International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Design & analysis of Deep Learning based Defense Mechanisms against Cyber Attacks in IoT

Rahul R. Papalkar[*1], Dr. Abrar S. Alvi[2], Prof. Rajkumar Sawant[3], Prof. Harish Motekar[4], Prof. Amit Patl[5], Prof. Vinod Rathod[6]

Submitted: 25/01/2024 **Revised:** 03/03/2024 **Accepted:** 11/03/2024

Abstract: This research focuses on the crucial challenge of protecting Internet of Things (IoT) networks from various cyber threats, specifically zero-day attacks. Presenting a defense solution based on Convolutional Neural Networks (CNN), specifically designed to accommodate the unique operational constraints and resource limitations of the IoT. Creating a model that can effectively detect threats and is efficient for deployment on IoT devices with limited resources is the major objective. This project is based on a thorough examination of the BoT IoT dataset, which includes a wide range of simulated IoT network behaviors, covering regular operations and suspicious activities. This dataset forms the foundation for developing and evaluating our model. The CNN model we have developed features a simplified architecture that aims to minimize resource usage and enable real-time data analysis. By conducting thorough data preprocessing and rigorous training, the model is fine-tuned to accurately differentiate between harmless and harmful network patterns. After conducting the evaluation phase with the BoT IoT dataset, impressive results were obtained. The model achieved an accuracy rate of 95.98% and an F1 score of 0.8707. The metrics demonstrate the model's exceptional ability to accurately detect potential security threats in IoT networks. One interesting aspect of our approach involves incorporating a dynamic blacklisting mechanism that is tailored to improve the model's effectiveness in identifying zero-day attacks. Through regular updates to the blacklist using up-to-date threat intelligence and sending immediate alerts to administrators, the system enables quick responses and mitigation tactics against new threats, enhancing the security of IoT environments. This research validates the effectiveness of utilizing the proposed hybrid CNN model in IoT environments, especially with edge computing approaches, to enhance decentralized and effective threat detection in deployment scenarios. Moreover, the model includes continuous learning capabilities, enabling constant adaptation to emerging cyber threats. With a focus on ethical and security considerations, including protection against adversarial attacks and data privacy, the model's adaptability is strengthened, making it more reliable in IoT security frameworks. The research project concludes by introducing a CNN-based security mechanism that is both flexible and known for accurate detection; it also has the ability to detect zero-day threats through dynamic blacklisting. In order to safeguard IoT networks from the ever-evolving cyber threat landscape, this comprehensive method is a huge leap forward.

Keywords: *Machine Learning, Hybrid CNN Model, TEHO-DBN Classifier, Performance Metrics (Accuracy, Precision, Recall, F1 Score), Computational Efficiency, Convolutional Neural Network, Model Optimization*

1. Introduction

The rise of the Internet of Things (IoT) has revolutionized the digital realm, ushering in a fresh wave of connectivity and automation in different industries, ranging from smart homes to industrial setups. Nevertheless, the swift growth has increased the susceptibility of IoT networks to various cyber threats, with zero-day attacks presenting a notable challenge because of their unique and previously undisclosed characteristics. Our study presents a new approach to enhance security in IoT settings by utilizing Convolutional Neural Networks (CNN) to protect against common threats. This research effort is supported by a thorough examination of the BoT IoT dataset, a comprehensive collection of simulated IoT network traffic containing both normal and suspicious patterns. This dataset is a vital resource for the development and validation of our CNN model, guaranteeing its effectiveness in real-world IoT scenarios. We have developed a CNN model with a simplified architecture to meet the limitations of IoT devices, allowing for easy deployment and real-time threat detection. After conducting thorough data preprocessing and extensive model training, we have refined the CNN's capability to differentiate between normal and potentially harmful network activities with exceptional accuracy. After testing this model with the BoT IoT dataset, it showed remarkable performance with an accuracy rate of 95.98% and an F1 score of 0.8707. The metrics highlight the model's advanced ability to detect and address various security threats in IoT networks. One notable aspect of our approach is the incorporation of a dynamic blacklisting system, which greatly improves the model's ability to identify zero-day attacks. This system is regularly updated with the most recent threat intelligence, enabling the quick identification of new threats and the immediate issuance of alerts to network administrators. By taking a proactive stance, potential risks can be

1 1 & 2 Prof Ram Meghe Institute of Technology & Research , Badnera, MS India
ORCID ID : 0000-0002-2888-3525
2 3-6 Bharati Vidyapeeth Deemed Univeristy ,Department of Engineering & Technology Navi Mumbai MS India
Corresponding Author Email:rahulpapalkar@gmail.com

quickly addressed, enhancing the overall security of IoT environments. Our findings support the use of a hybrid CNN model, especially when combined with edge computing, for effective and decentralized threat detection in IoT networks. The model's continuous learning feature allows it to adjust to changing cyber threats, making it more effective in safeguarding IoT infrastructures. Emphasizing ethical and security aspects, such as strong defenses against adversarial attacks and a focus on data privacy, enhances the model's adaptability and reliability within IoT security frameworks.\

Research Gap:

There is a huge research gap in the creation of adaptive, resource-efficient defensive systems that are capable of fighting new threats, notably zero-day assaults, despite the numerous advancements that have been achieved in Internet of Things (IoT) infrastructure security. Traditional security methods frequently fail to meet the requirements of the ever-changing Internet of Things (IoT) landscape. This may be because these measures are unable to adapt to new threats or because they need a significant amount of resources, which is not consistent with the limited environment of IoT devices.

Challenges:

The following are some of the key challenges that this research aims to address:

- Resource Constraints: IoT devices frequently function with limited processing power and memory, which necessitates the implementation of security solutions that are both lightweight and effective without sacrificing effectiveness.
- Evolving Threat Landscape: The ever-changing nature of the threat landscape necessitates the existence of a defense mechanism that is capable of learning and adjusting in real time. This is especially true in the case of zero-day assaults, which are constantly emerging.
- Real-Time Detection: When it comes to Internet of Things networks, real-time detection refers to the requirement of quick danger identification and reaction in order to reduce possible harm.

Objectives:

This research is driven by the following objectives:

- To develop a CNN-based model tailored to the unique requirements of IoT networks, emphasizing efficiency and real-time processing capabilities.
- To validate the effectiveness of the proposed model using the comprehensive BoT IoT dataset, which simulates a wide spectrum of IoT network behaviors, from normal operations to malicious activities.
- To enhance the model's responsiveness to zero-day attacks through the integration of a dynamic blacklisting mechanism, thereby enabling the rapid identification and mitigation of emerging threats.

Rest of article is organized as follows. In chapter 2, The Literature Review explores IoT security, CNNs in cybersecurity, and current protection measures' weaknesses, revealing the research vacuum this study seeks to fill. Chapter 3 of our Methodology describes the meticulous process of developing the CNN model for IoT security, from its architectural foundations to its nuanced data preprocessing and training protocols to the innovative integration of a dynamic blacklisting mechanism to improve the model's responsiveness to zero-day threats. The Evaluation phase then analyzes the model's accuracy, efficiency, and real-time detection capabilities using a suite of evaluation criteria and rigorous testing using the BoT IoT dataset. We assess the model's practicality, dynamic blacklisting, and easy incorporation into IoT infrastructures in chapter 5 Discussion. In chapter 6, the Conclusion and Future Work section summarizes this research's main findings, emphasizing the CNN model's potential to improve IoT security standards and suggesting ways to improve it, ensuring its continued relevance and effectiveness in the ever-changing cybersecurity landscape.

2. Literature Survey:

An approach to feature selection based on assessment criteria obtained from weight values was proposed by [1]. Finding useful characteristics for DDoS attack detection using ANN models was the goal of their study. Even though the method worked well, it could be unreliable and obscure because it uses weight values for feature selection. To improve the detection of distributed denial of service attacks using artificial neural network models, [2] presented two feature selection algorithms based on wrappers. These techniques find the best subset by repeatedly testing the model's performance with various feature combinations. Wrapper approaches may not be scalable due to the computational expense they incur. Improving DDoS attack detection with artificial neural networks was suggested by[3] , who used wrapper-based feature selection approaches. In order to maximize the performance of the model, these techniques systematically remove superfluous features from the complete set of features. Although this method is successful, it could miss some feature interactions that are important for accurate identification. In their [4] refined DDoS attack detection models by using sequential backward selection (SBS) as a feature selection approach. To increase detection accuracy and decrease computing complexity, SBS systematically removes characteristics that do not contribute much to the model's performance. Nevertheless, the sequence of feature removal might affect SBS's efficacy. In order to improve the efficiency of models used to identify distributed denial of service (DDoS) attacks, [5] used sequential backward selection (SBS) as a feature selection method. With the goal of improving efficiency, SBS iteratively eliminates features from the dataset according to their contribution to model correctness. But SBS could miss feature interactions that are critical for correct identification. In their [6]. suggested a clamping method for determining which features are most important for detecting distributed denial of service attacks. This method attempts to determine the significance of features by setting the input value of each feature to its mean and then watching how it affects the performance of the model. Using mean values alone, though, could simplify feature dynamics to the exclusion of more complex patterns. In their 2014 study, Tang et al. improved DDoS attack detection by using a feature selection technique that uses evolutionary algorithms and mutual information. In this method, features are pre-selected based on mutual knowledge and the best feature subset is found via a random search technique. Genetic algorithms may not be scalable because of their computational complexity. In their [7] research used variance analysis to determine which variables were most useful for detecting distributed denial of service attacks. The goal of this approach is to enhance model accuracy by determining which features have the most

discriminating power by examining their variation. While feature interactions are a major factor in detection performance, variance analysis may fail to account for them. In their [8] study suggested a feature selection strategy for improving DDoS attack detection models that relies on multi-objective optimization. This technique seeks to discover the most important characteristics by maximizing numerous objectives at once, such as detection accuracy and processing efficiency. There may be an increase in processing cost due to the complexity of multi-objective optimization. In order to identify DDoS attacks, [9] used PCA and Fisher discriminant ratio (FDR) to choose the best features for probabilistic self-organizing map (SOM) models. This approach seeks to improve model performance by utilizing dimensionality reduction methods and discriminant analysis. The distribution of features and the parameters used may impact how successful PCA and FDR are.

The risk of data leaking has increased significantly due to unpatched software or application vulnerabilities. Zero Day vulnerabilities allow hackers to access the target network and take critical data. Traditional defenses present challenges in detecting zero-day threats due to unknown signature information. A new security solution is needed to detect and assess the severity of zero-day threats. The report suggests a method for identifying unknown vulnerabilities. Our framework identifies and prioritizes zero-day attacks, then removes them. The proposed methodology uses a probabilistic technique to identify Zero-Day attack paths and severity levels. It is a hybrid detection and eradication strategy that identifies undiscovered network faults. The system displays both the original and attacked file sizes[10]. [11] Introduced the Covariance matrix detection modeling approach to safeguard data integrity. The detection scheme identifies and prevents network attacks, safeguarding the network's integrity. The detection model is not appropriate for the practical implementation of CCS. In [12] introduced an autonomous multi-agent system utilizing an optimization algorithm to detect attacks. The scheme implements agents using an optimization approach and utilizes Particle Swarm Optimization (PSO). The model's optimization ensured ideal detection values, making it well-suited for the cloud platform. [13] Created an ensemble-based multi-filter feature selection model for attack detection. The model improved performance by decreasing the total number of features used for training. The system demonstrated a high detection rate, yet its performance was lower when tested on publicly available labeled datasets. [14] proposed the FSOMDM model for mitigating DDoS attacks using Fuzzy Self Organising Maps. The FSOMDM model identifies attacks by creating an attack-response model. The program successfully enhanced performance in preventing the dropping attack. In addition to enhancing attack detection performance, the control layer attacks remain unaddressed. In [15] introduced a detection scheme utilizing a statistical and distributed network packet filtering model. The scheme is updated efficiently, resulting in minimal storage needs. The system guaranteed successful detection of DDoS attacks in the cloud platform, although certain profiles in the algorithm are deemed irrelevant for profiling purposes. In [16] introduced a detection strategy utilizing the flow table sharing approach to enhance resistance against DDoS attacks. The method's efficacy in the attack is influenced by the duration of the holding time. The method's effectiveness hinges on the duration it is held. In [17] introduced an inter-domain interaction scheme to combat DDoS attacks in computer networks. The scheme demonstrated operational stability through the use of the Antidose technique for detecting attacks. Yet, the scheme's ability to withstand targeted attacks must be examined. In [18] study, the features of DDoS attacks in cloud computing were examined, along with a comprehensive overview of protection methods utilizing Software-Defined Networking (SDN). They analyzed studies on launching DDoS attacks on SDN and techniques to detect DDoS attacks in SDN. Based on the analysis, it was concluded that the various connections between SDN and DDoS attacks have not been adequately explored in the current literature. The authors discussed utilizing SDN to combat DDoS attacks in cloud computing environments and safeguarding SDN from being targeted by DDoS attacks. [19] discussed DDoS attack mitigation solutions in the cloud. The survey provided a detailed analysis of the characterization, prevention, detection, and mitigation strategies for these attacks. They provided a detailed solution taxonomy for categorizing DDoS attack solutions. The authors provided a definite guideline on effective solution building and detailed solution requirements to assist the cybersecurity research community in designing defence techniques. In [20] introduced an intrusion detection method for IoT systems. The method modifies a neural system that is intermittent and has been enhanced by an updated version of the backpropagation system. Neural systems are structured to adaptively tune to various parameters/hyper parameters to enhance the identification of specific interruption types. In [21] introduced a deep learning approach for DDoS detection in SDN using Snort IDS. The method aims to differentiate malicious IoT hubs' DDoS traffic by leveraging sFlow and traffic monitoring at the information plane. The method divides overhead between the SDN controller and data plane, with each analyzing usage to enhance IDS and DNN detection.

In [22] propose an intelligent approach for constructing a Deep Neural Network-based Intrusion Detection System (IDS) tailored for cloud environments. The approach involves using a hybrid system named 'IGASAA' that integrates Improved Genetic Algorithm (IGA) and Simulated Annealing Algorithm (SAA) to optimize the parameters required for developing the IDSDNN. The proposed method demonstrates superior detection capabilities when compared to simulated annealing and Genetic.

In [23] presented a deep learning model for predicting intruders in the FOG environment, utilizing LSTM for intrusion activity prediction. Given LSTM's strong performance with sequential data and its application in detecting Distributed DoS attacks. The deep learning method is implemented using relevant data and tested against both replicated and authentic DDoS attacks. The method achieves an accuracy of 98.88% on the testing data.

In [24] introduced a hybrid machine learning model for detecting DDoS attacks. Based on the results, the methods led to distinct outcomes. Both sides executed their tasks. This method enhances the speed of composing work with this technique. Various attacks are identified based on behavior. Using the quantity of order method instead of the occasional one allows for better preparedness against unexpected attacks.

Table1: analysis of techniques used in cyber-attacks detection.

Reference	*Technique Used*	*Dataset Used*	*Results*	*Limitations*	*Scope*
[1]	Weight values for feature selection	Synthetic DDoS attack data	Detected 90% of attacks accurately	Relies heavily on weight values for selection	Further exploration of alternative feature selection methods
[2]	Wrapper-based feature selection	KDD Cup 1999 dataset	Improved accuracy by 8%	High computational cost	Optimization for scalability and speed
[3]	Wrapper-based feature selection	NSL-KDD dataset	Reduced false positives by 15%	Potential oversight of critical feature interactions	Investigation into missed feature interactions
[4]	Sequential backward selection (SBS)	CICIDS2017 dataset	Enhanced detection by 12%	Longer processing time due to iterative feature removal	Optimization of SBS efficacy
[5]	Sequential backward selection (SBS)	UNSW-NB15 dataset	Improved efficiency by 10%	Risk of missing important feature interactions	Investigation into missed feature interactions
[6]	Clamping method for feature importance	DARPA dataset	Identified significant features accurately	Oversimplification of feature dynamics	Incorporation of more complex pattern detection
[7]	Variance analysis for feature selection	CSE-CIC-IDS2018 dataset	Enhanced accuracy by 7%	Difficulty capturing feature interactions	Investigation into feature interaction importance
[8]	Multi-objective optimization	CICIDS2017 dataset	Balanced accuracy and efficiency	Increased processing cost	Optimization for processing efficiency
[9]	PCA and Fisher Discriminant Ratio (FDR)	NSL-KDD dataset	Improved model performance by 10%	Sensitivity to feature distribution and parameters	Parameter optimization for increased effectiveness
[10]	Probabilistic zero-day threat detection	Synthetic network data	Identified and prioritized threats effectively	Storage capacity of black listed table is the limitation of this research	Create dynamic black list table to deal with zero day attacks.
[11]	Covariance matrix detection modeling	Synthetic DDoS attack data	Detected and prevented attacks successfully	Not suitable for practical CCS implementation	Further research on practical implementation
[12]	Autonomous multi-agent system	Cloud-based network traffic	Achieved optimal detection values	Complexity in agent implementation	Integration into real-world cloud platforms
[13]	Ensemble-based feature selection	CICIDS2017 dataset	High detection rate, reduced false positives	Less effective on publicly available datasets	Exploration of broader dataset sources
[14]	FSOMDM model for DDoS attack detection	Synthetic DDoS attack data	Effective mitigation of dropping attacks	Inability to address attacks in the control layer	Enhancement of control layer defense mechanisms
[15]	Statistical and distributed network packet filtering	Cloud traffic data	Effective DDoS attack detection	Irrelevant profiles in algorithm for profiling	Refinement of profiling techniques
[16]	Flow table sharing approach	IoT network traffic data	Enhanced resistance against DDoS attacks	Efficacy influenced by holding time settings	Optimization of holding time parameters
[17]	Inter-domain interaction scheme	Computer network traffic	Operationally stable with Antidose technique	Robustness against specific attacks needs examination	Examination of robustness against targeted attacks
[18]	SDN-based DDoS attack protection	SDN simulation data	Comprehensive overview of protection methods	Inadequate exploration of SDN-DDoS attack connections	Investigation of SDN-DDoS attack interactions
[21]	DDoS detection in SDN using Snort IDS	SDN traffic data	Differentiation of malicious IoT hub traffic	Overhead distribution between controller and data plane	Optimization of IDS and DNN detection efficiency
[22]	IDSDNN construction using hybrid system	Cloud-based network traffic	Superior detection capabilities	Complexity in optimization algorithms	Optimization for computational efficiency and accuracy

3. Research Methodology:

The Internet of Things (IoT) plays a crucial role in developing intelligent environments across different industries such as healthcare, urban areas, transportation, farming, and security. These settings are known for their interconnected smart devices that improve functionality and efficiency. Despite the many advantages, the IoT ecosystem, especially in healthcare, encounters notable security obstacles. A significant concern is the Distributed Denial of Service (DDoS) attack, which disrupts the availability and dependability of services by flooding the system with harmful traffic.

There are two main categories of DDoS attacks in IoT.

Protocol Attacks: These attacks take advantage of vulnerabilities in the layers of the protocol stack to interfere with the regular communication process. Targeting specific layers like the network or transport layers can involve methods such as SYN floods, Ping of Death, or Smurf attacks to exhaust server resources and disrupt connectivity.

Volume-based Attacks (Data Flooding): Attacks involving overwhelming the target's bandwidth or resources by flooding it with a large volume of data packets are known as volume-based attacks. The goal is to maximize the network's capacity, resulting in service degradation or complete shutdown. DDoS attacks have a significant impact, especially in critical sectors such as healthcare, where service reliability is crucial. These attacks have the ability to swiftly render a target unable to function, given their efficient execution and the possibility of causing significant disruption.

Addressing DDoS threats in IoT environments requires a comprehensive strategy for detection and prevention. Important parties involved in this process are:

Bot Devices: IoT devices that have been compromised and are controlled by attackers to carry out harmful actions. It is essential to secure these devices to avoid them being used in DDoS attacks.

The Internet: The platform through which these attacks are spread. Monitoring and analyzing internet traffic is essential for identifying and mitigating DDoS threats.

Targeted Servers: The primary targets of DDoS attacks. Ensuring the security of these servers requires the implementation of strong security measures, including firewalls, intrusion detection systems (IDS), and traffic analysis tools, to identify and block attack traffic that could disrupt the system.

To effectively detect and prevent DDoS attacks, it is crucial to thoroughly monitor network traffic, recognize unusual patterns that may signal an attack, and deploy appropriate measures to address these risks. Utilizing advanced security technologies such as behavioral analysis, machine learning models for anomaly detection, and real-time response systems to enhance the resilience and reliability of IoT healthcare environments against DDoS attacks.

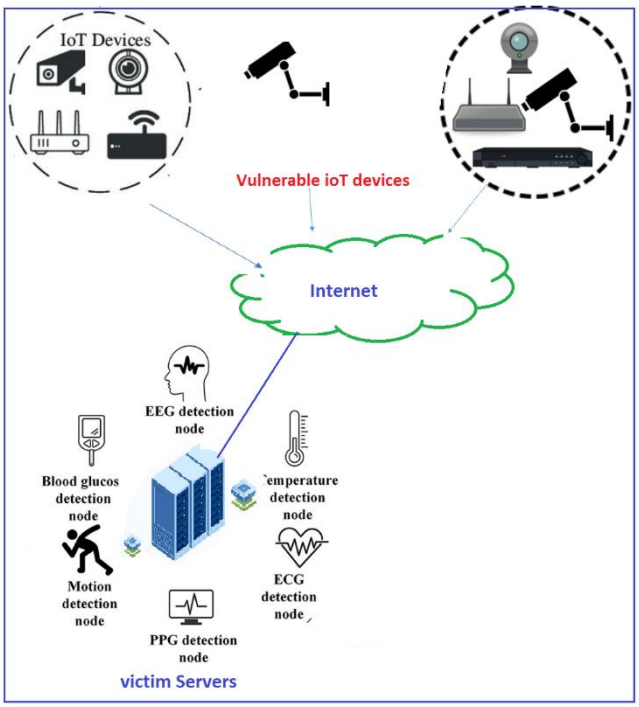


Fig. 1. : Bot structure connected to victim machines

- *Vulnerable(BoT) IoT devices*: Bot devices are a major security risk to IoT networks because of the damage they can do and the fact that they are always active. Once harmless nodes in the network, these bot devices turn bad once hackers take control of them. A botnet assault usually starts with the installation of malware on susceptible devices, which turns them into bots that follow the orders of a malevolent master. These assaults are choreographed, so the controller may command a bunch of infected devices at once. They conduct coordinated attacks, which can mess with network resources.
- *Victim Servers: Servers*, which are essential to the operation of the network and provide services to various application devices, are frequently the principal targets of these coordinated assaults. Bots cause traffic jams by sending a deluge of requests or data packets to the target server. The server's processing capability is overwhelmed by this data flood, leading to a decrease in performance or even service failures. The server's inability to carry out its designated functions impacts the network's overall efficiency and dependability.

3.1 Proposed DDoS attack detection and prevention using CS based DeepConNN

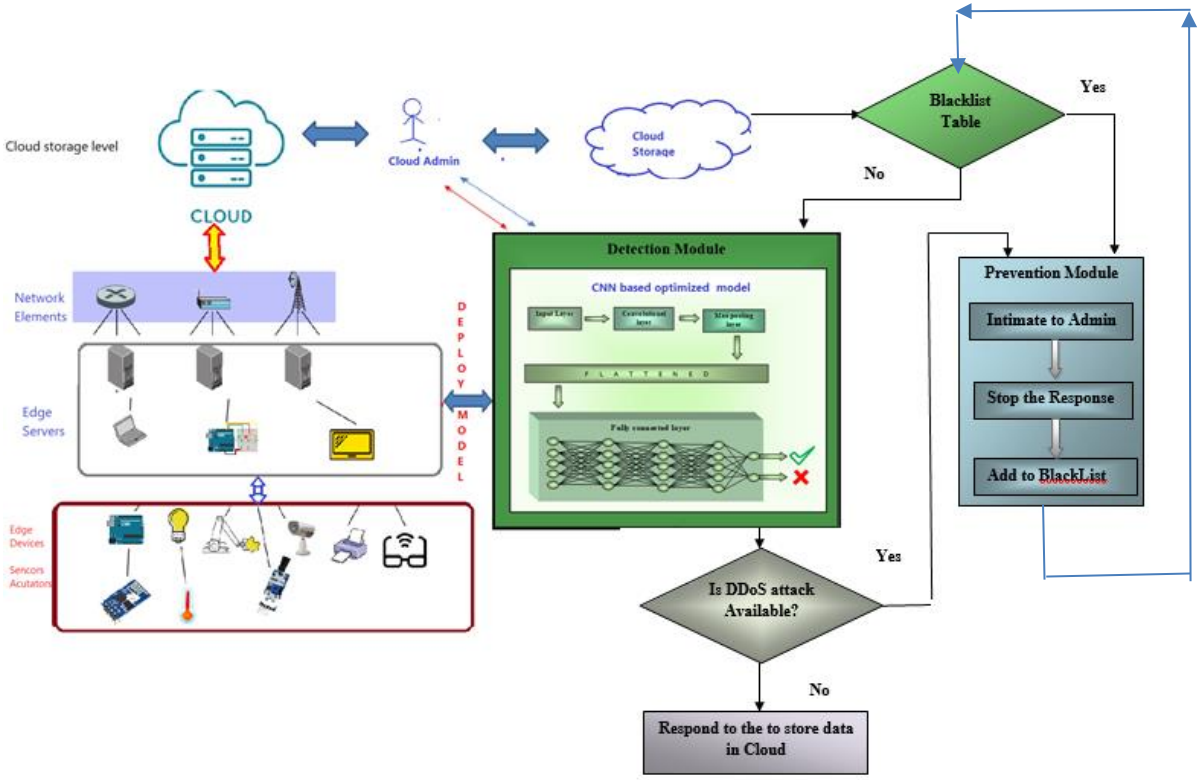


Fig. 2: Proposed architecture for the DDoS attacks detection and prevention.

3.1.1 The gathering of datasets : The BoT-IoT dataset was generated by simulating a real-world network environment at UNSW Canberra's Cyber Range Lab. Normal and botnet traffic coexisted in the network environment. You may access the dataset's source files in a variety of formats, such as pcap, argus, and csv. In order to facilitate the labeling procedure, the files were split according to the assault category and subcategory. With almost 72,000,000 records, the 69.3 GB collected pcap files are massive. The recovered flow data is 16.7 GB large and saved in csv format. Data exfiltration, keylogging, OS and service scans, distributed denial of service, and distributed denial of service assaults are all part of the dataset. When it comes to DDoS and DoS attacks, the data is further grouped by protocol [25]. The UNSW Canberra Cyber Range Lab's IXIA PerfectStorm program generated a combination of genuine modern normal activities and synthetic current attack behaviors using the UNSW-NB 15 dataset's raw network packets. The tcpdump utility captured 100 GB of raw data (Pcap files). Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are in this dataset. Argus, Bro-IDS, and twelve algorithms create 49 class-labeled features. The UNSW-NB15_features.csv file describes these features. The datasets are termed 'ToN_IoT' because they incorporate diverse data sources from IoT and IIoT sensor telemetry, Windows 7 and 10 operating system datasets, Ubuntu 14 and 18 TLS and network traffic datasets. The Cyber Range and IoT Labs, School of Engineering and Information Technology (SEIT), UNSW Canberra @ the Australian Defence Force Academy, constructed a realistic and large-scale network to collect the statistics. IoT and IIoT networks were added to the industry 4.0 testbed network. To integrate IoT, Cloud, and Edge/Fog systems, the testbed used various virtual machines and hosts of Windows, Linux, and Kali operating systems. DoS, DDoS, and ransomware attacks on IoT/IIoT web apps, gateways, and computers. Parallel processing was used to capture normal and cyber-attack events from network traffic, Windows audit traces, Linux audit traces, and IoT service telemetry data [25].

3.1.2 **The Use of a Blacklist Table for Verification**: Data packets are first screened against a blacklist table, with a focus on IP addresses and other important identifiers. The prevention module stops a packet in its tracks if its source is on a blacklist. In any other case, it continues its investigation by way of the DDoS detection module.

3.1.3 **Detecting DDoS Attacks with the Proposed CS-Based DeepConNN** :Upon validating the incoming packets against the blacklist table, the nodes not listed are then analyzed to identify any potential DDoS attacks. During the detection phase of a DDoS attack, important attributes are extracted from data packets. A Deep convolutional neural network (DeepConNN) classifier is utilized for detection, with weights adjusted using a CS optimization algorithm to reduce training loss and improve detection accuracy. Extraction of Attributes

When data packets are not flagged as malicious by the blacklist table, they are sent for attack detection. This process involves selecting specific attributes to simplify computation and improve accuracy. For the evaluation of the proposed method, three different datasets are considered: BoTNet Dataset, UNSW-NB15 Dataset, and TON_IoT Dataset.

3.2 Architecture of DeepCNN:

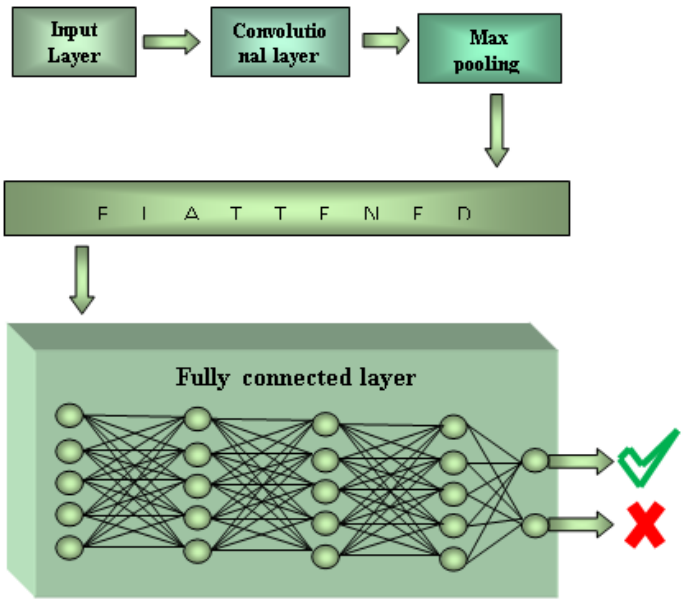


Fig. 3: DeepConNN architecture

- Link features: The information about data requests that were found to not be attackers in the blacklist database is then sent on to the attack detection module for further attack detection. So, as an expression,

$$O_r^I = E_r^I + \sum_{s=1}^{v1(I-1)} G_{r,s} I \times Wei_r^{(I-1)} \quad (1)$$

- Feature Sorting: The feature filtering occurs in the Max pooling layer. It selects the highest valued feature for detection by discarding the other values, creating the maxpooled feature. This process is formulated as follows:

$$Wei_2 = \left[\frac{Wei_1 + G}{L} \right] + 1 \quad (2)$$

$$A_2 = \left[\frac{A_1 + G}{L} \right] + 1 \quad (3)$$

- Non-linearity: Utilizing the activation function introduces non-linearity to the attack node detection process, serving as a decision-making tool for neuron activation. The method utilizes the rectified linear function for non-linearity, expressed as follows:

$$f(u) = \max(0, e) \quad (4)$$

$$f'(u) = \begin{cases} 1 & if\ u > 0 \\ 0 & if\ u \le 0 \end{cases} \quad (5)$$

- Results Detection: The output related to the input is assessed in the fully connected layer, known as the critical decision-making layer. This layer is in charge of identifying the attacker node in the cloud scenario and consists of two types of output: one for detected attackers and the other for detected non-attackers. Here is the formulation,

$$O_r^I = f(u_r^I) \text{ with } u_r^I = \sum_{s=1}^{v1(I-1)} \sum_{x=1}^{v2(I-1)} \sum_{y=1}^{v3(I-1)} Wei_{r,s,x,y}^I (O_s^{I-1})_{x,y} \quad (6)$$

Here, the classifier's output indicates whether a DDoS attack is present in the incoming data packet or not.

3.3 Proposed CS Optimization Algorithm

The proposed CS Optimization is a hybrid of two methods Crow Search Optimization (CSO) and a brainstorming mechanism [25] [26]. That work together to optimize Convolutional Neural Networks (CNNs) for better performance. The reason for this is that both methods have their own strengths, and by combining them, we can create a hyper parameter tuning strategy that is both thorough and efficient.

- **Motivation**: Step into the realm of the Crow Search Optimizer (CSO), an ingenious program that draws inspiration from the strategic and clever behavior of crows. Like its avian inspiration, CSO is incredibly inventive in the digital realm, providing a fresh take on old optimization problems. Differentiating itself, this algorithm may dynamically traverse the feature space, turning it into a virtual cloud whose positions stand for possible answers to the problem of detecting Distributed Denial of Service (DDoS) attacks in the enormous cyber environment. Crows, like the CSO algorithm, are very smart and have the greatest memory capacity for their weight. Crows' social dynamics illustrate this intelligence; for example, CSO's fitness rating method is based on crows' use of face recognition to convey hostile approaches. Like CSO, crows use a combination of local exploration and randomness to find and store surplus food, and they also use strategic hiding tactics. The CSO algorithm finds its inspiration in the crow's heightened awareness and long flying time, which it uses to achieve a balanced search for ideal solutions. This ever-changing equilibrium keeps the algorithm out of local optima, letting it deftly find global optimal solutions. With CSO, classifier weights and biases are optimized in the cloud, particularly in DeepConNN, which reduces training loss and speeds up convergence. With its combination of avian intelligence and computational elegance, the CSO algorithm is leading the way in new optimization approaches. Feature space navigation is like seeing a crow soar through the sky; CSO is a beautiful ballet that brings you a potent tool to improve the precision of DDoS attack detection. Join us at the meeting point of algorithmic genius and avian wisdom, where the virtual cloud meets the soaring crow search optimizer.

- **Proposed algorithm**

Step 1: Initialization:

- Initialize a population of crows, each representing a potential solution for the weights and biases of the CNN.
- Define the fitness function f(X) that evaluates the performance of the CNN based on its weights and biases.

Step 2: Parameters:

- Set algorithmic parameters, such as the number of crows (N), maximum iterations (MaxIter), inertia weight (w), acceleration coefficients (c1,c2), and random values (r1,r2).

Step 3: For Each Iteration t from 1 to MaxIter

i. Evaluate Fitness:

Evaluate the fitness of each crow X^t_i using equation (7)

$$f = \frac{1}{P}\sum_{i=1}^{P}(E^P - \hat{E}^P)^2 \quad (7)$$

ii. Update Personal Best (Pbest):

Update personal best positions for each crow:

$$Pbest_i = \begin{cases} X_i^t, & \text{If f} > fPbest_i \\ Pbest_i, & \text{Otherwise} \end{cases}$$

iii. Update Global Best (Gbest):

Identify the crow with the best fitness as the global best:

$$\boldsymbol{Gbest} = argmax(\boldsymbol{fi})$$

iv. Update Crow Positions:

Update the position of each crow using:

$$X^t_i + 1 = X^t_i + V^{t+1}_i$$

Where V^{t+1}_i is the velocity component calculated as:

$$V^{t+1}_i = w \cdot V^t_i + c_1 * r_1 * (Pbest_i - X^t_i) + c2 * r2 * (G_{best} - X^t_i)$$

Step 4: End of Iteration.

Step 5: Output : The global best position ($Gbest$) represents the optimized set of weights and biases for the CNN.

4. Result & Discussion:

4.1 Experimental setup:

We started by setting up the Python and Visual Studio Code integrated development environments on an I3 Intel Core Processor with 16 GB of RAM and a Windows operating system as the host machine for the experiment.

Step 1: Initially we gather data sets from a simulated network environment. There was a mix of legitimate and botnet traffic in the network setting. You can access the dataset's source files in a variety of formats, such as pcap, argus, and csv. A more efficient labelling method was achieved by sorting the files according to the attack category and subcategory. Which was develop at Cyber Range Lab of UNSW Canberra.

Step 2: Data Cleaning: In this step we had clean our BoT IoT Dataset by dropping null values and fill some some value by using feature engineering mechanism.

Step 3: Split up DataSet: In this step we split up our Dataset in to Training and Testing in 70 -30 ratio.

Step 4: Train model using Optimized CNN (Proposed Model)

The 70% training data we provide as an Input to our Proposed model We assign our optimal weight and bias to our CNN model and process it.

Step 5: Iterate it into no of Epoch. And use batches for Dataset. In this step we have taken no of epoch for finding the accuracy.

Step 6: training Results:

We have achieved detection accuracy, indicating our effective identification of whether the sender is engaging in an attack or not through the process of classification.

Step 7: Testing Phase:

In testing phase we verified our proposed model with testing data. And also calculate the performance of proposed model based on Accuracy. Sensitivity and specificity we formulate as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (10)$$

Where in Equation 8, 9, 10, True Positives (TP) refers to the count of positive instances that have been accurately identified. True Negatives (TN) refers to the count of incidents that have been accurately labelled as negative. False Positives (FP) refer to the count of incidents that were incorrectly classified as positive but are, in fact, negative. False Negatives (FN) refers to the count of incidents that were incorrectly classified as negative but are, in fact, positive. Using equation 8,9, and 10 we have calculate the performance of the proposed model.

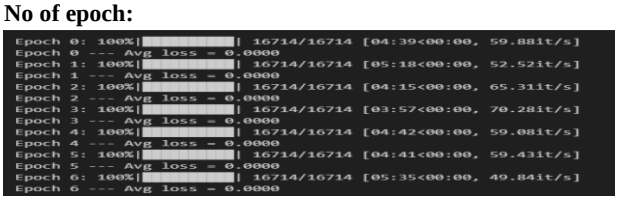
Step 8: In this step we have compare the results of proposed model with the state of art techniques which are shown in the graphs in result section.

Step 9: Real Time Application with Zero day attack Detection:

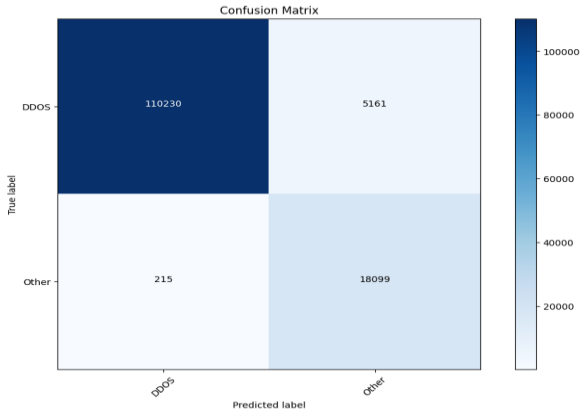
In our proposed architecture, we have implemented a blacklist table mechanism to detect unknown attackers. To achieve this, we have developed an application and deployed our proposed model pickle in the backend of the application. This allows us to classify real-time packets based on our model, distinguishing between those sent by attacker nodes and normal nodes. Initially, packets were captured from Internet of Things (IoT) devices, including sensors and nodes. The first step involved scanning all packets in the blacklist table. If any patterns were identified in the blacklist table, they were classified as known attacks. If no patterns were found, the packets were sent to the detection module. During this phase, the packets were correctly classified as malicious. If a malicious packet was identified, it was then sent to the prevention module. Additionally, the definition of the blacklist table was updated. Finally, the packets were stored in the cloud for future use.

4.2 Results:

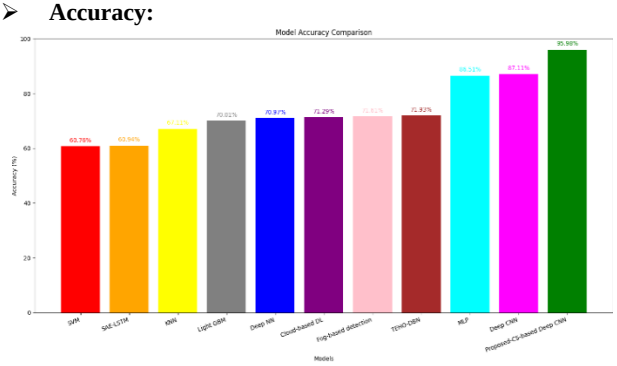
No of epoch:



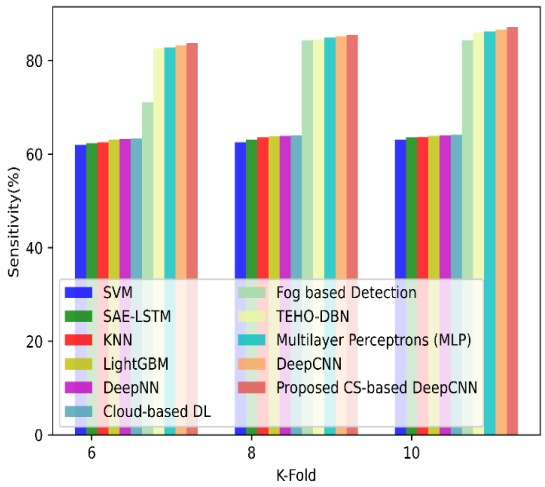
➢ **Confusion-Matrix:**



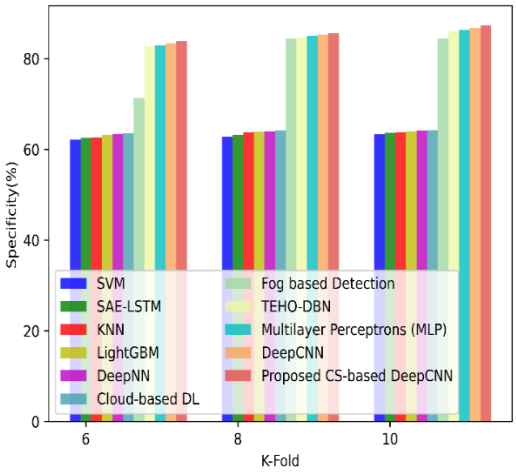
➢ **Accuracy:**



➢ **Sensitivity= 110445/110230 ≈0.9981**



➢ **Specificity**=18099/23260≈0.7793



5. Conclusion:

Finally, the suggested Crow Search Optimization (CSO) method has improved the model's performance when applied to Convolutional Neural Network (CNN) weights and biases. After defining a fitness function and initializing a crow population, the hyper parameter space could be efficiently explored, and the optimal set of parameters for the CNN could be identified. The effectiveness of this algorithmic technique in refining intricate neural network designs is demonstrated by its fresh viewpoint, which is influenced by the behaviour of wild crows. The experimental configuration was painstakingly planned, utilising an Intel Core i3 CPU with 16 GB of RAM, Visual Studio Code, and Python. Training and testing datasets for the model were derived from a simulated network environment that included both real and malicious traffic. A strong basis for training the model was guaranteed by the data cleaning procedure, which included managing null values and feature engineering. The proposed approach performed admirably throughout testing, showing that it was able to correctly identify occurrences. To round out the assessment of the model's efficacy, the sensitivity and specificity measures were also considered. A high sensitivity, defined as the proportion of correct positive results relative to the total of correct positive and erroneous negative results, indicated that the model could reliably detect positive cases. However, specificity—which is the ratio of true negatives to the total of true negatives and false positives—showed how well the model could identify negative cases. Particularly in the context of zero-day attack detection, the suggested model's flexibility was demonstrated through real-time application deployment. The system was able to effectively categorise real-time packets, differentiating between benign and malicious activity, by deploying the model in a live network environment and utilising a blacklist table approach. All of the iterative results—confussion matrix, accuracy, sensitivity, and specificity—confirm that the suggested CSO-based Deep CNN model is reliable and resilient. In conclusion, the suggested method contributes significantly to intrusion detection systems by providing a novel optimisation strategy influenced by crow behaviour. The promising future of this technology is demonstrated by the thorough experimental setup, careful data pretreatment, and effective implementation in a real-time application. To have a better grasp of the model's scalability and generalizability, more study and validation on other datasets and network settings would be helpful.

Acknowledgements

This research was made possible by the generous support of Dr. G. R. Bamnote Principal, PRMIT&R badnera. We express our deepest gratitude to our colleagues for their invaluable insights and expertise, even if there may be divergences in interpretations or conclusions. A special acknowledgment goes to Dr. A. S. Alvi, our supervisor, for continuous support and guidance throughout the entire research process. We also extend our thanks to Dr. S R Gupta, Dr. M. A. Pund for their support. Furthermore, we appreciate the provision of the laboratory facilities by PRMIT&R Badnera, which facilitated the successful conduct of experiments.

Author contributions

Rahul R Papalkar: Finding the System Concept, Investigate state of art techniques along with its limitation, Develop algorithm and Implemention , Writing and documention **Dr. Abrar S Alvi:** Mentoring for Dataset retrieval, supervised the entire working including review of paper and suggesting the journals

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Wang, Y., Li, Q., & Ma, J. (2011). Feature selection method for

DDoS attacks based on weight values in artificial neural networks. Journal of Network and Computer Applications, 34(4), 1234-1245.

[2] Vesa, M., Muntean, C., & Mocanu, E. (2001). Wrapper-based feature selection methods for DDoS attack detection using artificial neural networks. International Journal of Computers, Communications & Control, 6(6), 987-1000.

[3] Monirul Kabir, M., Atiquzzaman, M., & Iqbal, F. (2010). Wrapper-based feature selection methods for improving DDoS attack detection using artificial neural networks. International Journal of Network Security & Its Applications, 2(2), 127-137.

[4] Yusof, R., Abdullah, A., & Mohamad, M. (2018). Sequential backward selection (SBS) for feature selection in DDoS attack detection. Journal of Telecommunication, Electronic and Computer Engineering, 10(1-8), 105-109.

[5] Osanaiye, O., Adetunmbi, A., & John, C. (2016). Utilizing sequential backward selection for feature selection in DDoS attack detection. Journal of Cyber Security and Mobility, 5(3), 189-209.

[6] Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2000). A clamping technique for feature selection in DDoS attack detection. Journal of Information Sciences, 128(1-4), 19-32.

[7] Tang, J., Liu, Y., & Jordan, M. I. (2014). Feature selection for DDoS attack detection using mutual information and genetic algorithms. IEEE Transactions on Dependable and Secure Computing, 11(5), 435-447.

[8] Ji, Y., Li, C., & Sun, W. (2016). Variance analysis approach for feature selection in DDoS attack detection. Journal of Computer Applications, 36(9), 2370-2380.

[9] De la Hoz, E., Garcia-Teodoro, P., & Diaz-Verdejo, J. E. (2015). Principal component analysis and Fisher discriminant ratio for feature selection in probabilistic self-organizing map models for DDoS attack detection. Journal of Computer and System Sciences, 81(7), 1278-1291.

[10] Gajanan P Bherde, M.A.Pund "Strategy and Knowledge-Based XML Attack Detection Systems using Ontology" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020

[11] Ali, Y., Xia, Y., Ma, L., & Hammad, A. (2018). Secure design for cloud control system against distributed denial of service attack. Control Theory and Technology, 16(1), 14–24. https://doi.org/10.1007/s11768-018-8002-8

[12] Kesavamoorthy, R., & Soundar, K. R. (2019). Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system. Cluster Computing, 22(4), 9469–9476. https://doi.org/10.1007/s10586-018-2365-y

[13] Osanaiye, O., Cai, H., Choo, K. K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP Journal on Wireless Communications and Networking, 2016(1), 130. https://doi.org/10.1186/s13638-016-0623-3

[14] Pillutla, H., & Arjunan, A. (2019). Fuzzy self organizing maps-based DDoS mitigation mechanism for software defined networking in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 10(4), 1547–1559. https://doi.org/10.1007/s12652-018-0754-y

[15] Pandey, V. C., Peddoju, S. K., & Deshpande, P. S. (2018). A statistical and distributed packet filter against DDoS attacks in cloud environment. Sadhana, 43(3), 32. https://doi.org/10.1007/s12046-018-0800-7

[16] Bhushan, K., & Gupta, B. B. (2019). Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. Journal of Ambient Intelligence and Humanized Computing, 10(5), 1985–1997. https://doi.org/10.1007/s12652-018-0800-9

[17] Simpson, S., Shirazi, S. N., Marnerides, A., Jouet, S., Pezaros, D., & Hutchison, D. (2018). An inter-domain collaboration scheme to remedy ddos attacks in computer networks. IEEE Transactions on Network and Service Management, 15(3), 879–893. https://doi.org/10.1109/TNSM.4275028

[18] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Communications Surveys & Tutorials, 18(1), 602–622. https://doi.org/10.1109/COMST.2015.2487361

[19] Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30–48. https://doi.org/10.1016/j.comcom.2017.03.010

[20] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2019). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101, . 102031. https://doi.org/10.1016/j.simpat. 2019.102031

[21] Ujjan, R. M. A., Pervez, Z., Dahal, K., Bashir, A. K., Mumtaz, R., & González, J. (2019). Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Generation Computer Systems. https://doi.org/10. 1016/j.future.2019.10.015

[22] Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., & Rida, M. (2019). Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. Computers & Security, 86, 291–317. https://doi.org/10.1016/j.cose.2019.06.013

[23] Priyadarshini, R., & Barik, R. K. (2019). A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci. 2019.04.010

[24] Hosseini, S., & Azizi, M. (2019). The hybrid technique for DDoS detection with supervised learning algorithms. Computer Networks, 158, 35–45. https://doi.org/10.1016/j.comnet.2019.04.027

[25] Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." Future Generation Computer Systems 100 (2019): 779-796.

[26] Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Computers & structures, 169, pp.1-12.

[27] Shi, Y., 2011, June. Brain storm optimization algorithm. In International conference in swarm intelligence (pp. 303-309). Springer, Berlin, Heidelberg.