Journal of Information Systems Engineering and Management
2025, 10(9s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Integrating Graph-Based Features with CNI-VIF for Enhanced Botnet Detection in Network Traffic

Anagha Patil1*, Arti Deshpande2
1 Assistant Professor, Department of Information Technology, Vidyavardhini's College of Engineering and Technology, Palghar, India. anagha.patil@vcet.edu.in.
2 Associate Professor, Computer Engineering Department, Thadomal Shahani Engineering College, Mumbai, India. arti.deshpande@thadomal.org.
**Corresponding author: Ms. Anagha Patil, anagha.patil@vcet.edu.in*

ARTICLE INFO

Received: 26 Oct 2024

Revised: 22 Dec 2024

Accepted: 01 Jan 2025

ABSTRACT

The growing threat of social botnets demands advanced detection techniques to identify sophisticated malicious activities within network traffic. This paper introduces a graph-based detection framework leveraging the Composite Node Information - Variance Inflation Factor (CNI-VIF) method for enhanced feature selection. By integrating traditional statistical metrics with graph-specific attributes like centrality measures, CNI-VIF effectively reduces dimensionality while preserving crucial features. The proposed methodology is validated using multiple machine learning models across CTU-13, IoT-23, and NCC-2 diverse botnet datasets, demonstrating superior accuracy, reduced computational overhead, and robust detection performance. The framework integrates machine learning models, counting Logistic Regression, Random Forest, SVM, Ensemble, FFNN, and Convolutional Neural Networks, achieving near-perfect detection rates with minimal false positives and false negatives. Furthermore, the proposed methodology substantially reduces computational time, up to 80%, compared to the state-of-the-art method, highlighting its suitability for real-time botnet detection in complex datasets. Comparative analysis confirms the methodology's advantage over existing state-of-the-art solutions, emphasizing its practical utility for real-time botnet detection.

Keywords: Social botnet detection, Graph-based feature selection, CNI, CNI-VIF, Machine learning

I. INTRODUCTION

In recent years, social botnets have posed significant challenges to network security, transforming the landscape of cyber threats [1,2]. Social botnets are sophisticated networks of compromised devices, often IoT devices [3], orchestrated to perform coordinated malicious activities. These botnets are insidious because they can mimic legitimate social behaviors, such as sending messages or engaging in online interactions, making them difficult to detect using traditional cybersecurity measures. The impact of social botnets on network security is profound; they can facilitate distributed denial-of-service (DDoS) attacks, spread misinformation, steal sensitive data, and manipulate online platforms for political or financial gain. As these botnets become more advanced, the need for robust detection strategies has become critical [4].

A botnet consists of connected compromised devices, frequently referred to as bots or zombies, that a botmaster or attacker controls through Command and Control (C&C) mechanisms. These devices are typically infected with malware and operate collectively to perform malevolent activities such as spreading malware, spamming, data theft, data manipulation, and DDoS attacks [5]. Botnets vary in size and complexity, ranging from a few devices to millions, and often employ sophisticated evasion techniques like encrypted communication, peer-to-peer (P2P) networking, or domain generation algorithms (DGAs) to avoid detection and maintain persistence [6].

Researchers have contributed with plentiful solutions such as firewalls, honeypots, intrusion detection systems (IDS), and encryption to handle cyber-attacks. An IDS can detect botnet activities such as abnormal traffic patterns, unauthorized access attempts, and suspicious communication to C&C servers. Signature-based IDS recognizes

familiar botnets using predefined patterns, whereas anomaly-based IDS detect deviations from normal behavior, enabling the identification of unknown or emerging threats [7,8]. By integrating advanced techniques like machine learning and behavioral analysis, IDS enhances its ability to detect stealthy botnets and provides alerts for timely mitigation, thus serving as a key defense mechanism in securing networks against botnet-related threats.

To effectively detect social botnets, it is essential to analyze the relationships and interactions between network entities, such as IP addresses, devices, and users. Graph databases play a pivotal role here as they follow model graph properties. Graph databases shine at modeling and querying relationships, which makes them ideal for representing network traffic. For example, graphs, where nodes represent devices or IP addresses and edges, denote communication or data transfer between them [9]. By leveraging graph databases, cybersecurity experts can uncover anomalies and patterns that indicate the presence of a botnet, such as unusual clustering of communication between nodes or the emergence of new, suspicious connections.

Graph databases enable a deeper understanding of the network structure, allowing for identifying central nodes that may play a crucial role in the botnet's operation. In a botnet, specific nodes may act as C&C servers, directing the activities of other compromised devices. By analyzing the centrality measures [10] within a graph database, these critical nodes can be identified and targeted for disruption, impeding the botnet's ability to function.

However, using graph databases for botnet detection introduces new challenges, mainly feature selection and dimensionality reduction [11]. As network traffic data is converted into a graph structure, the resulting dataset can become highly dimensional, with numerous features representing different aspects of the network's topology and traffic patterns [12,13]. Selecting the most relevant features for botnet detection is crucial, as redundant or irrelevant features can amplify computational intricacy and reduce model performance. Traditional feature selection techniques may need to be revised when applied to graph-based data due to their inability to adequately capture the complex relationships inherent in graphs [14]. These methods often overlook important graph-specific features, such as centrality measures, which are critical for understanding the influence and connectivity of nodes within a network.

Given the limitations of traditional feature selection methods, a more sophisticated approach is needed that can effectively handle the unique characteristics of graph databases. The Composite Node Information—Variance Inflation Factor (CNI-VIF) method [15] offers a promising solution by integrating traditional VIF with graph-specific features, such as centrality measures. CNI-VIF not only addresses multicollinearity among predictor variables but also ensures that critical graph-based features are retained, improving the accuracy and efficiency of botnet detection models.

In this paper, the application of CNI-VIF to the detection of social botnets within network traffic is explored. By leveraging the strengths of graph databases and the advanced feature selection capabilities of CNI-VIF, the aim is to enhance the detection of social botnets, providing a powerful tool for network security professionals to combat this growing threat. The contributions of this study are:

- An enhanced graph-based botnet detection system is presented, which detects diverse botnets and behavioral characteristics.
- To prove the significance of the CNI-VIF feature selection algorithm for botnet detection in graph datasets, various feature selection algorithms, such as PCA, RFE, VIF, and CNI-VIF, are compared.
- Different machine learning (ML) models such as Logistic regression, Random Forest, Support Vector Machine (SVM), Ensemble model, Feedforward Neural Networks (FFNN), and Convolutional Neural Networks (CNN) are used to assess the proposed method for botnet detection.
- The proposed framework is validated on three real botnet datasets of varying dimensions and volume.
- The proposed graph-based botnet detection framework is compared to a state-of-the-art graph-based botnet detection system in terms of detection rate, Number of FPs, Number of FNs, and running time.

The introduction is covered in Section I. The former sections of the paper are systematized as follows: Section II offers a brief literature examination of present botnet detection systems and climaxes limits of the state-of-the-art. Section III describes the design of the proposed framework with all integral components. Section IV evaluates the assessment results of the proposed methodology in detecting botnets with the help of ML models. Also, it compares the proposed methodology with state-of-art methods. In Section V, the contribution and future research directions are provided, as well as a summary of this paper.

II. RELATED WORK

Bot malware and Botnet detection have been areas of interest, and a vast number of research papers [16-18], including review papers in recent years [19-21], prove this. Botnet detection methods can be roughly categorized into signature-based, anomaly-based, and DNS-based.

Signature-based botnet detection is a traditional yet effective technique in cybersecurity, where predefined patterns, also known as signatures, are used to identify malicious activities associated with botnets. These signatures typically consist of unique characteristics extracted from known botnet behaviors, such as specific communication protocols, payload structures, or sequences of commands used by botmasters to control infected devices. The approach in [22] and [23] compares network traffic or system logs against a database of botnet signatures. The corresponding activity is flagged as potentially malicious when a match is found. [22] works for snort rules and [23] exploits traits of modern DDoS attacks. Signature-based detection identifies known botnet variants with high accuracy and low false positive rates, relying on precise pattern matching. Despite its strengths, this method has limitations, particularly in detecting new or evolving botnets, which may use polymorphic techniques, encrypted communication channels, or other evasion strategies to circumvent detection.

An anomaly-based detection is a dynamic approach that identifies deviations from normal network behavior to uncover botnet-related activities. Unlike signature-based methods, which rely on predefined patterns, anomaly detection leverages statistical analysis, machine learning, and behavioral modeling to detect previously unknown or evolving botnets [24,25]. This approach operates at multiple granularities, including packet-level and flow-level anomaly detection.

At the packet level, this method examines individual data packets for unusual attributes such as size, header anomalies, or payload irregularities [26,27,28]. It identifies signs of malicious activities, such as malformed packets, unexpected protocol usage, or sudden spikes in traffic volume. Spiekermann et al. [26] proposed unsupervised packet-level anomaly detection, which analyzes packets using IsolationForest and LocalOutlierFactor algorithms. However, the method generates high false positives and false negatives in a dynamic environment. In [27], authors inspected packets to perceive payload anomalies using deep learning. Their block sequence construction method constructs the expression of payload, depicting short-term and long-term dependency relationships amongst block sequences. Authors [28] proposed two-staged packet-level anomaly detection to inspect packet bytes to flag events. But, packet-level detection is highly granular, offering precise insights into specific irregularities that may indicate botnet communications. However, it can generate high volumes of data, making it computationally intensive for large-scale networks.

On the other hand, Flow-level anomaly detection aggregates and analyzes traffic flows, representing sequences of packets with shared features like source and destination IPs, ports, and protocols. It identifies abnormal traffic patterns, such as unusually high connection rates, irregular session durations, or deviations in bandwidth usage. Flow-level detection efficiently identifies distributed botnets and C&C traffic, as it captures broader behavioral trends in network traffic. Recently, more research has been done on Flow-level anomaly detection using machine learning (ML), reinforcement learning (RL), and graph-based approaches. ML-based detection involves supervised, unsupervised, or semi-supervised algorithms to classify normal and anomalous traffic [29-32].

Hostiadi et al. [29] proposed a bot activity detection model using time partitioning. Using chain trace, the authors applied their method in the CTU-13 dataset to find similar activities in every time segment. In [30], authors proposed a novel B-Corr similarity measure that calculates similarity among bot activities using co-relation and probabilities. They applied it on CTU-13 using various ML models to specify a list of suspected Ips and normal activities. They further proposed a new method to detect linkages between bot activities in [31]. [32] extends their work to analyze bot's communication behavior on network traffic as centralized, distributed, and spread. Various clustering and classification algorithms are used to analyze flow features. These models learn from historical data to detect outliers or unusual patterns that may signify botnet traffic, including stealthy C&C communications. However, these methods could be more scalable due to increased dimensionality.

RL enhances anomaly detection by enabling adaptive learning in dynamic network environments. The study [33] proposed Gym-plus, a new RL model to generate new malware samples that can evade ML detection. The authors trained their model based on the new samples to increase the detection rate. Alauthman et al. [34] proposed RL-

based detection to detect known and unknown bots in P2P networks in online and offline phases. Here, RL agents interact with network flow data, receiving rewards for accurate anomaly identification and penalties for false detections. So, this approach is ineffective in environments with evolving botnets, as RL can continuously learn and optimize detection policies without extensive labeled data. Alavizadeh et al. [35] used deep Q-Learning to allow the system to adapt to new real-time attack patterns with a trial-error approach. They investigated several hyper-parameters of the agent to fine-tune for network intrusion. The method needs to adapt to the evolving tactics of social botnets, which can rapidly change their behavior to evade detection.

Graph-based flow-level anomaly detection employs graph theory to model network traffic as a graph, where nodes represent IPs, and edges represent communication flows between them [36-38]. This approach captures network traffic's structural and relational patterns, making it well-suited for detecting botnets and other complex anomalies. Machine learning enhances this method by analyzing graph-derived features like centrality, clustering coefficients, and community structures. Techniques such as Graph Neural Networks (GNNs), spectral clustering, and graph-based anomaly detection algorithms are used to identify suspicious nodes or edges indicative of anomalous behaviors. These models can detect patterns such as highly connected nodes, which can be potential C&C servers or unusual flow distributions. By focusing on the relationships and behaviors within the network, graph-based flow-level anomaly detection excels in uncovering stealthy botnets that evade traditional flow-based methods. Its ability to generalize to unseen data and uncover hidden patterns makes it a robust tool for modern network security.

Chowdhury et al. [36] used graph-based features to detect botnet on network traffic. The self-organizing map (SOM) method is applied to the CTU-13 dataset after extracting graph features to obtain clusters of nodes. Authors believed dense clusters imply normal behavior, whereas smaller clusters imply malicious behavior. Further, to reduce detection overhead, dense clusters are removed. After applying statistical measures, the remaining smaller clusters are further classified as benign or malicious. But, in the case of previously unknown attacks, the method could be more realistic and error-prone for more extensive networks. In [37], researchers proposed a bot detection system in two phases based on a graph and applied on CTU-13, where the first phase is unsupervised, which prunes normal hosts using clustering, and the second phase detects bots using ML algorithms, which are supervised. Authors again preferred SOM in the first phase to have malicious and benign clusters and proved their system robust against unknown attacks and more extensive networks. Zhou et al. [38] proposed GNN-based botnet detection, which can detect previously unknown bots along with known bots in a P2P network. The authors believed GNNs could capture structural properties in centralized and decentralized networks. However, challenges include computational complexity and scalability, especially in large, dynamic networks.

DNS-based botnet detection leverages the Domain Name System (DNS) to identify malicious botnet activity by analyzing DNS queries and responses. Botnets often rely on DNS to resolve domain names for their C&C servers, making DNS traffic an essential data source for detection. This approach examines DNS features such as query volume, domain names, response times, and TTL (Time-To-Live) values. Suspicious patterns include unusually high query rates, resolution of dynamically generated domain names (DGAs), or queries to known malicious domains [39-41]. By detecting these anomalies, DNS-based methods can identify botnets at an early stage, even before they execute malicious payloads. BotGAD [39] was proposed for real-time group activity botnet detection on DNS traffic. BotGAD and Botsniffer [40] were designed to detect synchronized botnet communication, except for the former, which worked on similarity estimation methods and later on string matching. Techniques such as machine learning enhance the effectiveness of DNS-based detection by classifying DNS traffic into normal and anomalous categories. Algorithms analyze features like entropy, lexical patterns of domain names, and frequency of requests to detect potential threats. DNS-based detection is efficient, scalable, and remarkably effective for identifying botnets during their communication or propagation phases. However, it may face challenges with false positives or botnets that use fast flux, peer-to-peer (P2P) communication, or IP-based connections instead of DNS [41].

Traditional methods for social botnet detection have evolved, employing a combination of behavioral analysis, anomaly detection, and machine learning models [42]. Despite these advancements, detecting social botnets remains a complex task. The ability of these botnets to mimic legitimate user behavior, coupled with the sheer volume of network traffic on social platforms, necessitates the development of more sophisticated detection techniques.

III. PROPOSED METHODOLOGY

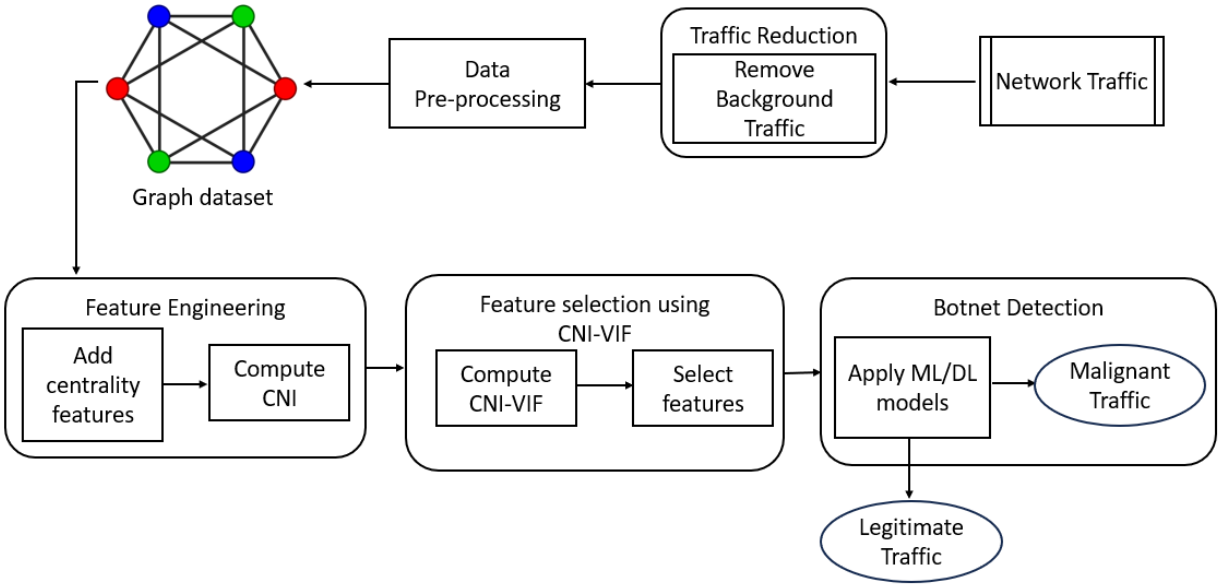


Fig. 1. The architecture of the proposed Botnet Detection System

As represented in Figure 1, the proposed architecture is intended to detect social botnets using a novel feature selection technique, CNI-VIF. The proposed architecture includes components: Data Bootstrap and Pre-processing, Feature Engineering, Feature Selection using CNI-VIF, and Botnet Detection. The system leverages graph-based representations of network traffic data to capture the complex relationships between nodes, such as IP addresses or devices, within the network. The components are discussed in the following sections.

A. DATA BOOTSTRAP AND PRE-PROCESSING

1) Network Traffic Consumption

The goal of the network traffic consumption stage is to turn bidirectional network flows into a list L so that they can be represented graphically. In this study, every node stands for a distinct IP address, and every edge denotes a link between two IP addresses. A list L created from network traffic has two tuples, Sip and Dip, representing source and destination IP addresses, respectively.

$$L = \{Sip, Dip\} \quad (1)$$

2) Traffic Reduction

Network traffic data can be noisy and vast, making it challenging to recognize relevant patterns. Traffic reduction techniques are applied to filter out irrelevant or redundant traffic, focusing on the most critical communication flows. This step reduces the size of the dataset and the computational burden on subsequent processes. The dataset contains a lot of background traffic that is useless for the botnet detection model; hence, this traffic is eliminated.

3) Data Pre-processing

This step is vital for ensuring that the data fed into the feature selection and detection model is high quality and relevant to the task at hand. Network Traffic Flow (NTF) with features can be represented as NTF = $\{f_1, f_2, \ldots\ldots, f_N\}$ where N indicates the total number of features, which can be categorical or numerical. Hence NTF can be rewritten as NTF = $\{f_{c1}, f_{c2}, \ldots., f_{cp}, f_{n1}, f_{n2}, \ldots\ldots, f_{nq}\}$ where

$$N \text{ (Total features)} = P \text{ (categorical features)} + Q \text{ (numeric features)} \quad (2)$$

One crucial part of pre-processing is to convert all fc into fn so that the proposed model works efficiently. Before this, missing and duplicate values are handled.

4) Graph Transform

The graph G is generated for network traffic after traffic reduction and pre-processing phases. G is a directed graph with V as a set of vertices and E as a set of edges. V is thus a union of Source ip (Sip) and Destination ip (Dip) represented as

$$V = (Sip\ U\ Dip) \quad (3)$$

For every duo of connected vertices in V, where Dipx = Vj and Sipx = Vi, directed edges ei,j, and ej,i exist. so that,

$$E = (Sip_x, Dip_x) \cup (Dip_x, Sip_x) \quad (4)$$

Once the network traffic is transformed in graph G, graph-based features are added.

B. FEATURE ENGINEERING

Feature engineering comprises altering raw data into meaningful features that improve the performance of machine learning models. Effective feature engineering improves model accuracy and interpretability, reduces dimensionality, ensures computational efficiency, and aligns the feature set with the specific objectives of botnet detection.

1) Add Centrality Features

This study uses graph-based characteristics, specifically centrality metrics, including betweenness, closeness, and Degree centrality.

- Betweenness Centrality (BC): Calculates along the shortest path between two other nodes, i.e., how many times a node acts as a bridge.
- Closeness Centrality (CC): Signifies closeness of a node to other nodes in the graph.
- Degree Centrality (DC): Calculates the number of direct connections of a node.

These features are essential to understanding the role of each node in the network and its potential involvement in botnet activities. Nodes with high BC often serve as key communication hubs or relay points, which can correspond to command-and-control servers or heavily connected malicious nodes in a botnet. High CC nodes are strategically positioned to efficiently spread information or malicious payloads across the network. Nodes with high DC often have more opportunities to interact with other nodes, making them central to the overall activity of the network. So, now only one graph-based composite feature, CNI, is added with the original features.

2) Compute CNI

Instead of using three separate centrality metrics as features, Composite Node Information (CNI) aggregates them into a single value, reducing dimensionality and redundancy in the feature set. As given in equation 4, CNI is calculated for each node in the graph, capturing key graph-based characteristics such as centrality, connectivity, and influence. Incorporating CNI into the CNI-VIF method ensures that graph-based features are appropriately weighted and selected, enhancing the model's predictive power while reducing redundancy.

$$CNI = \frac{BC+CC+DC}{3} \quad (4)$$

C. FEATURE SELECTION USING CNI-VIF

Feature selection is crucial to investigate and assess how well graph-based features identify and differentiate between a bot and benign nodes. The network flow graph produced in the previous phase is utilized to retrieve a collection of characteristics based on graphs. In graph databases, nodes represent entities, and edges represent their relationships. These relationships are crucial for understanding the structure and dynamics of the graph, making traditional feature selection methods inadequate. The core of the proposed architecture is the feature selection process, where CNI-VIF is used to identify the most appropriate features for botnet detection. The CNI-VIF equation enhances traditional feature selection by integrating graph-based features into the VIF calculation.

CNI-VIF integrates the traditional VIF approach with graph-specific features, particularly centrality measures. Centrality measures such as Betweenness, Closeness, and Degree centrality assess the importance and influence of

nodes within the graph. By incorporating these measures into the VIF calculation, CNI-VIF ensures that important graph-based features are retained while addressing multicollinearity.

1) Compute CNI-VIF

Once the CNI values are computed, average_CNI ($\overline{CNI}$) is calculated for Source and destination IPs, and they have integrated into the traditional Variance Inflation Factor (VIF) calculation as shown in equation (5).

$$CNI - VIFi = \frac{1}{1-Normalised(Ri^2 + \alpha * \overline{CNI})} \quad (5)$$

where:

- R_i^2 is the coefficient of determination.
- $\overline{CNI}$ is the mean value of the average_CNI for the dataset.
- α is a parameter that adjusts the influence of the graph-based feature.

A high R_i^2 indicates that feature i is highly correlated with others, suggesting redundancy. By combining R_i^2 with the normalized centrality score, $\overline{CNI}$, the method evaluates a feature's correlation and significance in the network context. The parameter α balances the influence of the centrality measure relative to the R_i^2 term. The term $Normalised(Ri^2 + \alpha * \overline{CNI})$ ensures that the combined effect of redundancy and centrality is scaled appropriately, allowing the remainder always to be positive and non-zero.

2) Select Features

The CNI-VIF equation considers both the standard features and the graph-specific CNI features. This results in a more accurate assessment of multicollinearity, ensuring that the selected features are relevant and non-redundant.

Features are selected based on their CNI-VIF scores. Typically, a threshold is set, and features with CNI-VIF scores above this threshold are eliminated. This process retains structurally significant and non-redundant features, ensuring the final feature set is optimized for the botnet detection task.

D. BOTNET DETECTION

To evaluate the efficacy of graph-based features in detecting botnets, the selected features are used as inputs for various ML models to identify malicious traffic and detect botnet activity. The proposed model classifies the incoming network traffic into malignant and legitimate. Several powerful ML algorithms, including Logistic Regression, Random Forest, and SVM, are investigated to identify botnet behavior. These models are known for their interpretability and are trained on the selected features to categorize network traffic as benign or malicious. Furthermore, DL models, such as FFNN and CNN, capture more complex patterns in the data.

The botnet detection component returns malignant traffic as a graph that is a subset of the original network traffic graph. The original graph encompasses all traffic, including benign and malicious interactions, with a dense structure and a wide range of nodes and edges. After applying CNI-VIF for feature selection, effectively reducing dimensionality while retaining significant graph-based features, the detected botnet graph isolates the subset of nodes and edges involved in malicious activities.

IV. PROPOSED EXPERIMENTAL SETUP

A. ENVIRONMENT

Hardware:

The experiments are conducted for data pre-processing, features engineering, visualization, model training, and validation. Table 1 indicates the specifications of Hardware and Software used to implement the proposed methodology.

Table 1. Specification of Hardware and Software

Item	Description
Processor	I7
Memory	16 GB
GPU	Nvidia GeForce Trx 4080
Python	3.8

Software:

The entire implementation of the proposed methodology is done in Python. To convert network flows into graphs, Graph-tool [43] is used, which also extracts multiple graph-based features. Graph-tool is a free, Python-based, and widely used module for statistical analysis of graphs. To implement CNI-VIF, listed ML models, and different feature evaluation measures, several Python libraries, such as Pandas, NumPy, Matplotlib, sci-kit, time, and Keras, are used.

B. DATASETS

To test the proposed framework from a broader perspective, three datasets, CTU-13 [44], IOT-23 [45], and NCC-2 [46], with diverse attack scenarios, are considered. All the datasets contain network traffic from infected ips from various botnet families and are publicly available. All three datasets contain scenarios with benign and malicious traffic from numerous botnet families. The NCC-2 dataset is a binetflow file that simulates botnet attacks with simultaneous attack characteristics based on CTU-13 and NCC [47] datasets. NCC-2 covers sporadic attacks of CTU-13 and periodic attacks of NCC. The dataset is simulated using three sensors with more than one type of botnet simultaneously and, hence, is suitable for developing a distributed botnet detection model. Table 2 describes the diversity in the datasets used.

Table 2. Details of Datasets Used

Dataset	**Size (in GB)**	**No. of botnet tools used for attack**	**Periodicity of Activity**
CTU-13	1.9	7	Sporadic
IoT-23	20	13	Sporadic
NCC-2	45	7	Periodic + Sporadic

C. PERFORMANCE METRICS

As all three datasets are labeled, performance metrics such as Accuracy, Precision, Recall, and F1 measure are considered to assess the proposed methodology's performance. The proposed method is assessed using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), which can be defined as:
TP is when the proposed framework correctly predicts Botnet (Malevolent samples) from network traffic.
FP is when the proposed framework wrongly predicts Botnet (Malevolent samples) from network traffic.
TN is when the proposed framework correctly predicts Benign (Normal samples) from the network traffic.
FN is when the proposed framework wrongly predicts Benign (Normal samples) from the network traffic.

These evaluation measures are represented as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (7)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (8)$$

$$F1\ Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (9)$$

V. RESULTS AND DISCUSSION/ EXPERIMENT EVALUATION

The experiments evaluate the proposed botnet detection system with various feature selection techniques like VIF, PCA, RFE, and CNI-VIF on CTU-13, IoT-23, and NCC-2 datasets. This will give insights into how CNI-VIF is better than listed feature selection algorithms in terms of running time and performance metrics. Later, the proposed framework is compared with state-of-art work [13], which also uses graph features to detect the botnet. In [13], filter-based feature evaluation techniques are used for two datasets: CTU-13 and IoT-23. Multiple graph-based features increase dimensionality, and grouping them into multiple feature sets increases computational time. Here, in addition to the listed two datasets, the NCC-2 dataset is used to justify the efficiency of the proposed framework. The proposed framework achieves dimensionality reduction with the help of CNI and reduced computational time with the CNI-VIF algorithm for feature selection.

A. DATA BOOTSTRAP AND PRE-PROCESSING

Table 3 indicates the details after the Traffic Reduction phase. As the IoT-23 dataset has no background traffic, the number of tuples remains unchanged after the traffic reduction phase. All the necessary pre-processing is done on all three datasets, as mentioned in the pre-processing section. While doing this, all the missing values are handled by dropping the rows of the respective dataset. Also, required normalization and scaling are done on various attributes.

Table 3. Detail activity recorded for datasets after removing background traffic

Dataset	Original Traffic	After Background traffic removal			% of tuple reduction
		Normal Traffic	Botnet Traffic	Total	
CTU-13	2,950,000	261,354	126,762	388,116	86.84
IoT-23	1446639	1,246,861	199,778	1,446,639	NA
NCC-2	14,779,085	193,755	804,002	997,757	93.25

The network traffic is then converted into graphs using the Graph tool to add graph-based features. Figure 2 show graph representations for the network traffic and dynamics in the three datasets. The graph for the entire dataset is too dense and not interpretable; hence, 25000 random samples from every dataset are considered for experiment purposes. Red (malicious traffic) and blue (benign traffic) colors help differentiate between distinct traffic types. This categorization enhances the graph's interpretability, separating normal and abnormal behavior. The edges among the nodes show communication from one ip to another ip.

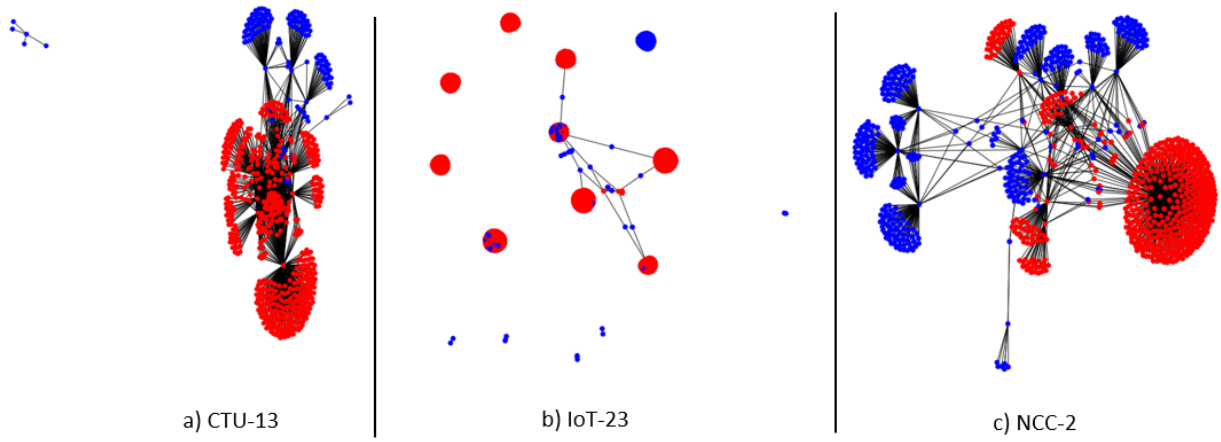


Fig. 2. Graph representation for a) CTU-13 b) IoT-23 and c) NCC-2

B. FEATURE ENGINEERING AND FEATURE SELECTION

The proposed framework uses the CNI-VIF method for feature selection, which proved more efficient than other feature selection algorithms for graph databases [15]. In the context of the CNI-VIF equation, a threshold of 10 is commonly used as a benchmark to select features, aligning with traditional VIF (Variance Inflation Factor) practices where values above 10 indicate high multicollinearity. For equation (5), this threshold ensures that features with a combined influence of redundancy and centrality significance that result in a normalized score leading to CNI-VIF_i >10 are excluded. This effectively removes features that are either highly redundant or have minimal structural importance, ensuring the retained features are independent and critical for the task. The threshold thus balances statistical reliability with graph-based importance, optimizing the feature set for performance in tasks like botnet detection. Table 4 indicates all features and selected features for all datasets using CNI-VIF.

Table 4. Features selected by CNI-VIF

Features	CTU-13	IoT-23	NCC-2
Original Features	**Dur**	Ts	**StartTime**
	Proto	uid	**Dur**
	SrcAddr	**id.orig_h**	Proto
	Sport	**id.orig_p**	**SrcAddr**
	Dir	**id.resp_h**	**Sport**
	DstAddr	**id.resp_p**	Dir
	Dport	**proto**	**DstAddr**
	State	**service**	**Dport**
	sTos	**duration**	**State**
	dTos	**orig_bytes**	sTos
	TotPkts	**resp_bytes**	dTos
	TotBytes	conn_state	TotPkts
	SrcBytes	local_orig	TotBytes
	Label	local_resp	**SrcBytes**

	Train	missed_bytes	**Label**
	StartTime	**history**	**ActivityLabel**
	ActivityLabel	orig_pkts	BotnetName
		orig_ip_bytes	SensorId
		resp_pkts	
		resp_ip_bytes	
		ActivityLabel	
Graph-based Feature	**CNI**	**CNI**	**CNI**

C. BOTNET DETECTION

The following section describes the results of experiments on CTU-13, IoT-23, and NCC-2 datasets.

1) CTU-13:

Figure 3 compares the accuracies of various ML models, such as Logistic Regression, Random Forest, SVM, Ensemble Voting, FFNN, and CNN, when different feature selection techniques like VIF, CNI-VIF, PCA, and RFE are applied to the CTU-13 dataset. Notably, models using VIF and CNI-VIF consistently achieve near-perfect accuracy (1.0) across most cases, highlighting their effectiveness in selecting relevant and independent features. In contrast, PCA and RFE show slightly lower accuracy for specific models, such as PCA with SVM (0.83) and RFE with Logistic Regression (0.83). This indicates that while PCA and RFE can reduce dimensionality, they may lose critical features, which affect classification accuracy. The consistent performance of CNI-VIF suggests this technique is better suited for preserving the importance of statistical and structural features, particularly for botnet detection in this dataset.

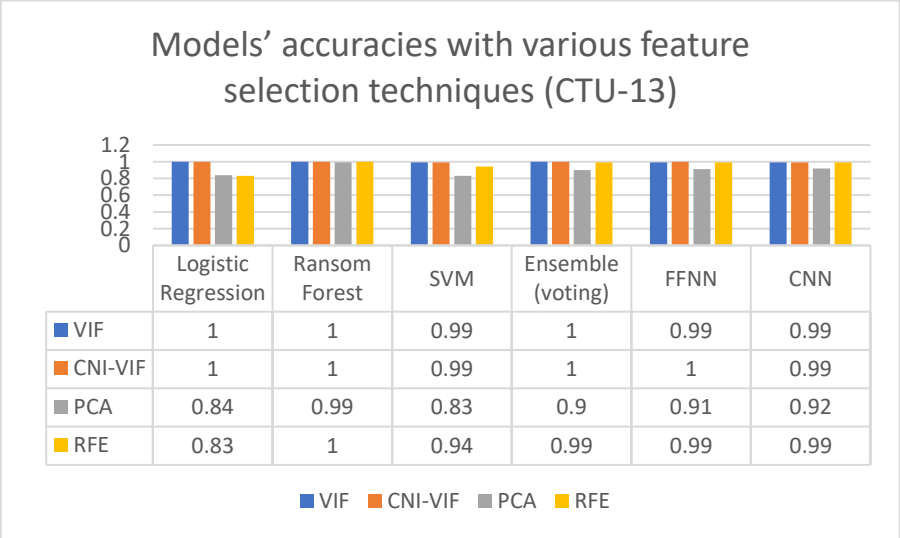
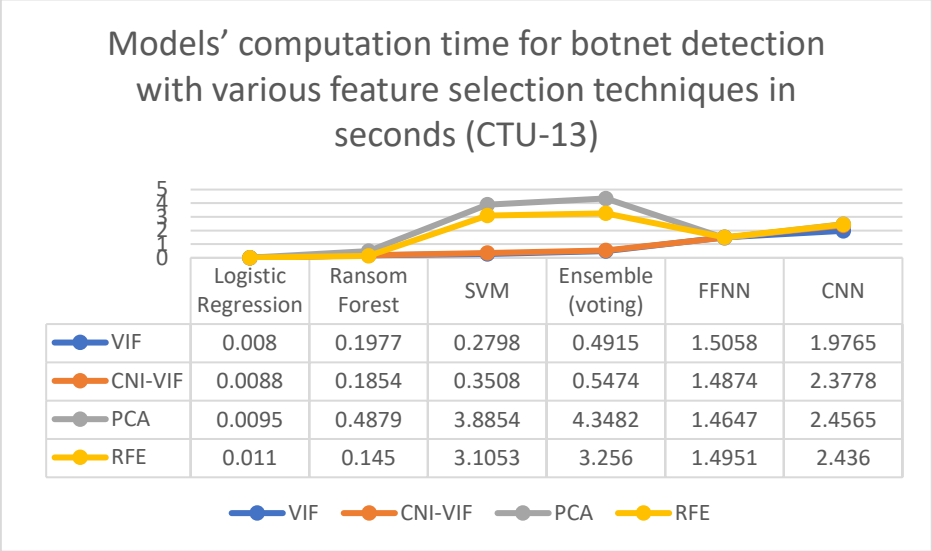


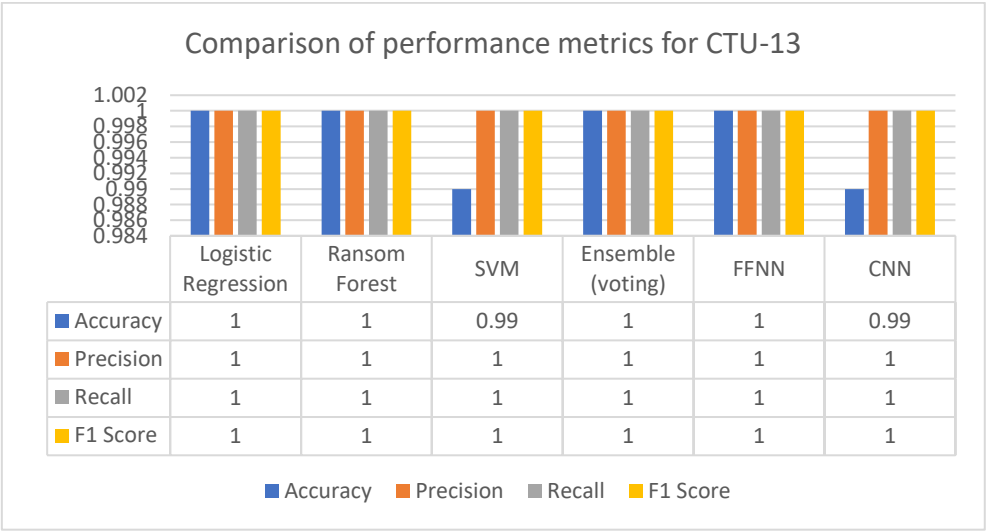
Fig. 3. Comparison of models' accuracies with various feature selection techniques (CTU-13)



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
VIF	0.008	0.1977	0.2798	0.4915	1.5058	1.9765
CNI-VIF	0.0088	0.1854	0.3508	0.5474	1.4874	2.3778
PCA	0.0095	0.4879	3.8854	4.3482	1.4647	2.4565
RFE	0.011	0.145	3.1053	3.256	1.4951	2.436

Fig. 4. Comparison of models' computation time for botnet detection with various feature selection techniques (CTU-13)

Figure 4 illustrates the computation time in seconds for various ML models when different feature selection techniques are applied to the CTU-13 dataset. It is evident that VIF consistently incurs the lowest computation time across all models, followed closely by CNI-VIF, demonstrating their computational efficiency. In contrast, PCA exhibits the highest computation time, particularly for models like SVM (3.8854 seconds) and Ensemble Voting (4.3482 seconds), which suggests a significant overhead due to its complex dimensionality reduction process. RFE also shows relatively higher computation times than VIF and CNI-VIF but is generally faster than PCA. This comparison highlights that VIF and CNI-VIF maintain high accuracy and ensure faster computation, making them ideal for real-time applications such as botnet detection.



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
Accuracy	1	1	0.99	1	1	0.99
Precision	1	1	1	1	1	1
Recall	1	1	1	1	1	1
F1 Score	1	1	1	1	1	1

Fig. 5. Comparison of performance metrics using CNI-VIF for botnet detection

Figure 5 presents a comparative analysis of performance metrics for various machine learning models applied to the CTU-13 dataset. Logistic Regression, Random Forest, Ensemble Voting, FFNN, and CNN achieve perfect scores of 1 across all metrics, indicating their robustness and reliability in detecting botnet traffic. SVM demonstrates slightly lower performance with an accuracy and F1 score of 0.99, while maintaining a precision and recall of 1. This suggests that while SVM occasionally misclassifies traffic, its ability to identify botnet and normal traffic correctly is still highly reliable. The results highlight the efficacy of the proposed methodology in achieving near-perfect detection across multiple models, reinforcing the reliability and precision of the approach for botnet detection in network traffic.

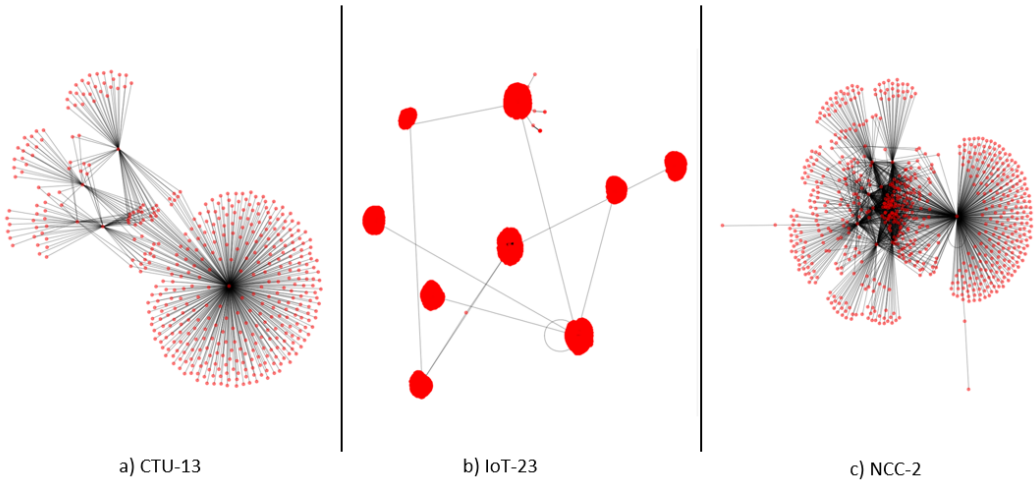
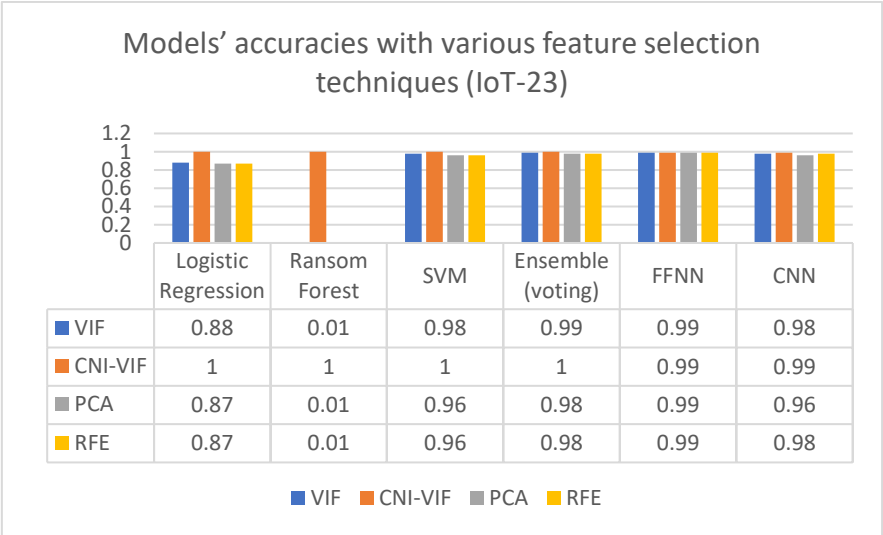


Fig. 6. Detected Botnet a) CTU-13 b) IoT-23 and c) NCC-2

As indicated in Figure 6, the detected botnet graph is a subgraph derived from the original network traffic graph, representing only the malicious nodes and their connections identified during the botnet detection process. These subgraphs are typically less dense, indicating botnet nodes' focused and distinct communication patterns compared to the broader and more diverse interactions in the original traffic graph. The ability to extract this subgraph highlights the effectiveness of graph-based features, CNI, and CNI-VIF in identifying botnet-related traffic within complex network environments.

2) IoT-23:



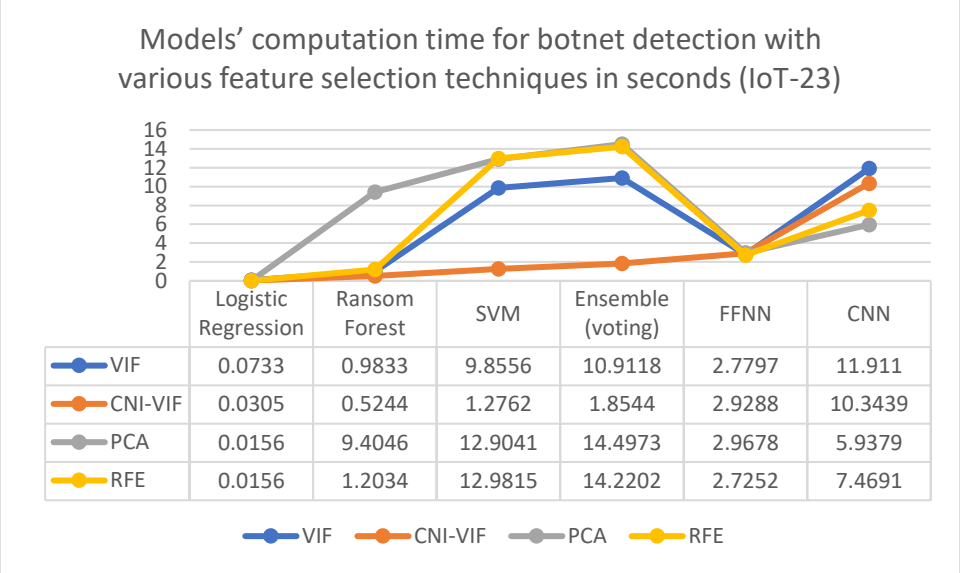
	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
VIF	0.88	0.01	0.98	0.99	0.99	0.98
CNI-VIF	1	1	1	1	0.99	0.99
PCA	0.87	0.01	0.96	0.98	0.99	0.96
RFE	0.87	0.01	0.96	0.98	0.99	0.98

Fig. 7. Comparison of models' accuracies with various feature selection techniques (IoT-23)

Figure 7 highlights the comparative performance of various models using different feature selection techniques: VIF, CNI-VIF, PCA, and RFE, based on accuracy, precision, recall, and f1 score. CNI-VIF consistently outperforms or matches other techniques, achieving a perfect score (1.0) across most models, particularly in Logistic Regression, Random Forest, SVM, and Ensemble Voting. In contrast, VIF, PCA, and RFE exhibit minor deviations, with PCA and RFE slightly underperforming for Logistic Regression, SVM, and CNN models. Random Forest shows significant differences, with CNI-VIF yielding a perfect score while other techniques remain much lower. This indicates that CNI-VIF is robust across various models and ensures consistently high performance compared to traditional feature selection methods.

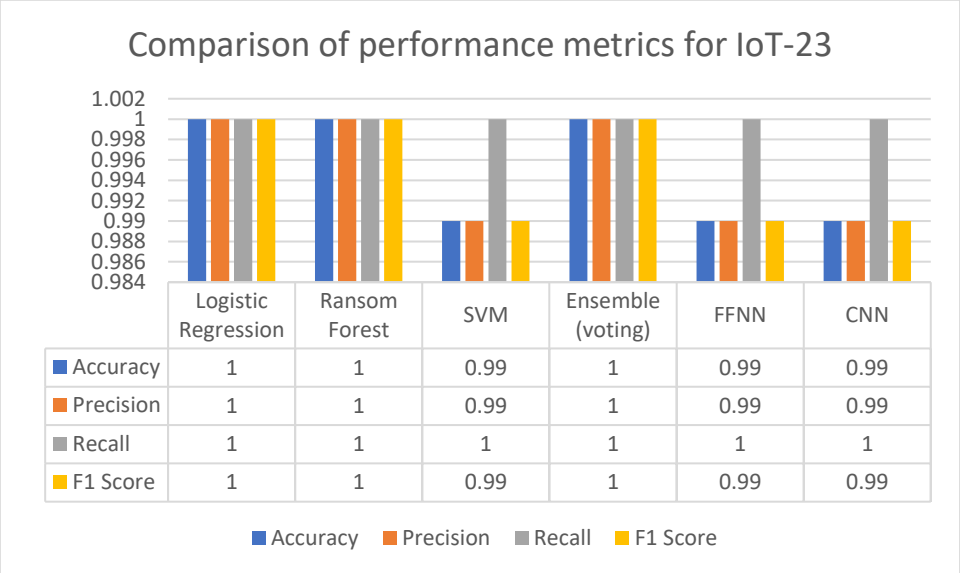
Figure 8 illustrates the computation times of various models using different feature selection techniques: VIF, CNI-VIF, PCA, and RFE. Across most models, CNI-VIF demonstrates a significant reduction in computation time

compared to PCA and RFE, particularly in computationally intensive models such as Random Forest, SVM, and Ensemble Voting. For example, in SVM, CNI-VIF's computation time (1.2762) is substantially lower than that of PCA (12.9041) and RFE (12.9815), reflecting its efficiency in feature selection. Similarly, for Ensemble Voting, CNI-VIF achieves a much lower computation time (1.8544) compared to PCA (14.4973) and RFE (14.2202). While VIF also shows low computation times, it often underperforms in accuracy compared to CNI-VIF. These results highlight the effectiveness of CNI-VIF in optimizing computational resources while maintaining or improving the performance of machine learning models.



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
VIF	0.0733	0.9833	9.8556	10.9118	2.7797	11.911
CNI-VIF	0.0305	0.5244	1.2762	1.8544	2.9288	10.3439
PCA	0.0156	9.4046	12.9041	14.4973	2.9678	5.9379
RFE	0.0156	1.2034	12.9815	14.2202	2.7252	7.4691

Fig. 8. Comparison of models' computation time for botnet detection with various feature selection techniques (IoT-23)



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
Accuracy	1	1	0.99	1	0.99	0.99
Precision	1	1	0.99	1	0.99	0.99
Recall	1	1	1	1	1	1
F1 Score	1	1	0.99	1	0.99	0.99

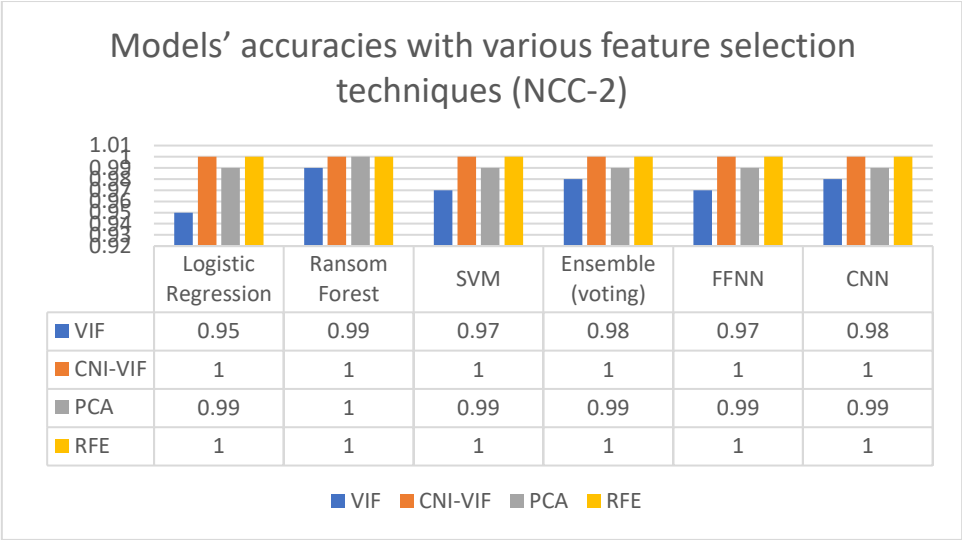
Fig. 9. Comparison of performance metrics using CNI-VIF for botnet detection

Figure 9 showcases the performance metrics after applying the proposed methodology using various ML models. Models such as Logistic Regression, Random Forest, and Ensemble Voting achieve perfect scores of 1 across all metrics, indicating their exceptional ability to accurately classify and detect botnet traffic without false positives or false negatives. Other models, including SVM, FFNN, and CNN, exhibit slightly lower scores, with values of 0.99 for accuracy, precision, and F1 score, while still achieving perfect recall (1), suggesting they effectively detect all botnet

instances. These results emphasize the efficacy of CNI-VIF for feature selection, demonstrating high reliability and precision in botnet detection tasks.

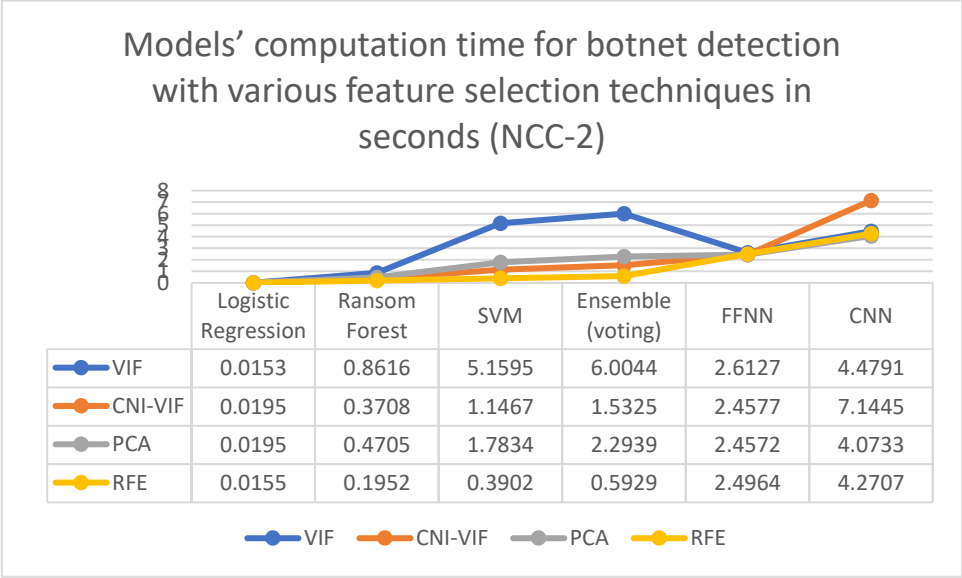
3) NCC-2:

Figure 10 demonstrates that the CNI-VIF feature selection consistently achieves the highest performance across all models, with a perfect score of 1 in most cases, signifying optimal feature selection and classification. Other techniques, such as PCA and RFE, also perform well, with scores close to or equal to 1 for Random Forest and Ensemble Voting but slightly lower for other models. VIF exhibits lower scores than CNI-VIF in several cases, underscoring the enhanced capability of CNI-VIF in improving feature relevance and model accuracy. This comparison emphasizes the superiority of CNI-VIF in ensuring robust and reliable botnet detection across diverse machine learning models.



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
VIF	0.95	0.99	0.97	0.98	0.97	0.98
CNI-VIF	1	1	1	1	1	1
PCA	0.99	1	0.99	0.99	0.99	0.99
RFE	1	1	1	1	1	1

Fig. 10. Comparison of models' accuracies with various feature selection techniques (NCC-2)

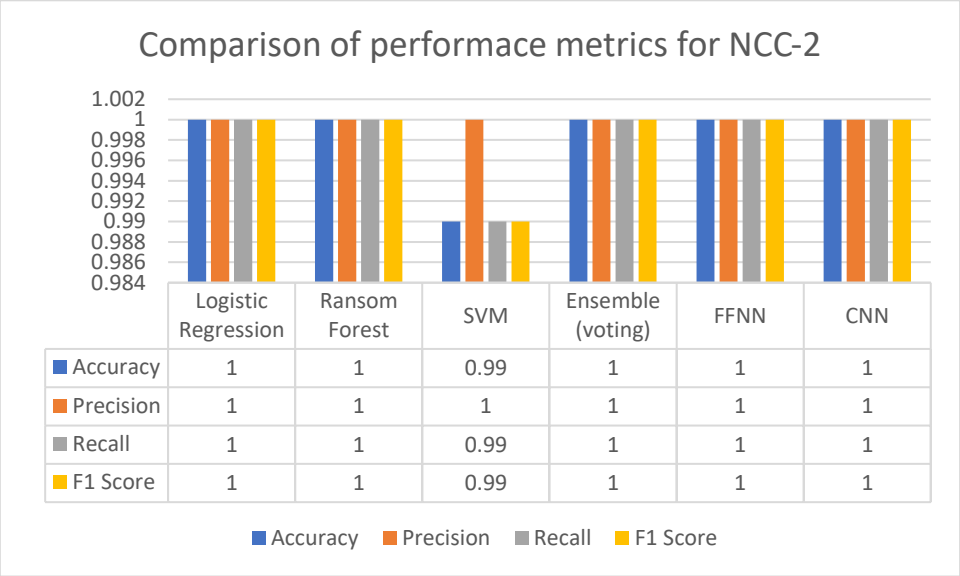


	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
VIF	0.0153	0.8616	5.1595	6.0044	2.6127	4.4791
CNI-VIF	0.0195	0.3708	1.1467	1.5325	2.4577	7.1445
PCA	0.0195	0.4705	1.7834	2.2939	2.4572	4.0733
RFE	0.0155	0.1952	0.3902	0.5929	2.4964	4.2707

Fig. 11. Comparison of models' computation time for botnet detection with various feature selection techniques (NCC-2)

Figure 11 illustrates the computation time for botnet detection across various machine learning models on the NCC-2 dataset, comparing different feature selection techniques: VIF, CNI-VIF, PCA, and RFE. Among the models,

Logistic Regression consistently demonstrates the lowest computation time across all methods, making it the most computationally efficient. CNI-VIF outperforms PCA and RFE in most models, achieving significantly lower computation times, particularly in computationally intensive models like SVM, Ensemble Voting, and CNN. PCA exhibits relatively higher computation times, especially for SVM and Ensemble Voting, indicating its higher resource requirement for feature extraction. RFE shows moderate computation times across models but exceeds CNI-VIF in complexity for some models like CNN. The results highlight that CNI-VIF provides an optimal balance of feature selection effectiveness and computational proficiency, making it a favorable choice for real-time botnet detection systems, especially in resource-constrained environments.



	Logistic Regression	Ransom Forest	SVM	Ensemble (voting)	FFNN	CNN
Accuracy	1	1	0.99	1	1	1
Precision	1	1	1	1	1	1
Recall	1	1	0.99	1	1	1
F1 Score	1	1	0.99	1	1	1

Fig. 12. Comparison of performance metrics using CNI-VIF for botnet detection

Figure 12 compares performance metrics for several machine learning models on the NCC-2 dataset. Logistic Regression, Random Forest, Ensemble, FFNN, and CNN models achieve perfect scores (1) across all metrics, indicating their effectiveness in identifying botnet traffic without errors. The SVM model demonstrates slightly lower performance, with a recall, accuracy, and F1 score of 0.99 while maintaining a precision of 1. This slight dip in recall indicates that SVM missed a small fraction of positive instances compared to other models. Overall, the results demonstrate that most models, particularly Logistic Regression, Random Forest, and Ensemble Voting, exhibit optimal performance on the NCC-2 dataset, underscoring the robustness of the proposed approach for botnet detection.

4) Comparison with state-of-art method:

All three datasets compare the proposed method with the state-of-the-art graph-based botnet detection technique [13]. The comparison concerns the following parameters: detection rate, Number of FPs, Number of FNs, performance metrics, and running time. All graph-based features listed in [13] are used, and then the Gini index measure for feature selection is applied. Researchers proved the decision tree to be the best model for their methodology, so the same is used when comparing it with the proposed model.

Figures 13, 15, and 17 compare the proposed methodology with the state-of-art method w.r.t detection rate, FP, and FN for CTU-13, IoT-23, and NCC-2 datasets. The proposed method achieves a perfect % detection rate of 100% for the CTU-13 and IoT-23 datasets and nearly perfect performance (99.99%) for the NCC-2 dataset, with no false positives and only one false negative in total across all datasets. In contrast, the state-of-the-art method achieves a slightly lower detection rate of 98.81% for CTU-13 and IoT-23 and 99.76% for NCC-2 while exhibiting 40 false negatives consistently across all datasets. These results highlight the superior performance of the proposed methodology in accurately detecting botnets without introducing errors, making it a more reliable and effective solution for network traffic analysis and botnet detection.

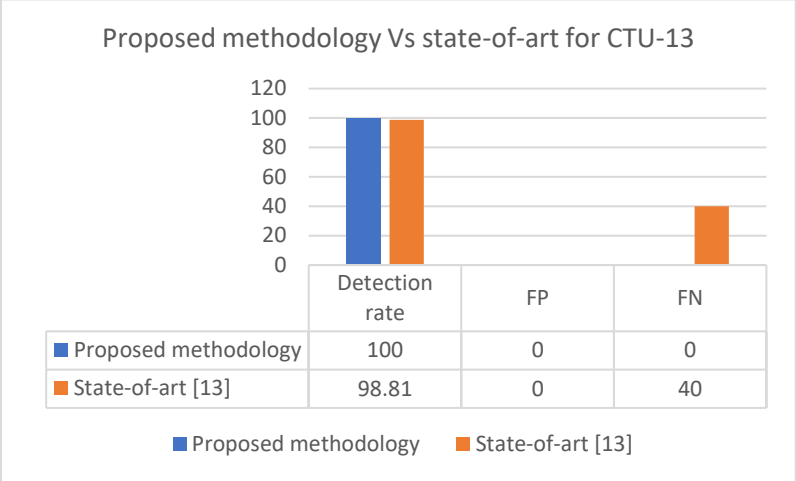


Fig.13. The proposed methodology Vs. state-of-art method w.r.t detection rate, FP, and FN

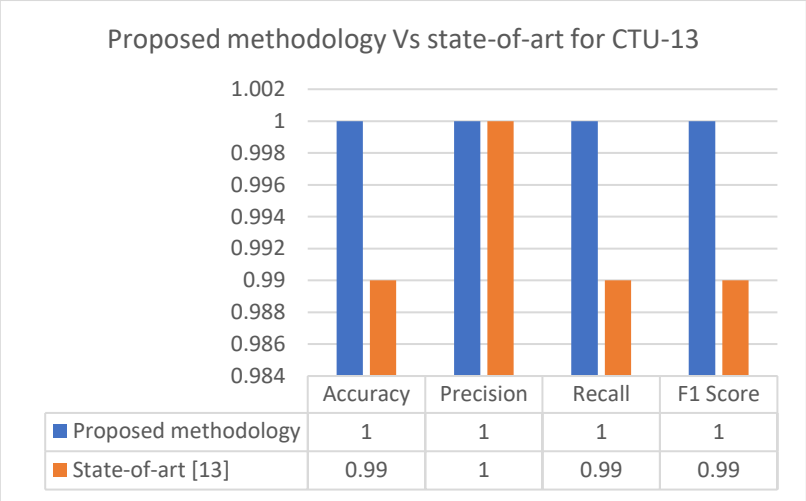


Fig. 14. The proposed methodology Vs. state-of-art method w.r.t performance metrics

Figures 14, 16, and 18 depict the performance of the proposed methodology with the state-of-the-art approach across the CTU-13, IoT-23, and NCC-2 datasets using standard evaluation metrics. The proposed method consistently achieves perfect scores of 1 across all metrics and datasets, indicating flawless detection and classification of botnet traffic. The state-of-the-art method also achieves near-perfect or perfect scores for most metrics, with a slight deviation in the recall and F1 score (0.99) for the CTU-13 dataset. These results underscore the robustness and reliability of the proposed methodology, particularly in its ability to outperform or match existing methods while ensuring comprehensive detection of botnet activity with minimal errors.

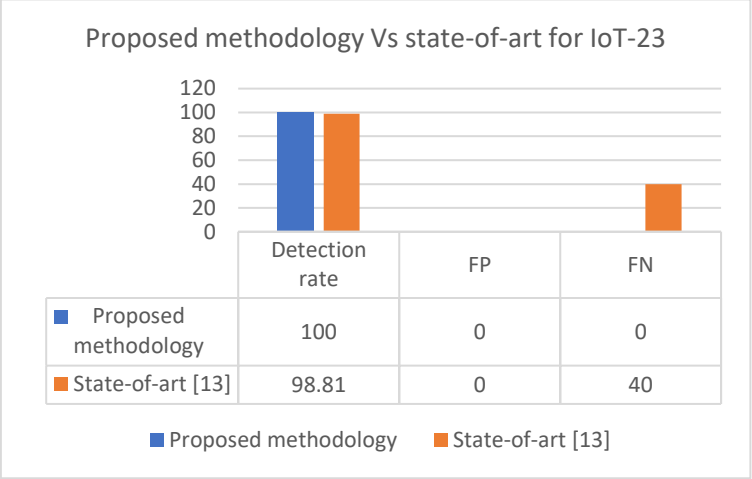


Fig. 15. The proposed methodology Vs. State-of-art method w.r.t detection rate, FP, and FN

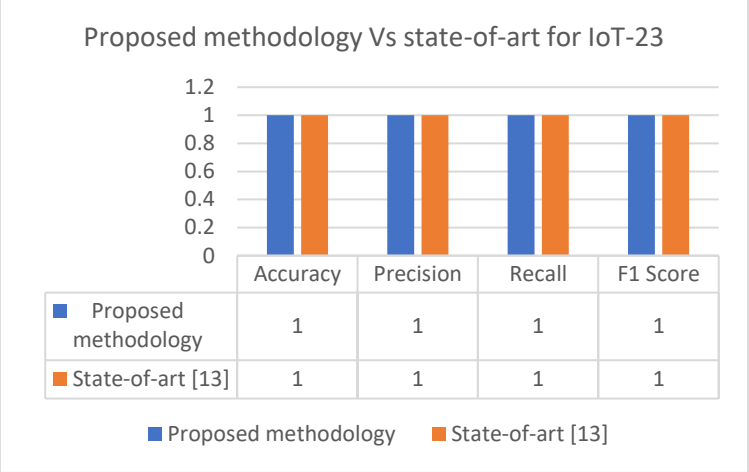


Fig. 16. The proposed methodology Vs. state-of-art method w.r.t performance metrics

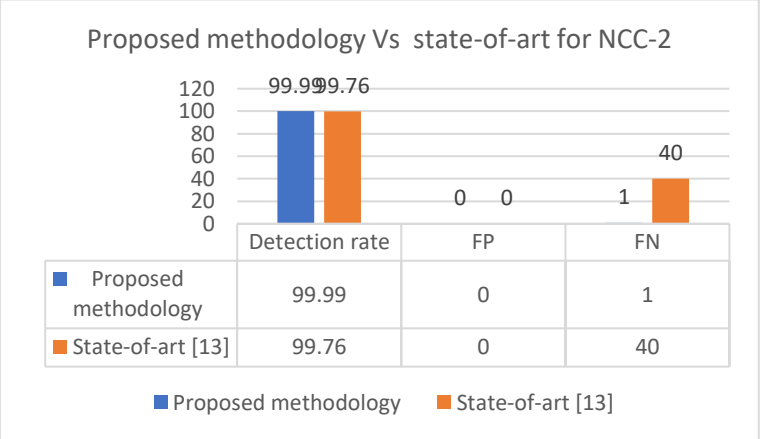


Fig. 17. The proposed methodology Vs. State-of-art method w.r.t detection rate, FP, and FN

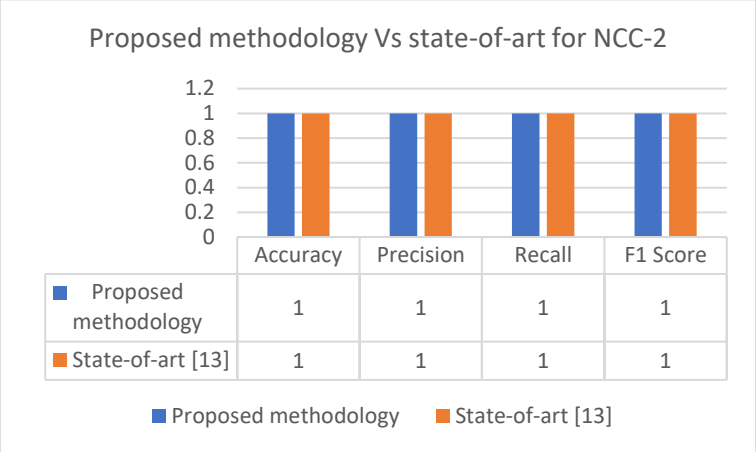


Fig. 18. The proposed methodology Vs. state-of-art method w.r.t performance metrics

Table 5 compares the proposed method's computation time and reduction rate (percentage) with the state-of-art method across three datasets: CTU-13, IoT-23, and NCC-2. As shown in Figure 19 and Table 5, the proposed method demonstrates a significant reduction in computation time compared to the state-of-the-art, with the reduction rates ranging from 76.60% for CTU-13 to 80.99% for IoT-23. This indicates that the proposed method is more efficient, requiring substantially less time for computation while maintaining or improving performance. These reductions are particularly beneficial for real-time or large-scale network traffic analysis, making the proposed methodology a more practical choice for botnet detection in complex datasets. By integrating graph-specific features with traditional feature selection methods, CNI-VIF identifies the most relevant features for botnet detection, improving accuracy and computational efficiency.

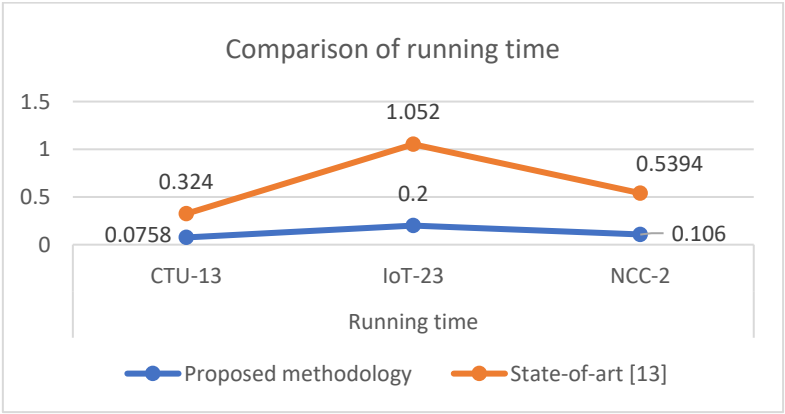


Fig. 19. The proposed methodology Vs. state-of-art method w.r.t running time

Table 5. Comparison of computation time

Dataset	Computation time (in seconds)		Reduction rate (percentage)
	State-of-art method [13]	Proposed method	
CTU-13	0.324	0.0758	76.60
IoT-23	1.052	0.20	80.99
NCC-2	0.5394	0.106	80.35

V. CONCLUSION

The rise of sophisticated social botnets, capable of mimicking legitimate behaviors, underscores the need for advanced detection systems tailored to dynamic and complex network environments. Traditional detection methods often struggle with the scale and intricacy of modern botnet activities, particularly when utilizing graph-based data. The proposed CNI-VIF framework for botnet detection addresses these challenges by combining graph-specific features with traditional statistical measures, providing a robust mechanism for feature selection and dimensionality reduction. This approach enhances the interpretability and efficiency of machine learning models, making it particularly well-suited for large-scale and real-time botnet detection scenarios. This method offers a practical and powerful solution to counter the evolving threat landscape by leveraging the structural insights provided by graph-based analysis.

The experimental evaluation of the proposed methodology on CTU-13, IoT-23, and NCC-2 datasets demonstrates its superior accuracy, detection rate, and computational efficiency. ML models like Random Forest and Convolutional Neural Networks, integrated with CNI-VIF-selected features, achieved detection rates exceeding 99.9% across diverse scenarios. The confidence interval results demonstrate that the proposed model achieves near-perfect or perfect performance across all datasets, with slight variability on NCC-2. This provides strong statistical evidence for the reliability and effectiveness of the proposed botnet detection methodology.

Compared to state-of-the-art methods, the proposed framework significantly reduces computational time by up to 80%, making it an efficient choice for resource-constrained environments. Additionally, the system minimizes false positives and negatives, ensuring reliable and actionable insights for network administrators. These results highlight the versatility and robustness of the framework, validating its efficacy across varying botnet attack patterns and network conditions. The proposed method achieves perfect or near-perfect performance across all datasets, outperforming the state-of-the-art method on IoT-23 and NCC-2.

Future research could extend this methodology to more heterogeneous network environments involving encrypted traffic and advanced evasion techniques. Enhancements to the CNI-VIF framework might include adaptive feature selection mechanisms that dynamically adjust to evolving botnet behaviors. Investigating the scalability of the cloud and distributed systems approach also holds promise for addressing broader cybersecurity challenges in IoT and industrial networks. By expanding the scope and adaptability of the framework, future work can ensure its relevance in combating increasingly complex cyber threats.

REFERENCES

[1] Editors, 2024. SonicWall Cyber Threat Report. Available at: https://www.sonicwall.com/resources/white-papers/mid-year-2024-sonicwall-cyber-threat-report. Last accessed: 5th December 2024.

[2] A. Nazir and R. A. Khan, “Network intrusion detection: Taxonomy and machine learning applications,” in Machine Intelligence and Big Data Analytics for Cybersecurity Applications (Studies in Computational Intelligence), vol. 919. Springer, 2021, pp. 3–28. DOI:

[3] S. Dange and M. Chatterjee, “IoT BotNet: The largest threat to the IoT network,” in Data Communication and Networks. Singapore: Springer, 2020, pp. 137–157. DOI:

[4] Aljabri, M., Zagrouba, R., Shaahid, A. et al. Machine learning-based social media bot detection: a comprehensive literature review. Soc. Netw. Anal. Min. 13, 20 (2023). https://doi.org/10.1007/s13278-022-01020-5.

[5] Venkatesh, G.Kirubavathi & Anitha, R.. (2014). Botnets: A Study and Analysis. Advances in Intelligent Systems and Computing. 246. 203-214. 10.1007/978-81-322-1680-3_23.

[6] Thanh Vu, S.N.; Stege, M.; El-Habr, P.I.; Bang, J.; Dragoni, N. A Survey on Botnets: Incentives, Evolution, Detection and Current Trends. Future Internet 2021, 13, 198. https://doi.org/10.3390/fi13080198.

[7] M. S. Koli and M. K. Chavan, "An advanced method for detection of botnet traffic using intrusion detection system," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 481-485, doi: 10.1109/ICICCT.2017.7975246.

[8] Javed Ashraf, Marwa Keshk, Nour Moustafa, Mohamed Abdel-Basset, Hasnat Khurshid, Asim D. Bakhshi, Reham R. Mostafa, IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities, Sustainable Cities and Society, Volume 72, 2021, 103041, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2021.103041.

[9] Patil, N.S. & Kiran, P. & Kavya, N.P. & Patel, K.M. (2018). A Survey on Graph Database Management Techniques for Huge Unstructured Data. International Journal of Electrical and Computer Engineering. DOI: 81. 1140-1149. 10.11591/ijece.v8i2.pp1140-1149.

[10] Ping Qiu, Chunxia Zhang, Dongping Gao, Zhendong Niu, “A fusion of centrality and correlation for feature selection”, Expert Systems with Applications, Volume 241, 2024,122548, ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122548.

[11] Pokorný, Jaroslav, “Graph Databases: Their Power and Limitations”, 9339, pp. 58-69, 2015.DOI:10.1007/978-3-319-24369-6_5., Ling Zheng, Fei Chao, Neil Mac Parthaláin, Defu Zhang, Qiang Shen, “Feature grouping and selection: A graph-based approach” , Information Sciences, Volume 546, 2021, Pages 1256-1272, ISSN 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.09.022.

[12] F. Cheng, C. Zhou, X. Liu, Q. Wang, J. Qiu and L. Zhang, "Graph-Based Feature Selection in Classification: Structure and Node Dynamic Mechanisms," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 4, pp. 1314-1328, Aug. 2023 doi: 10.1109/TETCI.2022.3225550.

[13] Afnan Alharbi and Khalid Alsubhi, “Botnet Detection Approach Using Graph-Based Machine Learning”, IEEE, July 19, 2021, Volume 9. DOI:

[14] Jiehong Cheng, Jun Sun, Kunshan Yao, Min Xu, Yan Cao, “A variable selection method based on mutual information and variance inflation factor”, Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, Volume 268, 2022, 120652, ISSN 1386-1425, DOI: https://doi.org/10.1016/j.saa.2021.120652.

[15] Anagha Patil, Arti Deshpande, "CNI-VIF: Enhanced Feature Selection for Graph Databases by Integrating Composite Node Information in VIF," SSRG International Journal of Electrical and Electronics Engineering, vol. 11, no. 11, pp. 100-113, 2024. Crossref, https://doi.org/10.14445/23488379/IJEEE-V11I11P111.

[16] Oliver Kornyo, Michael Asante, Richard Opoku, Kwabena Owusu-Agyemang, Benjamin Tei Partey, Emmanuel Kwesi Baah, Nkrumah Boadu, “Botnet attacks classification in AMI networks with recursive feature elimination (RFE) and machine learning algorithms”, Computers & Security, Volume 135, 2023, 103456, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2023.103456.

[17] Raihan Ur Rasool, Hafiz Farooq Ahmad, Wajid Rafique, Adnan Qayyum, Junaid Qadir, “Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML”, Journal of Network and Computer Applications, Volume 201, 2022, 103332, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2022.103332.

[18] Xuexiong Luo, JiaWu1, JianYang, Shan Xue, Hao Peng, Chuan Zhou, Hongyang Chen, Zhao Li & Quan Z. Sheng, “Deep graph level anomaly detection with contrastive learning”, Scientifc Reports, nature portfolio (2022) 12:19867 | https://doi.org/10.1038/s41598-022-22086-3.

[19] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, and Hui Xiong, and Leman Akoglu, "A Comprehensive Survey on Graph Anomaly Detection With Deep Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12012-12038, 1 Dec. 2023, doi: 10.1109/TKDE.2021.3118815.

[20] Tikekar, P.C., Sherekar, S.S. (2023). Comparative Analysis of Botnet Detection Techniques Using Machine Learning Classifier. In: Buyya, R., Misra, S., Leung, YW., Mondal, A. (eds) Proceedings of International Conference on Advanced Communications and Machine Intelligence. MICA 2022. Studies in Autonomic, Data-driven and Industrial Computing. Springer, Singapore. https://doi.org/10.1007/978-981-99-2768-5_19.

[21] Rimsha Malik, Bhavya Alankar, “Botnet and Botnet Detection Techniques”, International Journal of Computer Applications (0975 – 8887), Volume 178 – No. 17, June 2019. DOI:

[22] Foram Suthar, Nimisha Patel, Samarat V.O. Khanna, "A Signature-Based Botnet (Emotet) Detection Mechanism," International Journal of Engineering Trends and Technology, vol. 70, no. 5, pp. 185-193, 2022. Crossref, https://doi.org/10.14445/22315381/IJETT-V70I5P220.

[23] Szynkiewicz, P. (2022). Signature-Based Detection of Botnet DDoS Attacks. In: Kołodziej, J., Repetto, M., Duzha, A. (eds) Cybersecurity of Digital Service Chains. Lecture Notes in Computer Science, vol 13300. Springer, Cham. https://doi.org/10.1007/978-3-031-04036-8_6.

[24] Kaize Ding, Jundong Li, Huan Liu, “Interactive Anomaly Detection on Attributed Networks”, WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Pages 357 – 365, https://doi.org/10.1145/3289600.3290964.

[25] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," in *IEEE Access*, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[26] Daniel Spiekermann, Jörg Keller, “Unsupervised packet-based anomaly detection in virtual networks”, Computer Networks, Volume 192, 2021, 108017, ISSN 1389-1286, DOI: https://doi.org/10.1016/j.comnet.2021.108017.

[27] Jiaxin Liu, Xucheng Song, Yingjie Zhou, Xi Peng, Yanru Zhang, Pei Liu, Dapeng Wu, Ce Zhu, “Deep anomaly detection in packet payload”, Neurocomputing, Volume 485, 2022, Pages 205-218, ISSN 0925-2312, DOI: https://doi.org/10.1016/j.neucom.2021.01.146.

[28] Matthew V. Mahoney, 2003, “Network traffic anomaly detection based on packet bytes”, In Proceedings of the 2003 ACM symposium on Applied computing (SAC '03). Association for Computing Machinery, New York, NY, USA, 346–350. https://doi.org/10.1145/952532.952601.

[29] D. P. Hostiadi, T. Ahmad and W. Wibisono, "A New Approach of Botnet Activity Detection Model based on Time Periodic Analysis," 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2020, pp. 315-320, doi: 10.1109/CENIM51130.2020.9297846.

[30] Dandy Pramana Hostiadi, Waskitho Wibisono and Tohari Ahmad, “B-Corr Model for Bot Group Activity Detection Based on Network Flows Traffic Analysis”, KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 14, NO. 10, Oct. 2020 DOI:10.3837/tiis.2020.10.014.

[31] Hostiadi, D.P., Ahmad, T., Wibisono, W. (2021). A New Approach to Detecting Bot Attack Activity Scenario. In: Abraham, A., et al. Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020). SoCPaR 2020. Advances in Intelligent Systems and Computing, vol 1383. Springer, Cham. https://doi.org/10.1007/978-3-030-73689-7_78.

[32] Muhammad Aidiel Rachman Putra, Tohari Ahmad, Dandy Pramana Hostiadi, “Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks”, International Journal of Intelligent Engineering and Systems, Vol.15, No.4, 2022, 10.22266/ijies2022.0831.48.

[33] Cangshuai Wu, Jiangyong Shi, Yuexiang Yang, Wenhua Li, “Enhancing Machine Learning Based Malware Detection Model by Reinforcement Learning”, ICCNS '18: Proceedings of the 8th International Conference on Communication and Network Security, Pages 74 – 78, https://doi.org/10.1145/3290480.3290494.

[34] Mohammad Alauthman, Nauman Aslam, Mouhammd Al-kasassbeh, Suleman Khan, Ahmad Al-Qerem, Kim-Kwang Raymond Choo, An efficient reinforcement learning-based Botnet detection approach, Journal of Network and Computer Applications,Volume 150, 2020, 102479, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2019.102479.

[35] Alavizadeh, Hooman, Hootan Alavizadeh, and Julian Jang-Jaccard. 2022. "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection" Computers 11, no. 3: 41. https://doi.org/10.3390/computers11030041.

[36] Chowdhury, S., Khanzadeh, M., Akula, R. et al. Botnet detection using graph-based feature clustering. J Big Data 4, 14 (2017). https://doi.org/10.1186/s40537-017-0074-7.

[37] A. A. Daya, M. A. Salahuddin, N. Limam and R. Boutaba, "A Graph-Based Machine Learning Approach for Bot Detection," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019, pp. 144-152. Doi: https://ieeexplore.ieee.org/abstract/document/8717821.

[38] Jiawei Zhou, Zhiying Xu, Alexander M. Rush, Minlan Yu, “Automating Botnet Detection with Graph Neural Networks”, MLSys 2020 Conference, https://doi.org/10.48550/arXiv.2003.06344.

[39] Hyunsang Choi, Heejo Lee, and Hyogon Kim. 2009. BotGAD: detecting botnets by capturing group activities in network traffic. In Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE (COMSWARE '09). Association for Computing Machinery, New York, NY, USA, Article 2, 1–8. https://doi.org/10.1145/1621890.1621893.

[40] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In Proceedings of the 15th Annual Network and DistributedSystem Security Symposium (NDSS’08), February 2008.

[41] Manmeet Singh, Maninder Singh, Sanmeet Kaur, “Issues and challenges in DNS based botnet detection: A survey”, Computers & Security, Volume 86, 2019, Pages 28-52, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2019.05.019.

[42] Wai Weng Lo, Gayan Kulatilleke, Mohanad Sarhan, Siamak Layeghy, Marius Portmann, XG-BoT: An explainable deep graph neural network for botnet detection and forensics, Internet of Things, Volume 22, 2023, 100747, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2023.100747.

[43] T. de Paula Peixoto. Graph-Tool. Accessed Oct. 28, 2024. [Online]. Available:https://graph-tool.skewed.de/.

[44] S. García, M. Grill, J. Stiborek, A. Zunino, “An empirical comparison of botnet detection methods”, Comput. Secur. 45 (2014) 100-123, doi: 10.1016/j.cose.2014.05.011.
[45] A. Parmisano, S. Garcia, and M. Jose Erquiaga. (Jan. 22, 2020), Aposemat IoT-23, Stratosphere Laboratory, A Labeled Dataset With Malicious and Benign IoT Network Traffic. Accessed: Feb. 2, 2023. [Online]. Available: https://www.stratosphereips.org/datasets-iot23.
[46] Putra, M Aidiel Rachman; Ahmad, Tohari; Hostiadi, Dandy Pramana (2022), “NCC-2 Dataset: Simultaneous Botnet Dataset”, Mendeley Data, V2, doi: 10.17632/8dpt85jrhp.2.
[47] Dandy Pramana Hostiadi, Tohari Ahmad, “Dataset for Botnet group activity with adaptive generator”, Data in Brief, Volume 38, 2021, 107334, ISSN 2352-3409, doi: 10.1016/j.dib.2021.107334.