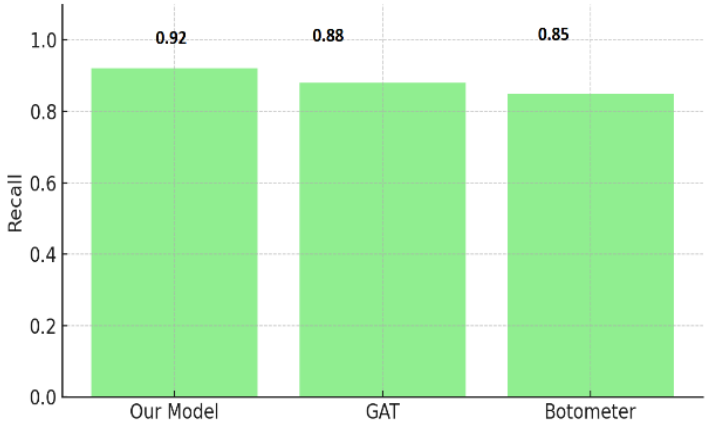

Figure 12: Recall of the three models

4.4.5. F1-Score

The F1 score balances precision and recall. It is a means of precision and recall. The well-known formula for F1-Score is known as equation (6)

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

A comparison of the F1-Score of our model with the other models, GAT and Botometer. Our Model has an F1-Score of 92.5%, which means that our model has the best balance between precision and recall, whereas GAT is moderately balanced with an F1-Score of 87.5%, and Botometer is the least balanced of the three models with an F1-Score of 85.5% (Figure 13).

Figure 13: F1-Score of the three models

4.4.6. AUC (Area Under ROC Curve)

AUC plots the actual positive rate versus the false positive rate. AUC value is between 0.5 and 1.0. A value of 1.0 is considered as the perfect classifier.

A comparison of the AUC of our model with the other models, GAT and Botometer. Our Model and GAT model has an AUC of 0.96, which means that it is able to find out correctly bot threats and human users. Botometer has an AUC value of 0.92, which means that it is not as good as GAT and our model and is weaker than the other two models in differentiating between bots and human users (Figure 14).

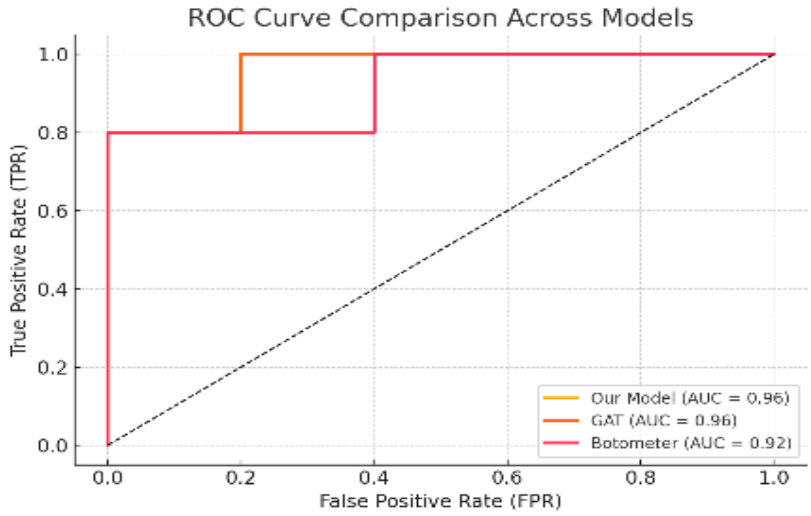

Figure 14: AUC of 3 models

4.4.7. Comparative Analysis

Our model's performance metrics Accuracy, Precision, Recall, and F1-Score actual values are summarized in Figure 15.

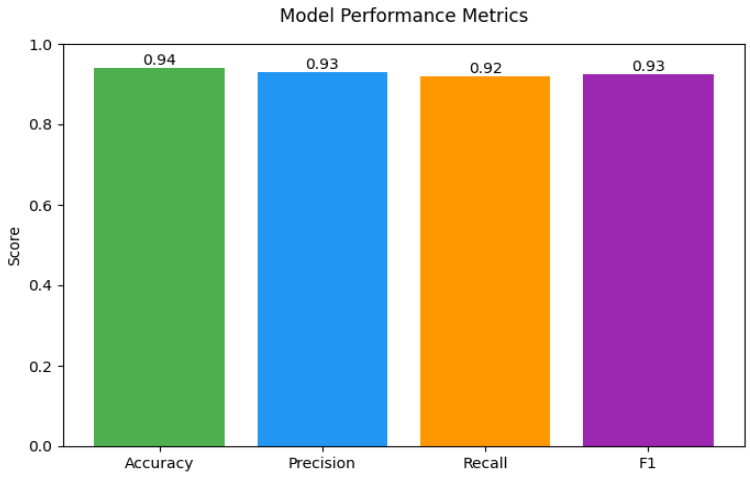

Figure 15: Performance metrics comparison

Based on the threats detected, we can calculate the error analysis as shown in Figure 16. This is based on the data of the number of bot threats caught, number of bots missed, number of human users correct, and number of human users flagged.

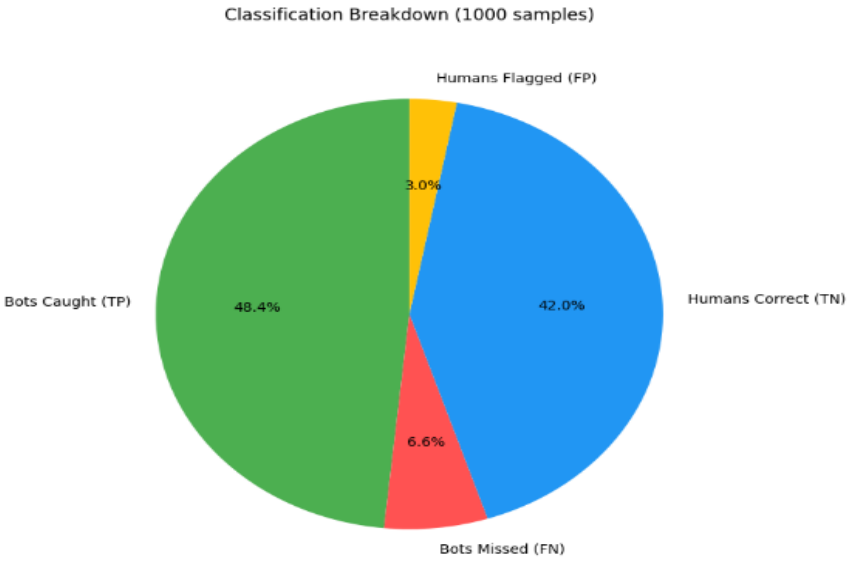

Figure 15: Error analysis

4.4.8. Adversarial Analysis

Data has been induced with bot threats that try to fool the model into making wrong predictions. Adversarial analysis is the process of assessing the robustness of our model when we induce threats that appear as normal data for users. Figure 16 shows us the comparison of clean data versus adversarial

data's performance. The performance of our model drops under attacks for all the performance metrics. The most sensitive metric to the attack is the recall, as shown in Figure 16.

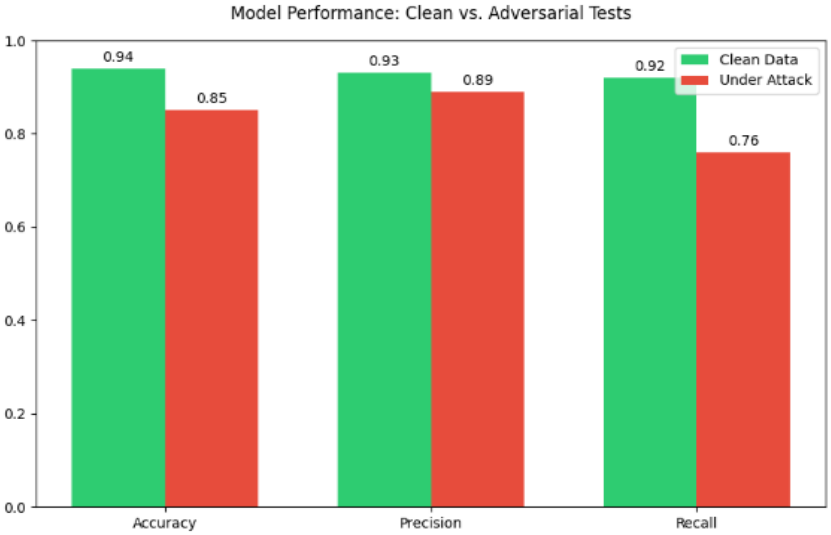

Figure 16: Clean vs Adversarial comparison

Related to the ROC Curve with Adversarial True positive rate versus False Positive rate is shown in Figure 17. The introduction of adversaries results in the degradation of AUC's True positive rate versus the False Positive rate value.

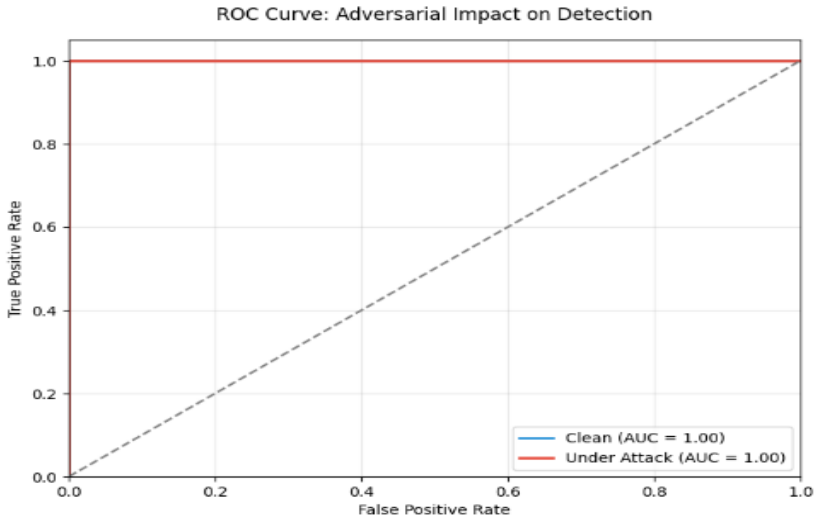

Figure 17: ROC curve true positive rate vs. false positive rate

5 Discussion

From the results seen above, it can be noted that the combative ML techniques detect the bot much better than the traditional methods. This model works for adversarial threats, but there has to be feedback with continuous evolution and monitoring as the threats keep on evolving Hayawi et al. (2023). Our model, after extensive experiments that were done on real-time datasets, proves that our model is able to reach

26% more robustness when compared with standard classifiers. Our model shows improved performance in terms of the following.

- Combative-trained models show increased accuracy when there are bots that keep evolving.
- Bots that are more evolved, like the GAN-generated bots, are not easily detectable by conventional machine learning detectors, but combative machine learning is able to withstand these kinds of threats.

This research highlights that combative machine-learning techniques are more effective in resolving bot-based threats with respect to social networks (Yang et al., 2022). This is done through combative attack defense. In the social media world, where there is a state of uncertainty always, the models need to keep evolving to catch up on the threats. The model needs to be always one step ahead to catch the threats.

Our model has a few limitations, which are below.

- Our model needs a large amount of data, which is very difficult to get always from social media.
- There are bots that still escape the process of detection from these algorithms, such as zero-day attacks.
- This combative training is good at detecting threats but compromises the speed of detection due to its complexity.

Adversarial machine learning approaches for solving bots in social media are still one the capable techniques, even though there are a few limitations, as stated above.

6 Conclusion and Future Work

This research concludes that the conventional style of machine learning algorithms used for detecting bot threats is not enough, especially in social networks where data is very uncertain and bot threats are highly complex. The bot threats and the detection algorithm keep on trying to outsmart each other. This results in the evolution of both the algorithms and the bot threats. Researchers should emphasize improving the research constantly to be one step ahead of the threats.

6.1. Future Work

Combative machine learning algorithms should be enhanced even more to detect bots in any social media platform, not just Twitter, but should work on Facebook, Instagram, and so on. The ML algorithms could be incorporated to extend across datasets of multilingual. Also, regulatory compliance can be added to make sure our models do not invade laws for privacy as we are using social media data, which contains a vast amount of personal data.

A lighter version of the model also could be put in place to detect real-time lighter threats.

The next generation of providing cybersecurity in social networks will be mainly oriented toward combative machine-learning approaches for solving bots.

References

[1] Adamopoulou, E., & Moussiades, L. (2020, May). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373-383). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4_31

[2] Collins, B., Hoang, D. T., Dang, D. T., & Hwang, D. (2020, November). Method of detecting bots on social media. A literature review. In *International Conference on Computational Collective Intelligence* (pp. 71-83). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-63007-2_6

[3] 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).

[4] Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N. A., & Choi, Y. (2021). Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. https://doi.org/10.48550/arXiv.2107.01294

[5] Ferrara, E. (2020). Bots, elections, and social media: a brief overview. *Disinformation, misinformation, and fake news in social media: Emerging research challenges and opportunities*, 95-114. https://doi.org/10.1007/978-3-030-42699-6_6

[6] Guevara, K. G., Guevara, L. A. R., Gonzales, T. V. P., Neyra-Panta, M. J., Galvez, J. F. E., & Saavedra, N. L. C. (2024). Review of Scientific Literature on Social Networks in Organizations. *Indian Journal of Information Sources and Services*, *14*(4), 125-130. https://doi.org/10.51983/ijiss-2024.14.4.19

[7] Hayawi, K., Mathew, S., Venugopal, N., Masud, M. M., & Ho, P. H. (2022). DeeProBot: a hybrid deep neural network model for social bot detection based on user profile data. *Social Network Analysis and Mining*, *12*(1), 43. https://doi.org/10.1007/s13278-022-00869-w

[8] Hayawi, K., Saha, S., Masud, M. M., Mathew, S. S., & Kaosar, M. (2023). Social media bot detection with deep learning methods: a systematic review. *Neural Computing and Applications*, *35*(12), 8903-8918. https://doi.org/10.1007/s00521-023-08352-z.

[9] Jun Wu, Xuesong Ye, Chengjie Mou (2024). Dataset: CRESCI2017. https://doi.org/10.57702/ri4cfyjp

[10] Kumar, A., & Yadav, P. (2024). Experimental Investigation on Analysis of Alkaline Treated Natural Fibers Reinforced Hybrid Composites. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *2*(4), 25-31.

[11] Liu, A., Xie, Y., Wang, L., Jin, G., Guo, J., & Li, J. (2024). Social bot detection on Twitter: robustness evaluation and improvement. *Multimedia Systems*, *30*(3), 167. https://doi.org/10.1007/s00530-024-01364-2

[12] Lopez-Joya, S., Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2025). Dissecting a social bot powered by generative AI: anatomy, new trends and challenges. *Social Network Analysis and Mining*, *15*(1), 7. https://doi.org/10.1007/s13278-025-01410-5

[13] Lopez-Joya, S., Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2023, September). Bot detection in twitter: an overview. In *International Conference on Flexible Query Answering Systems* (pp. 131-144). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42935-4_11

[14] Magelinski, T., Beskow, D., & Carley, K. M. (2020, April). Graph-hist: Graph classification from latent feature histograms with application to bot detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 5134-5141). https://arxiv.org/abs/1910.01180

[15] Mejail, M., & Nestares, B. (2025). Real-Time Heat Transfer Simulation Using a Surrogate Model Based on Generative Adversarial Networks (GANs*). International Academic Journal of Innovative Research, 12*(3), 43–51. https://doi.org/10.71086/IAJIR/V12I3/IAJIR1224

[16] Najari, S., Rafiei, D., Salehi, M., & Farahbakhsh, R. (2024). Adversarial botometer: adversarial analysis for social bot detection. *Social Network Analysis and Mining*, *14*(1), 220. https://doi.org/10.1007/s13278-024-01387-7

[17] Nayak, A., & Rahman, F. (2024). A Deep Learning-based Intelligent Automatic Detection and Classification of Fish Species in Marine Environment. *Natural and Engineering Sciences*, *9*(3), 144-153. https://doi.org/10.28978/nesciences.1606623

[18] Shirke, S., & Udayakumar, R. (2019, April). Lane datasets for lane detection. In *2019 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0792-0796). IEEE.

[19] Shukla, A., Jureček, M., & Stamp, M. (2024). Social media bot detection using dropout-gan. *Journal of Computer Virology and Hacking Techniques*, *20*(4), 669-680. https://doi.org/10.1007/s11416-024-00521-5

[20] Soares, H. (2022, April). The spread of fake news by social bots: Perspectives on social bot regulation. In *International Conference on Autonomous Systems and the Law* (pp. 189-204). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-47946-5_11

[21] Stević, I., Hadrović, S., & Jovanović, J. (2023). Environmental, social and other non-profit impacts of mountain streams usage as Renewable energy resources. *Archives for Technical Sciences*, *29*, 57-64. https://doi.org/10.59456/afts.2023.1529.057S

[22] Thavasimani, K., & Srinath, N. K. (2021). A custom classifier to detect spambots on cresci-2017 Dataset. In *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 1* (pp. 181-191). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-16-1338-8_16

[23] Tzoumanekas, G., Chatzianastasis, M., Ilias, L., Kiokes, G., Psarras, J., & Askounis, D. (2024). A graph neural architecture search approach for identifying bots in social media. *Frontiers in Artificial Intelligence*, *7*, 1509179. https://doi.org/10.3389/frai.2024.1509179

[24] Varol, O., Ferrara, E., Davis, C., Menczer, F., & Flammini, A. (2017, May). Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the international AAAI conference on web and social media* (Vol. 11, No. 1, pp. 280-289).

[25] Vij, P., & Prashant, P. M. (2024). Predicting aquatic ecosystem health using machine learning algorithms. *Int. J. Aquat. Res. Environ. Stud.*, *4*(S1), 39-44. https://doi.org/10.70102/IJARES/V4S1/7

[26] Wang, L., Qiao, X., Xie, Y., Nie, W., Zhang, Y., & Liu, A. (2023, October). My brother helps me: Node injection based adversarial attack on social bot detection. In *Proceedings of the 31st ACM International Conference on Multimedia* (pp. 6705-6714). https://arxiv.org/abs/2310.07159

[27] Wu, G., & Wang, X. (2025). A privacy-enhanced framework with deep learning for botnet detection. *Cybersecurity*, *8*(1), 9. https://doi.org/10.1186/s42400-024-00307-8

[28] Yang, K. C., Ferrara, E., & Menczer, F. (2022). Botometer 101: Social bot practicum for computational social scientists. *Journal of computational social science*, *5*(2), 1511-1528. https://doi.org/10.1007/s42001-022-00177-5

Authors Biography

N. Sheba Pari is a Ph. D. Research Scholar in the School of Computer Science Engineering and Information Systems (SCORE) in VIT, Vellore, India. Her research interest includes Information and Cyber Security, Social Networks, Artificial Intelligence and Network Engineering. She received her M.Tech in Software Engineering from Visvesvarya Technological University, Belgaum, India and B.E. in Information Science & Engineering from Visvesvarya Technological University, Belgaum, India. She has over 11 years of teaching experience as an Assistant Professor in various Engineering colleges.

Dr.K. Senthil Kumar, a distinguished professional with a Ph.D and extensive experience in both research and teaching, is a prominent figure in the fields of Cloud Computing and Machine Learning. His expertise is widely recognized, and he has a substantial publication record in esteemed international conferences and peer-reviewed journals. He currently holds the position of Professor at the School of Computer Sciences and Engineering (SCOPE), VIT University, Vellore. He also serves as a mentor to research scholars across diverse IT domains. As the corresponding author, he plays a pivotal role in coordinating and communicating research efforts.