Received: November 23, 2024. Revised: December 10, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

GrMA-CNN: Integrating Spatial-Spectral Layers with Modified Attention for Botnet Detection Using Graph Convolution for Securing Networks

Mohan H G[1]* **Jalesh Kumar[1]** **Nandish M[1]**

[1]*Department of Computer Science and Engineering, JNNCE Shivamogga, Visvesvaraya Technological University, Belagavi – 590018, Karnataka, India*
*Corresponding author's Email: mohan@jnnce.ac.in

Abstract: Network botnet attacks have been increasing rapidly because of the widespread use of interconnected Internet of Things (IoT) devices. These devices can be used for many malicious actions, such as phishing, fraud, data theft, and distributed computing attacks against IoT networks. The traditional methods of botnet detection fail to capture the relationships between network nodes that exhibit coordinated behavior. In this paper, we introduce a novel Graph-based Modified Attention with Convolutional Neural Network (GrMA-CNN) for the effective detection of botnet attacks. The novelty of GrMA-CNN lies in its integration of spectral and spatial layers within a Graph Convolutional Network (GCN). It combines the GCN with a modified attention mechanism to effectively capture relationships and coordinated behaviours among IoT devices in graph-structured data. The approach extract features from network flow traffic using hybrid feature selection techniques, which include mutual information, correlation analysis, and principal component analysis. The extracted features are then processed through a GCN, with spectral and spatial layers that operates directly on graph-structured data. In this context, each IoT device and its associated features are represented as nodes, while the relationships between these devices are modelled as edges in the graph. The robustness of the model is verified on different datasets, such as N-BaIoT, BoT-IoT, CTU-13, and CICIDS. The proposed model obtained an accuracy of 99.1% on N-BaIoT, 99.2% on BoT-IoT, 99.15% on CTU-13, and 99.3% on CICIDS datasets. Further the model has achieved an average precision of 98.82%, a recall of 99.02%, and F1-score of 98.51%. The performance comparison demonstrates that the proposed model outperforms state-of-the-art botnet detection methods, including DNN, SGDC, WCC, and IHHO-NN with high detection rate.

Keywords: Cybersecurity, Botnet, Internet of things, Feature engineering, Graph convolution network.

1. Introduction

There has been tremendous growth in the Industrial sector led by the Industry 4.0 revolution, in the last decade. The adoption of the IoT has been vital to this industrial revolution. The IoT devices are used in a diverse array of applications, such as warehouse management, logistics management, energy systems, healthcare, agriculture, smart cities, traffic management [1-2] etc. According to the study presented in [3], the authors predicted that by 2050, over 100 billion devices will be online. Consequently, attacks on these IoT networks are also increasing. Hence, there is a need to identify malicious botnet attacks [4]. Several categories of attacks have been identified, such as ransomware attacks, physical attacks, data breaches, and exploiting vulnerabilities.

A 'Botnet' refers to a group of compromised systems under the control of a hacker, termed a botmaster, operating remotely. Combining the terms 'robot' and 'network' illustrates how the botnet serves as a collective entity at the botmaster's command. Essentially, the botnet is tasked with executing various attacks according to the directives provided by its botmaster [5]. In general, compromised IoT devices operate inconspicuously, revealing no external signs of being hacked, and essentially function as zombies at the command of the botmaster, facilitating the execution of attacks [6]. Until the command is received from the botmaster, bots hide themselves by remaining inactive and not

performing any attacks [7]. This behavior of bots increases the difficulty of identifying infected systems.

Although there has been significant development in the utilization of deep learning (DL) and machine learning (ML) algorithms [8] to identify botnet through network traffic analysis, a gap still exists in the complete understanding of the relationships among botnet entities. These relationships provide significant insight into how bot activities can be differentiated from normal patterns of communication. Graph convolutional networks (GCNs) have shown potential in capturing structural dependencies in networks [9], yet they remain largely unexplored for detecting botnet traffic. The GCN processes a node based on its neighbourhood, capturing the inherent relationships exhibited by the nodes in the network graph. Botnet has a coherent relationship among the participating bots, making it suitable for applying a GCN-based method in detecting them.

1.1 Problem statement

The rapid proliferation of interconnected IoT devices has led to an alarming increase in botnet attacks, enabling malicious activities such as phishing, data theft, and distributed computing assaults. Traditional botnet detection methods struggle to effectively capture the complex relationships between IoT devices exhibiting coordinated behaviour. There is a need for advanced detection techniques that incorporates both structural and feature-based relationships in network traffic to identify and mitigate botnet threats with higher accuracy and reliability. The primary contributions of the research are as follows:

- A hybrid feature extraction model is employed, which combines mutual information, correlation analysis and principal component analysis (PCA) to derive the most relevant features.
- A graph convolutional method integrated with a modified attention mechanism, the GrMA-CNN, is introduced to enhance the learning of spatial and spectral patterns for effective botnet detection.
- The performance of the proposed botnet detection system is evaluated on standard benchmark datasets, and the performance is compared against other state-of-the-art methods.

The rest of the article is organized as: Section 2 presents a review of the related works; the proposed GrMA-CNN model is described in Section 3; Section 4 presents the results obtained; and Section 5 offers the concluding remarks.

2. Literature survey

A brief discussion of existing ML and DL-based botnet detection approaches is presented in this section. The ensemble approaches using ML algorithms are presented in [4]. The methods, such as random forest and logistic regression [5] are used to detect botnets. However, these methods suffer from overfitting problems and fail to capture complex patterns and correlations between the features in network traffic. The sequential activity mining [6] is based on packet inspection. The method fails when packets are encapsulated and is resource intensive. The nearest neighbour method employed in [7] has a high false-negative rate and suboptimal detection efficiency.

In [10], a sequential architecture considers a hybrid feature selection model with time gap analysis using ML classifiers to process the obtained features. The ML approach in [11] is a two-fold method to identify and isolate botnet attacks. It includes a scanning mode to detect DDoS attacks, and the ResNet-18 model to make predictions. The traditional methods fail because of their poor generalization ability, which degrades their ability to learn complex patterns [12]. Hence, these models suffer from model drift. To overcome these problems, the authors in [13] used an autoencoder-based DL approach to detect bots using packet length sequence.

Srinivasan et al. [14] developed an ensemble classification-based approach; these protocols do not support the detection of these botnets in their early stages. Popoola et al. [15] used a recurrent neural model for botnet identification with imbalanced traffic data. To solve class imbalance difficulties, Xu et al. [16] modelled an autoencoder utilizing cosine loss. Furthermore, bi-directional long short-term memory (LSTM) for attack identification was used in [17] for IoT networks.

In [18], a stochastic gradient descent classifier with reduced dimensions was utilized for botnet detection, but the method was weak in detecting minority samples. The Federated Deep Learning method was used in [19], but it suffers from backdoor and poisoning attacks. The fuzzy method employed in [20] is complex with a high number of detection rules. The method results in lower detection rates and higher false positive rates.

The authors in [21] merged clustering with SVM for botnet identification. The method relies on previous communication data from senders and receivers to differentiate traffic. The decision tree combined with the SVM classifier [22] and C4.5 [23] are less efficient than the other models. The filter and wrapper methods are employed in [24] to extract the

features with unsupervised clustering. Wrapper methods [25] and self-training neural models [26] are used to select features from network traffic. It suffers from computational complexity, as the model is trained on multiple subsets of features multiple times.

The work presented in [27] uses a CNN combined with flow features to detect bots. However, CNN and LSTM were used to detect specific botnets in [8], with a focus on IoT camera systems. The method in [28] uses an RNN to detect bots from traffic flow data, but the method is less accurate, with higher false positive rates. These methods do not capture the long-range dependencies in network flow between the entities. The real-world botnet datasets are highly imbalanced, with very little bot traffic compared with normal traffic; hence, botnet detection models must consider approaches to handle imbalances in the data. The dung beetle optimizer with genetic algorithms [29] and the Adaboost classifier [30] with an improved grey wolf optimizer perform poorly when dealing with imbalanced datasets. To address this issue, Generative Adversarial Networks (GAN) were used to generate the synthetic samples in [31-32].

The current botnet detection methods that use network traffic flow methods rely on flow-based statistics to identify the botnet. The dynamic behaviour of bots can modify the flow characteristics, which bypasses traffic flow-oriented signature-based detection methods. These detection methods can be easily evaded by bots with encrypted traffic. The bots launch a coordinated attack from multiple compromised devices. The existing methods focus on individual devices and lack the visibility and scalability needed to identify distributed attacks.

Existing botnet detection methods face several limitations that hinder their effectiveness. Packet inspection-based methods are resource-intensive and ineffective against encapsulated packets, while techniques like nearest neighbours and clustering exhibit high false-negative rates and suboptimal detection efficiency. Deep learning models, including CNNs, LSTMs, and autoencoders, often struggle with handling class imbalances, learning long-range dependencies, and generalizing to diverse botnet patterns. Flow-based and signature-based detection approaches are easily bypassed by bots using encrypted traffic, and they lack the scalability needed to identify distributed attacks across multiple devices. Moreover, these methods generally overlook the inherent relationships within the nodes of the network, which are crucial for detecting coordinated bot actions. Recent research has leveraged advanced methods like Transformers [37-39] and Graph Attention Networks (GAT) [40] for IoT security by analysing network traffic. Transformers excel at capturing temporal dependencies, while GATs effectively model communication structures. However, both Transformers and GAT suffer from high computational complexity on large datasets and are prone to overfitting. In contrast, GCN overcomes these limitations with localized aggregation, offering computational efficiency and robustness to noise.

To address these challenges, the proposed approach employs a graph-based representation that captures the inherent relationships among devices in the network. A network graph is constructed with nodes representing devices and edges signifying communication links, allowing for the identification of coordinated bot actions through relational analysis. A hybrid feature extraction model combining mutual information, correlation analysis, and PCA ensures the selection of the most relevant features, enhancing detection accuracy. The GrMA-CNN model further improves botnet detection by integrating a graph convolutional method with a modified attention mechanism, enabling the detection of spatial and spectral patterns while mitigating class imbalance issues. This method provides scalability and visibility to detect distributed attacks, even in the presence of encrypted traffic, and cannot be bypassed by modifying packet statistics, thus making it a robust solution to the shortcomings of existing techniques.

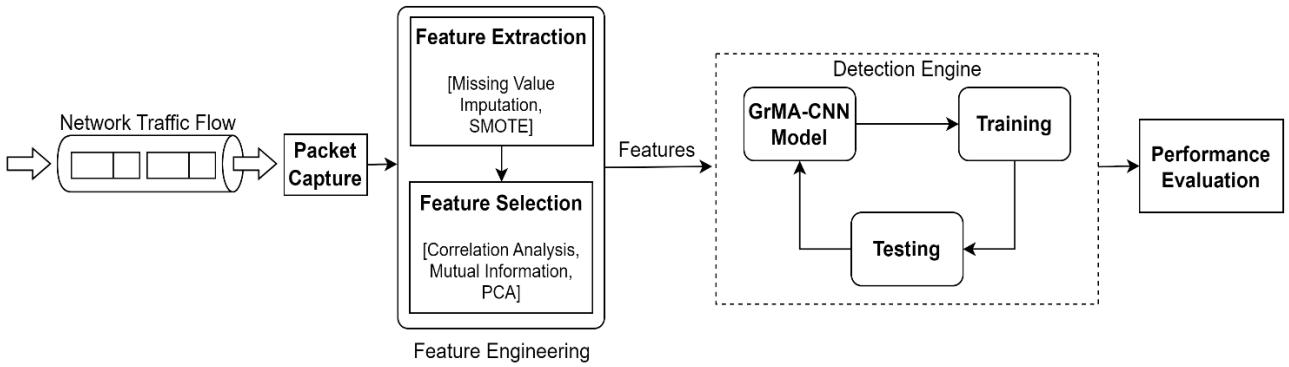

Figure. 1 Overview of the proposed botnet detection model

Table 1. Notations List

Symbol	Meaning
$G(V, E)$	Graph with vertex (V) and edges (E)
A	Adjacency matrix
D	Degree matrix
L	Laplacian matrix
U	Eigenvectors
Θ	Filter in spectral domain
X	Features of node
Y	Target value
k	Number of features
F	Set of features
$I(x_i, y)$	Mutual information of feature x_i and y
CM	Covariance matrix
$\rho_{X,Y}$	Correlation between X and Y
F_{Mutual}	Features selected by Mutual Information
F_{PCA}	Features selected by PCA
F_{Corr}	Features selected by Correlation Analysis
$\mathcal{N}$	Neighborhood of a node
Q	Queries of attention mechanism
K	Keys of attention mechanism
V	Values of attention mechanism
S_{head}	Attention head size
A_h	Prediction from attention head
$ReLU$	Rectified Linear Unit
$Softmax$	Softmax activation function
Acc	Accuracy
Pr	Precision
Re	Recall
$F1$	F1-score

3. The proposed method

This section presents the proposed GrMA-CNN, a deep learning approach for botnet detection on IoT networks. Fig. 1 shows an overview of the botnet detection system. It involves feature selection using hybrid approach and GrMA-CNN model is applied to detect the botnets. Table 1 presents the various notations used in representing the proposed model.

3.1 Preprocessing and feature extraction

This phase considers the data loading, missing value handling, and label encoder tasks. The issue of missing values is overcome by applying the k-nearest neighbour imputation mechanism. However, network traffic data exhibit the problem of data class imbalance; therefore, the SMOTE approach is used to produce synthetic instances. The new instances are generated by expanding the minority class, increasing the size of the dataset, and not resorting to the repetitive nature of conventional oversampling techniques. This methodology entails crafting synthetic instances belonging to the minority class within the feature space of existing minority examples.

The feature space is represented as X and $X \in R^{m \times n}$, where R represents the set of real numbers, the number of features is n and m is the instance count. Similarly, the target variables are denoted by $y, y_i \in \{0,1\}$. The dominant class in these samples is denoted by 0, and the minority class is denoted by 1. The distribution of y is changed to have y' in a uniform distribution to balance the entire dataset. SMOTE helps in estimating the k-nearest minority class, and one neighbor value is chosen to generate the synthetic instance; the new instance can be expressed as:

$$x_{new} = x_i + \beta * (x_{nn} - x_i) \quad (1)$$

Here, the random value β ranges from 0 to 1. This process of generating synthetic instances is repeated until the appropriate class balance is obtained in the dataset.

The dataset is processed using missing value imputation and the synthetic minority oversampling technique to avoid class imbalance issues. The complete balanced dataset is processed to extract the features. In this stage, data normalization is performed by applying a standard scalar mechanism. It rescales the values to have a mean of 0 and a standard deviation of 1.

To obtain the most relevant features, a hybrid approach using PCA, correlation analysis and mutual information is performed. The hybrid feature extraction process is presented in Algorithm 1. Each method ranks the features based on their ability to distinguish the samples as normal or botnet. The top *k*-ranked features obtained by the three methods are selected with a majority vote to generate the final feature vector.

Algorithm 1: Pseudocode for feature extraction

Input: *Node features for each node: X*
Target values: Y
No. of features: k

Output: *Set of features: F*

Stage 1: Apply Mutual Information Analysis

1. *Calculate mutual information values:* $I(x_i, y) = \sum_{x_i \in X} \sum_{y_i \in Y} P(x_i, y_i)\, log\left(\frac{P(x_i, y_i)}{P(x_i)P(y_i)}\right), P(x_i,$
2. *Assign* F_{Mutual} *with top k features from step 1*

Stage 2: Apply Principal Component Analysis

3. *Standardize the features:*

$$F_S = \frac{F_v - \mu}{\sigma}$$

4. *Compute the covariance matrix and extract eigenvalues and eigenvectors from* F_S*:*
5. *Assign* F_{PCA} *with top k features corresponding to k largest eigenvalues*

Stage 3: Apply Correlation Analysis

6. *Find correlation between each pair of feature variable and target variable over matrix* X *as:*

$$\rho_{X,Y} = \frac{\sum_{i \in \mathcal{N}} (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i \in \mathcal{N}} (X_i - \mu_X)^2 \sum_{i \in \mathcal{N}} (Y_i - \mu_Y)^2}}$$

Where, $\rho_{X,Y}$ *is correlation between X and Y*

X_i and Y_i *are individual data points*

μ_X and μ_Y *are mean values*

$\mathcal{N}$ *is set of sample instances*

7. *Assign* F_{Corr} *with top k features with large correlation coefficients*

Stage 4: Select the Features

8. *Select top k features with majority vote from* $\{F_{Mutual}, F_{PCA}, F_{Corr}\}$ *as F*
9. ***return*** *F*

3.2 Detection engine

This section presents the proposed approach, GrMA-CNN, for predicting botnet attacks. In order to construct the graphs, the multigraph structure method is adopted, which is denoted as $G_m = (V, E, F_v, F_e)$; Where, V represents the set of nodes, E is the set of edges representing network traffic flow between nodes, F_v is a feature matrix for nodes and, F_e is feature vector for edges. Fig. 2 depicts the proposed, GrMA-CNN with spectral and spatial layers to detect botnets.

The input graph passes through multiple spectral and spatial layers repeatedly to learn the relationships between the nodes. The spectral layers perform convolution to capture smooth variation in graph data detecting global patterns and structures. The spatial layers define convolutions directly by aggregating the features from neighbouring nodes. The final layer of the GCN maps the aggregated and transformed node features to the desired output.

To increase the detection accuracy, the GrMA-CNN architecture combines spatial and spectral layers. Algorithm 2 depicts the spectral convolution layer that performs the following operations:

A) Graph Laplacian (L): Given an undirected graph $G = (V, E)$ with N nodes and adjacency matrix A, $L = D - A$, with degree matrix D. The Laplacian encapsulates the graph's structural properties and is essential for spectral graph analysis.

B) Spectral Transformation: The spectral convolution operates in the spectral domain by transforming the input signal X into the Fourier basis of the graph. This transformation is achieved using the eigenvectors U of the graph laplacian matrix L. The transformed signal is computed as $X = U^T * X$.

C) Filter Operation: After transforming the signal into the spectral domain, a convolution operation is applied using a learnable filter Θ. Here, Θ is learned to optimize the loss during training using Chebyshev polynomial approximation. The filtered signal is computed as $\hat{Y} = \Theta \hat{X}$.

D) Inverse Transformation: To return the filtered signal to the original domain, an inverse spectral transformation is performed. This involves projecting the filtered signal back using the eigenvectors U, resulting in the output: $Y = U \hat{Y}$.

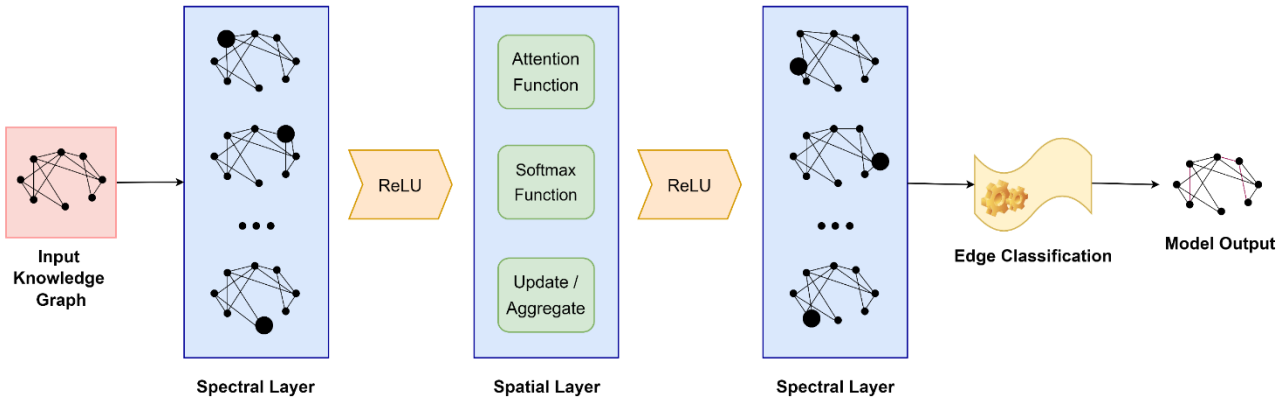

Figure. 2 GrMA-CNN Architecture

Algorithm 2: Pseudocode for spectral layer

Input: *Node features for each node: X*
Network Graph: G(V, E)
Degree Matrix: D
Adjacency Matrix: A

Output: *Node outputs: Y*

1. *Compute symmetric normalized Laplacian matrix:*
   $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$
   $U = eigen\ vectors\ of\ L$
   $\nabla = eigen\ values\ of\ L$
2. *Perform eigen decomposition:*
   $L = U \nabla U^T$
3. *Apply spectral transformation:*
   $\widehat{X} = U^T * X$
4. *Compute Chebyshev polynomial approximation:*
   $g_\theta(\nabla) \approx \sum_{i=0}^{K} \theta_i T_i(\nabla)$
   Where, K is polynomial order
   $T_i(\nabla)$ *is the Chebyshev polynomial of L*
   $g_\theta(\nabla)$ *is spectral filter with learnable parameter* θ
5. *Perform spectral filter operation:*
   $\hat{X}_{filtered} = g_\theta(\nabla)\, \hat{X}$
6. *Apply inverse transformation:*
   $Y = U\, \hat{X}_{filtered}$
7. *Apply a nonlinear activation function:*
   $Y = ReLU(Y)$
8. ***return*** *Y*

The spatial layer operations are shown in Algorithm 3. The steps involved in spatial layer are:

A) Weighted Aggregation and Normalization: In spatial graph convolution, convolutional filters are applied directly to the local neighborhood of each node in the graph. Let, $\mathcal{N}(i)$ represent the neighborhood of node i. The output feature Y of node i is computed using Eq. (2).

$$Y_i = \sigma\left(\sum_{j \in N(i)} \frac{1}{c_{ij}} X_j W\right) \quad (2)$$

Where, X_j is the input feature of node j, σ is an activation function, W is the weight matrix, and normalization factor c_{ij}.

B) Parameter Sharing: Spatial Graph Convolutional filters share weights across different node neighborhoods. This allows the model to learn from the entire graph structure and generalize better.

C) Aggregation Function: To aggregate the information from neighboring nodes, different aggregation functions such as simple averaging, weighted sum, or more complex attention mechanisms, can be used.

D) Activation Function: Softmax activation function is applied to have non-linearity in the model.

Algorithm 3: Pseudocode for spatial layer

Input: *Node features for each node: X*
Neighbourhood: $\mathcal{N}$
Network Graph: G(V, E)

Output: *Node outputs: Y*

1. ***for each*** *node i*
2. ***for each*** *node* $j \in \mathcal{N}(i)$
3. $Y_i = weighted\_aggregation(X_i, X_j)$
4. $Y_i = normalize(Y_{i_i})$
5. ***end***
6. $Y_i = activation(Y_i)$
7. *propagate updated* Y_i *with* $\mathcal{N}(i)$
8. ***end***
9. ***repeat*** *until Y values stabilize*
10. ***return*** *Y*

In order to capture diverse and complex patterns in botnet activity, an attention mechanism is modified by applying a linear transformation to the input data X, which generates a transformed matrix of X into queries (Q), keys (K), and values (V).

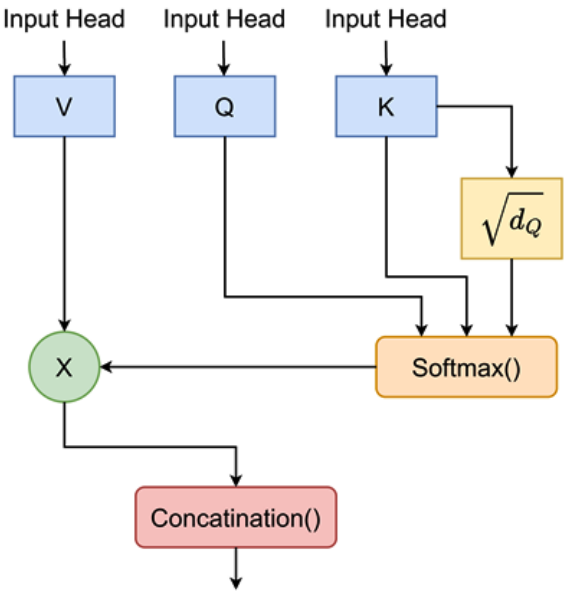

Figure. 3 Modified Attention Block

Table 2. Botnets used for generating real-world datasets

Dataset	Botnet Used
N-BaIoT [33]	BASHLITE and Mirai
BoT-IoT [34]	Tool Generated
CTU-13 [35]	Neris, NSIS.ay, Menti, Murlo, Sogou, Rbot and Virut
CICIDS [36]	ARES

The proposed attention mechanism is presented in Fig. 3. In a multi-head attention mechanism, each head independently learns different aspects of the input data by applying separate linear transformations and attention processes. The linear transformation for *Q*, *K*, and *V* can be expressed in Eq. (3):

$$Q = XW_Q + b_Q; K = XW_K + b_K; V = XW_v + b_v \quad (3)$$

Where, b is bias vector and W is weight matrix.

The layered GrMA-CNN architecture is mathematically denoted in Eq. (4).

$$g(F_v, F_e, A) = \hat{A}\ U\left(A\ F_e\ U\left(\hat{A}\ F_v\ W_1^g\right)W_2^{att}\right)W_3^g \quad (4)$$

Here, W_1^g W_2^{att} and W_3^g are hidden layer weights. $\hat{A}$ is the normalized Laplacian matrix of A. Furthermore, a non-linear activation, ReLU, is employed to learn complex patterns and relationships. It is shown in Eq. (5):

$$ReLU(H) = max(0, H) = \begin{cases} H_{ij}, if\ H_{ij} > 0 \\ 0, if\ H_{ij} \leq 0 \end{cases} \quad (5)$$

In the proposed approach, the first layer focuses on learning the graph structure with the help of the eigen decomposition of the graph Laplacian matrix. Furthermore, the mean aggregation of edge weights is used to update the graph nodes. In the next stage, a non-linear activation function is applied after the aggregation operation. The attention mechanism concurrently processes all heads, determining the attention of each head. The final prediction of an attention head, A_h, is derived using Eq. (6).

$$A_h = softmax\left(\frac{Q \times K^T}{\sqrt{d_q}}\right) \times V \quad (6)$$

4. Results

4.1 Experimental setup

The proposed GrMA-CNN model is developed using Python 3.9.7 along with the Pandas, Numpy, Sciki-Learn, Network, TensorFlow, and Keras libraries, which are installed on Windows 11 OS. The performance of the model is evaluated on a wide range of real-world botnets shown in Table 2. The top features extracted by the proposed hybrid feature extraction method is presented in Table 3. For training 70% of the samples are used, and the remaining 30% are reserved for testing.

CTU-13: It includes the logs of botnet traffic that were obtained in 2011 at the Czech University of Technology [35]. Real botnet traffic was seen in this dataset, intermingled with background and regular traffic. In CTU- 13 dataset, the traffic flow is labelled into three classes: normal, background, and botnet traffic. It has 20,643,076 traffic flows, of which 369,806 are normal flow, 19,829,404 are background flow, and 443,866 flows are botnet traffic.

Table 3. List of top features extracted and their rationale for botnet detection

Feature	Rationale
Node Degree	Botnet devices often establish numerous connections having high degrees.
Source-Destination Pair Frequency	Botnets often communicate frequently between a small set of devices. Hence exhibits high Source- Destination Pair Frequency.
Edge Weight	High-weight edges indicate frequent or long-lasting communication between nodes, indicating botnet-controlled devices.
Betweenness Centrality	Nodes with high betweenness centrality facilitate connection between other nodes acting as botnet controllers.
Flow Duration	Botnet traffic often exhibits abnormal session durations, either very short or very long.
Community Identification	Botnets often form isolated communities of devices that communicate more frequently within the group.
Node Feature Similarity	Bots in a botnet often exhibit similar behavioural patterns across devices.
Bytes Transferred	Abnormal data flows, such as large-scale exfiltration or DDoS attacks, indicate botnet activity. Monitoring byte transfers helps in detecting unusual traffic patterns.

Table 4. Performance metrics and their descriptions for evaluating the proposed model

Metric	Description	Formula	
Accuracy	It is the proportion of instances that are correctly identified by the model as either bot or normal.	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$	(7)
Precision	It is the fraction of samples correctly identified as bot to the overall samples classified as bot by the model.	$Pr = \frac{TP}{TP + FP}$	(8)
Recall	It is the fraction of the bot instances in the dataset that are correctly identified as bot by the model.	$Re = \frac{TP}{TP + FN}$	(9)
F1-score	It is the harmonic mean of recall and precision. It is useful when the class distribution is imbalanced.	$F1 = \frac{2 * P * R}{P + R}$	(10)

Table 5. Hyperparameters used in Implementation and Simulation of the proposed approach

Parameter	Value
Activation Function	ReLU
Attention Dropout	0.1
Attention Mechanism	Multi-head
Batch Size	64
Correlation Method	Pearson
Dropout	0.5
Epochs	50
Hidden Units	[64, 32]
Learning Rate	0.001
Loss Method	Cross-entropy
Message Aggregation	Mean
Number of Neighbors	5
Number of heads	8
Number of Layers	8
Number of Neurons	64
Optimizer	Adam
Testing Samples	30%
Training Samples	70%
Weight Decay	0.0001

Table 6. Performance of the proposed GrMA-CNN

Metric	N-BaIoT	BoT-IoT	CTU-13	CIC-IDS
Acc	99.1	99.2	99.15	99.3
Pr	98.5	98.8	99.3	98.5
Re	99.2	99.25	99.85	97.8
F1	98.6	98.75	98.5	98.2

N-BaIoT: This dataset [33] was generated by considering the attacks of Mirai and Bashlite botnets in IoT devices. It has 116 distinct features providing significant information. The N-BaIoT dataset has 7,062,606 instances and is labelled into two classes, i.e., normal and botnet instances. It has 555,932 normal instances and 6,506,674 bot instances.

Bot-IoT: The Cyber Range Lab created this dataset [34] to simulate an authentic network environment for IoT. There are 46 features which includes parameters unique to IoT devices and their framework. The BoT-IoT dataset has 72,000,000 records belonging to two classes. Here, 9,543 records are normal flows and the remaining are botnet attack flows that performs various malicious activities.

CICIDS: The Canadian Institute generated the CICIDS dataset [36] for cybersecurity, which consists of a large volume of network traffic, with the most recent prevalent attacks representing real-world data. It contains both malicious and benign records of network traffic. The CICIDS dataset has 529,918 normal records and 191,033 bot attack records.

4.2 Metrics for performance measurement

The true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values are used to measure the performance of the model. The statistical performance metrics like F1-score, accuracy, recall, and precision, of the proposed GrMA-CNN approach are measured using the Eq. (7)-(10) mentioned in Table 4.

4.3 Performance analysis

The implementation and simulation parameters used in this research work are studied using grid search. The parameter values that provide optimal performance are presented in Table 5.

The proposed model demonstrated 99.10%, 99.20%, 99.15%, and 99.30% accuracy on the N-BaIoT, BoT-IoT, CTU-13, and CICIDS datasets, respectively. The performance of GrMA-CNN on different datasets is mentioned in Table 6.

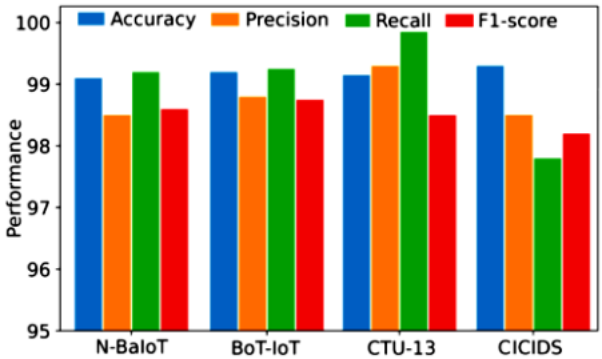

Figure. 4 Performance of the proposed GrMA-CNN on different datasets

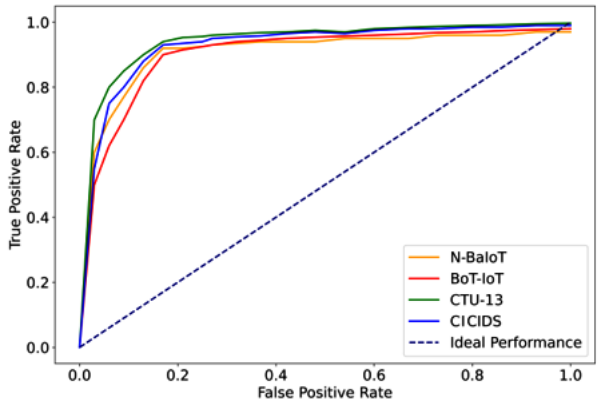

Figure. 5 The ROC curve obtained the proposed model on different datasets

Table 7. Performance of proposed approach with different resampling methods on CTU-13 for 100K samples

Method	Class	Acc	F1
GrMA-CNN + SMOTE	Normal	99.23	98.62
	Botnet	99.10	98.44
GrMA-CNN + GAN	Normal	98.66	97.16
	Botnet	97.22	95.86
GrMA-CNN + Cost-Sensitive	Normal	98.92	98.36
	Botnet	97.56	97.26
GrMA-CNN (No resampling)	Normal	96.25	95.16
	Botnet	93.65	93.15

The Fig. 4 demonstrates the model's effectiveness in stopping and identifying IoT botnet attacks. The model achieves an average precision rate of 98.8%, a recall rate of 99%, and an F1-score of 98.5% on all the datasets.

The ROC curve obtained at multiple thresholds that represent the range of TPR and FPR values is displayed in Fig. 5. The area under the curve (AUC) of the ROC curve indicates the effectiveness of the model. The proposed model has obtained an AUC of 0.96 on N-BaIoT, 0.94 on BoT-IoT, 0.97 on CTU-13, and 0.95 on CICIDS datasets. The suggested model has strong discriminatory strength in separating bots from normal traffic, as indicated by the ROC curve.

Table 7 compares the performance of GrMA-CNN with various resampling techniques on the imbalanced CTU-13 dataset. Among the methods, SMOTE achieves the best overall results, significantly enhancing Accuracy and F1-score for both majority normal and minority botnet classes, followed closely by GAN. In contrast, the model without balancing struggles with class imbalance, demonstrating the necessity of resampling techniques for robust botnet detection.

Table 8 presents the performance metrics of the proposed method on the CTU-13 dataset across varying numbers of nodes, highlighting its scalability and efficiency. The performance of the model remains consistently high, with a slight decline as the number of nodes increases. Training time and memory usage grow proportionally with dataset size, reflecting the computational demands of larger graphs. Inference time per node remains manageable, demonstrating the method's suitability for real-world applications with substantial data volumes.

4.4 Comparative analysis

The Table 9 highlights the superiority of the proposed GrMA-CNN compared to advanced models on the CTU-13 dataset. GrMA-CNN achieves the highest accuracy (99.15%) and F1 score (98.52%) while maintaining efficient training time, lower memory consumption, and linear scalability up to 1M nodes. In contrast, Transformer and GAT models exhibit significantly lower performance, higher resource demands, and limited scalability, particularly struggling with larger datasets.

Table 8. Performance of the proposed method on different numbers of nodes on CTU-13 dataset

Nodes	Acc	Pr	Re	F1	Training Time	Inference Time (per node)	Memory
1K	99.28	99.18	99.18	98.84	2 min	5 s	0.21 GB
10K	99.25	99.15	99.14	98.83	9 min	7 s	0.35 GB
100K	99.22	99.10	99.09	98.75	39 min	11 s	0.73 GB
1M	99.18	99.04	99.05	98.72	92 min	19 s	1.19 GB

Table 9. Performance comparison of proposed method with advanced models on CTU-13 dataset for 100K nodes

Method	Acc	F1	Training Time	Inference Time (per node)	Memory Consumed	Scalability (Devices)
GrMA-CNN	99.15	98.52	39 min	11 s	0.73 GB	Linear scaling up to 1M
Transformer	89.32	89.32	56 min	22 s	1.34 GB	Sub-linear scaling, struggles > 500K
GAT	85.62	85.62	72 min	35 s	1.03 GB	Poor scaling > 100K

Table 10. Performance comparison of GrMA-CNN with existing state-of-the-art works

Dataset	Work	Year	Accuracy	Precision	Recall	F1-score
N-BaIoT	SGDC [18]	2024	98.42	98.43	98.42	98.41
	DNN [6]	2024	97.21	91.41	87.31	88.48
	IHHO-NN [24]	2023	98.07	97.04	98.73	97.87
	WCC [21]	2022	96.70	94.90	94.70	94.80
	GrMA-CNN	**-**	**99.10**	**98.50**	**99.20**	**98.60**
CTU-13	[10]	2024	96.73	92	91.03	84
	[26]	2024	91.73	89.73	94.69	92.14
	[22]	2023	92.21	92.21	92.21	92.21
	[4]	2021	97.00	98.10	99.60	98
	GrMA-CNN	**-**	**99.15**	**99.30**	**99.85**	**98.50**
CICIDS	UNR-IDD [5]	2023	99.00	96.00	91.00	93.00
	[7]	2022	98.58	96.67	97.15	96.21
	GrMA-CNN	**-**	**99.30**	**98.50**	**97.80**	**98.20**
BoT-IoT	SGDC [18]	2024	92.28	92.28	92.48	92.37
	Genetic [25]	2023	97	97	97	97
	DBO-Catboost [29]	2023	98.57	98.62	98.57	98.57
	Fuzzy [20]	2022	96.41	98.80	98.80	98.80
	GrMA-CNN	**-**	**99.20**	**98.80**	**99.25**	**98.75**

The performance comparison across N-BaIoT, CTU-13, CICIDS, and BoT-IoT datasets presented in Table 10 highlights the superiority of the GrMA-CNN model. GrMA-CNN achieves the highest accuracy consistently surpassing other approaches. It also demonstrates strong precision, recall, and F1-scores, reflecting its balanced and robust performance. Competing models like SGDC, DBO-CatBoost, and IHHO-NN exhibit good results but fall short of GrMA-CNN's adaptability and effectiveness, particularly in datasets with complex patterns like BoT-IoT. For instance, on CTU-13, GrMA-CNN outperforms [4] (97.00% accuracy), and on CICIDS, it exceeds UNR-IDD ([5], 99.00%). Notably, the Fuzzy method ([20]) shows high precision and recall for BoT-IoT but lower accuracy (96.41%). GrMA-CNN's ability to generalize across diverse datasets establishes it as a state-of-the-art model for botnet detection, combining efficiency and scalability for real-world applications.

4.5 Discussions

The results obtained by the proposed GrMA-CNN model are discussed here to present insights into the findings. In our research, the dimensionality challenge is mitigated by employing a hybrid feature selection method that selects only the most relevant features. SMOTE is then applied to this reduced feature set, ensuring that synthetic samples are generated in a compact feature space, thereby eliminating the impact of high dimensionality. The proposed model does not read the contents of network packets and is thus immune to encrypted botnet traffic, overcoming the drawbacks of existing methods. To verify the performance on a wide range of large-scale real-world botnets, the proposed method is evaluated on four different real-world network traffic flow datasets. The hyperparameters are tuned to provide optimal performance using a grid search. The GrMA-CNN model has obtained a mean accuracy of 99.2%, precision of 99.15%, recall of 98.95%, and F1-score of 98.55% across all the datasets. This suggests that the model performs well, with lower false positive and false negative rates. The ROC curve indicates that the model converges during the training stage and has strong discriminatory power in detecting botnets. The class-wise performance obtained by the GrMA-CNN model shows the model's ability to detect minority classes. Futher, the experimental results shows that the proposed model is stable and scalable for large datasets.

5. Conclusions and future work

The attack detection in IoT environments is critical to many real-time applications, since the increased demand for IoT devices leads to an increase in vulnerability to different attacks. Many models have been suggested to enhance the performance of bot attack detection. However, traditional methods face several challenges due to imbalanced training data. In this work, a hybrid feature extraction is employed using PCA, correlation analysis and mutual information methods to extract robust features. Finally, the GCN based approach with modified attention block is implemented to enhance the ability of the learning process with spectral and spatial layers to capture complex information. The

performance of the GrMA-CNN is validated on benchmark datasets such as N-BaIoT, BoT-IoT, CTU-13, and CICIDS, which yield over a 99% detection rate with lower false positive rates. The proposed approach demonstrated better performance than other detection methods such as DNN, SGDC, WCC, and IHHO-NN. Future work on the proposed botnet detection model could focus on several areas to expand and enhance the security solution for IoT networks. The GrMA-CNN model can also be applied to detect other types of network attacks, such as intrusion detection, distributed denial-of-service (DDoS) attacks, and phishing attacks. Additional features beyond traffic flow data, such as device behaviour metrics, binary code analysis, and domain name server requests, to derive a holistic understanding of botnet activities.

Conflicts of Interest

All authors declare no conflict of interest

Author Contributions

Conceptualization, methodology, MHG; writing-original draft preparation, MHG, NM and JK; supervision, JK.

References

[1] Bharathi Malakreddy A et al., "FLQL-VANET: a Hybrid of Fuzzy Logic and Q-learning Schemes for QoS Aware Routing in VANET", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 6, pp. 1268-1280, 2024, doi: 10.22266/ijies2024.1231.92.

[2] D.H. Mustafa and I. M. Husien, "Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, 2023, doi: 10.22266/ijies2023.0831.33.

[3] Haroon et al., "Constraints in the IoT: The world in 2020 and beyond", *International Journal of Advanced Computer Science and Applications*, Vol. 7, pp. 252-271, 2016.

[4] C. Joshi et al., "Botnet Detection Using Machine Learning Algorithms", In: *Proc. of the International Conference on Paradigms of Computing*, Communication and Data Sciences, pp. 717-727, 2021.

[5] T. Das et al., "UNR-IDD: Intrusion Detection Dataset using Network Port Statistics", In: *Proc. of 2023 IEEE 20th Consumer Communications & Networking Conference*, pp. 497-500, 2023.

[6] Wardana et al., "Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of Things devices", *Sci Rep*, Vol. 14, No. 1, 2024, doi: 10.1038/s41598-024-54438-6.

[7] M. Andrecut, "Attack vs Benign Network Intrusion Traffic Classification", *arXiv:2205.07323*, 2022.

[8] M. Y. Alzahrani and A. M. Bamhdi, "Hybrid deep-learning model to detect botnet attacks over internet of things environments", *Soft computing*, Vol. 26, No. 16, 2022.

[9] X. Zhou et al., "Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System", *IEEE Internet Things J*, Vol. 9, No. 12, pp. 9310-9319, 2021.

[10] M.A.R. Putra et al., "Botnet sequential activity detection with hybrid analysis", *Egyptian Informatics Journal*, Vol. 25, 2024.

[11] F. Hussain et al., "A two-fold machine learning approach to prevent and detect IoT Botnet attacks", *IEEE Access*, Vol. 9, pp. 163412-163430, 2021.

[12] M. Al-Fawa'reh et al., "MalBoT-DRL: Malware Botnet Detection Using Deep Reinforcement Learning in IoT Networks", *IEEE Internet Things J*, Vol. 11, No. 6, 2024.

[13] C. Wei et al., "A lightweight deep learning framework for Botnet detecting at the IoT edge", *Computer Security*, Vol. 129, 2023.

[14] S. Srinivasan and P. Deepalakshmi, "Enhancing the security in cyber-world by detecting the Botnets using ensemble classification-based Machine Learning", *Measurement: Sensors*, Vol. 25, 2023.

[15] S. I. Popoola et al., "Stacked recurrent neural network for botnet detection in smart homes", *Computers & Electrical Engineering*, Vol. 92, 2021.

[16] X. Xu et al., "Toward Effective Intrusion Detection Using Log-Cosh Conditional Variational Autoencoder", *IEEE Internet Things J*, Vol. 8, No. 8, 2020.

[17] O. Alkadi et al., "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", *IEEE Internet Things J*, Vol. 8, No. 12, pp. 9463-9472, 2021.

[18] J. Azimjonov and T. Kim, "Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets", *Expert System Applications*, Vol. 237, 2024.

[19] S. Popoola et al., "Optimized Lightweight Federated Learning for Botnet Detection in Smart Critical Infrastructure", *TechRxiv*, 2023, doi: 10.36227/techrxiv.23620674.v1.

[20] M. Almseidin and M. Alkasassbeh, "An Accurate Detection Approach for IoT Botnet Attacks Using Interpolation Reasoning Method", *Information*, Vol. 13, No. 6, 2022.

[21] Y. Masoudi-Sobhanzadeh and S. Emami-Moghaddam, "A real-time IoT-based botnet detection method using a novel two-step feature selection technique and the support vector machine classifier", *Computer Networks*, Vol. 217, 2022.

[22] R. S. S. Moorthy and N. Nathiya, "Botnet Detection Using Artificial Intelligence", *Procedia Computer Science*, Vol. 218, 2023.

[23] R. T. Wiyono and N. D. W. Cahyani, "Performance Analysis of Decision Tree C4.5 as a Classification Technique to Conduct Network Forensics for Botnet Activities in Internet of Things", In: *Proc. of 2020 International Conference on Data Science and Its Applications*, IEEE, pp. 1-5, 2020.

[24] F. Taher et al., "Reliable Machine Learning Model for IIoT Botnet Detection", *IEEE Access*, Vol. 11, pp. 49319-49336, 2023.

[25] S. Ifikhar et al., "A Supervised Feature Selection Method for Malicious Intrusions Detection in IoT Based on Genetic Algorithm", *International Journal of Computer Science and Network Security*, Vol. 23, No. 3, pp. 49-56, 2023.

[26] T.-C. Lo et al., "Tackling Evolving Botnet Threats: A Gradual Self-Training Neural Network Approach", *IEEE Access*, 2024.

[27] Mohan HG et al., "A CNN based deep learning model for detecting P2P Botnets using flow features", In: *Proc. of 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), IEEE*, pp. 1-6, 2024, doi: 10.1109/NMITCON62075.2024.10698766.

[28] D. P. Hostiadi et al., "A New Approach to Detecting Bot Attack Activity Scenario", In: *Proc. of the 12th International Conference on Soft Computing and Pattern Recognition*, Springer, pp. 823-835, 2021.

[29] C. Yang et al., "IoT Botnet Attack Detection Model Based on DBO-Catboost", *Applied Sciences*, Vol. 13, No. 12, 2023.

[30] B. Padmavathi and B. Muthukumar, "An efficient botnet detection approach based on feature learning and classification", *Journal of Control and Decision*, Vol. 10, No. 1, 2023.

[31] O. Habibi et al., "Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection", *Eng Appl Artificial Intelligence*, Vol. 118, 2023.

[32] S. Gupta and B. Singh, "An intelligent multi-layer framework with SHAP integration for botnet detection and classification", *Computer Security*, Vol. 140, 2024.

[33] The N-BaIoT Dataset. Link: https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset

[34] The Bot-IoT Dataset. Link: https://research.unsw.edu.au/projects/bot-iot-dataset.

[35] The CTU-13 Dataset. Link: https://www.stratosphereips.org/datasets-ctu13.

[36] The CICIDS Dataset. Link: https://www.unb.ca/cic/datasets/ids-2017.html.

[37] L. D. Manocchio et. al, FlowTransformer: A transformer framework for flow-based network intrusion detection systems", *Expert System Applications*, Vol. 241, p. 122564, 2024, doi: 10.1016/j.eswa.2023.122564.

[38] Z. Long et. al, "A Transformer-based network intrusion detection approach for cloud security", *Journal of Cloud Computing*, Vol. 13, No. 1, 2024.

[39] Archana Kalidindi and Mahesh Babu Arrama, "A Tabtransformer Based Model for Detecting Botnet-Attacks on Internet of Things using Deep Learning", *Journal of Theoretical and Applied Information Technology*, Vol. 101, No. 13, 2023.

[40] P. M. V. S. Sai et. al, "Enhancing IoT Botnet Detection: An Ensemble Approach for Improved Network Security", In: *Proc. of 2023 International Conference on Next Generation Electronics (NEleX)*, IEEE, pp. 1-6, 2023, doi: 10.1109/NEleX59773.2023.10421421.