
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 26, Issues 4, pp. 1789–1796, DOI 10.12694/scpe.v26i4.4452

MECHANISM FOR DETECTING DOMAIN NAME SYSTEM BASED DENIAL OF SERVICE ATTACKS

SHUWEN LI*

Abstract. The Domain Name System (DNS) is a critical component of the internet infrastructure, responsible for translating human-readable domain names into IP addresses. However, the DNS is vulnerable various attacks like DNS cache poisoning, DNS tunneling, denial of service (DoS), etc. Thus, an effective attack detection mechanism is required to prevent the malicious entry in the DNS. In this article, an Elman Neural Network-based attack detection mechanism was proposed to predict the normal and malicious traffic in DNS system. The proposed model utilizes Recursive Feature Elimination (RFE) approach to extract and select most relevant features to train the ENN model. The proposed work predicts the incoming network traffic as normal or malicious based on the trained feature set. Furthermore, an alert notification module was designed to notify the administrator about the entry of attack. The proposed model was trained, tested and validated with the ICS DNS dataset and the outcomes are estimated. The developed model earned greater performances of 99.89% accuracy, 99.76% precision, 99.59% recall, and 99.68% f-measure. Furthermore, the estimated outcomes are compared with some recent optimization and deep learning-based attack detection techniques. From the comparative assessment, it is observed that the performances are improved in the proposed technique compared to existing algorithms.

Key words: Domain Name System, Denial of Service attack, Optimization, Machine Learning, Neural Network

1. Introduction. The Domain Name System (DNS) is a critical component of the internet infrastructure, responsible for translating human-readable domain names into IP addresses that can be used by computers to communicate with each other [1]. However, DNS is vulnerable to various attacks, including DNS cache poisoning, DNS amplification, DNS tunneling, and DNS hijacking [2-5]. These attacks can lead to the disruption of internet services, data theft, and other cyber-security risks [6]. To address these challenges, a DNS attack detection system can be developed using intelligent techniques. The development of a DNS attack detection system is not without its challenges [7]. One of the significant challenges is the complexity of the DNS protocol, which makes it difficult to identify malicious activity from legitimate DNS traffic [8]. Additionally, attackers use a variety of techniques to obfuscate their activity, such as DNS tunneling, which can be difficult to detect using traditional techniques [9]. Another challenge is the sheer volume of DNS traffic on the internet, making it challenging to identify anomalies and potential attacks [10]. Moreover, there is a need to ensure that the detection system does not generate false positives or negatives, which can lead to unnecessary disruption or a lack of protection [11].

Several DNS attack detection systems exist today, which employ various techniques to identify malicious activity [12]. One such system is the DNS Intrusion Detection System (DNSIDS), which uses rule-based algorithms to identify known DNS attacks [13]. Another system is the Passive DNS Replication and Analysis (PDNS), which uses a passive DNS replication technique to detect and analyze DNS traffic for anomalies [14]. Additionally, machine-learning algorithms have been employed in DNS attack detection systems, such as the DNS-Based Malware Detection (DBMD) system, which uses machine learning to identify malware communication channels in DNS traffic [15]. Moreover, the DNS-Entropy system uses entropy analysis to detect DNS tunneling activity. This approach helps to identify emerging threats more quickly and improve the overall effectiveness of DNS attack detection systems [16]. Further, a collaborative DNS attack detection framework was designed to and identify complex patterns of behavior indicative of DNS attacks [17].

Recently, the Machine Learning (ML) algorithms are used in DNS attack detection to quickly predict the malicious traffic. These ML-based techniques utilizes large amount of network traffic data to analyze the patterns of DNS attacks. This high accuracy earned by the ML-based techniques enables to share information

*Shanxi Engineering Vocational College, Taiyuan, Shanxi, China. (Corresponding author, ShuwenLi6@126.com)

across different networks securely. However, these techniques consume more time to train the system and increase the computational complexity. Although various approaches are developed to predict the DNS attacks, they face challenges like high false-positive rates, low detection probability, and increased complexity. To resolve these issues, an intelligent DNS attack detection framework was proposed in this article.

The key contributions of the research is listed below,

- An intelligent attack detection framework was developed using the Elman Neural Network to detect the DNS-DoS attack. This model utilizes the publically available ISC DNS dataset (network traffic data) for the identification of attacks.
- A Recursive Feature Elimination (RFE) technique was applied to extract and select most relevant features from the dataset. The selected feature set if fed into the ENN model for model training.
- The ENN model uses the select relevant feature set to analyze the pattern of DNS-DoS attack. Thus, the ENN classifier detects the incoming network traffic data as normal or malicious. An alert notification module was created to notify the identification of attack to the network administrator.
- The proposed technique was implemented in MATLAB tool and the results are evaluated in terms of accuracy, precision, recall, and f-measure.

The arrangement of the presented research article is described as, the recent research works related to the DNS attack detection are reviewed in section 2, the system model of the attack detection is detailed in section 3, the proposed methodology is explained in section 4, the outcomes of the proposed technique are analyzed in section 5, and the conclusion of the research is described in section 6.

2. Related works. Few recent works related to the proposed work is described below,

Ömer KASIM et al [18] proposed a novel Deep Learning (DL)-based framework for the detection of DNS flood attacks. This framework utilizes the convolutional neural network (CNN) and long short-term memory (LSTM) to provide solution for direct identification of DNS flood attacks. This model was evaluated with CICIDS dataset derived from real world data. The DL structure with LSTM achieved less low false-positive rate compared to other techniques. However, the developed model is computationally intensive and is prone to over fitting.

Minzhao Lyu et al [19] designed Machine Learning (ML)-based algorithm to identify distributed DNS attacks. This proposed model examines the DNS traffic data and highlights the incoming DNS queries, and malicious entities query scans. Further, hierarchical graph architecture is deployed to monitor DNS activity. The proposed technique provides greater performances in real-time. However, the noisy features in the traffic data affect the performance of the ML algorithm.

Naotake Ishikura et al [20] presented DNS tunneling identification approach based on the cache-property-aware features. The proposed technique utilizes the LSTM-based filter and rule-based filter to extract the tunneling features. The integration of rule-based filter attains a higher rate of attack detection. In addition, it lowers the misdetection rate and quickly identifies the DNS tunneling attack. However, it cannot detect all different types of DNS tunneling attacks.

Tahmina Zebin et al [21] Artificial Intelligence (AI)-based Intrusion Detection System (IDS) for DNS over HTTPS attack detection. The proposed model utilizes the Random Forest (RF) classifier to categorize the network traffic as malware or normal. Further, a publically available CIRA-CIC- DoHBrw-2020 dataset was utilized to predict and classify the DNS over HTTPS attacks. However, the AI-based IDS produce false positives, which refer to the detection of a threat or attack when there is none.

Randhir Kumar et al [22] presented an IDS scheme using the fog computing technology to predict the Denial-of-service (DoS) attacks in DNS system. The performance of the developed model is determined by training the RF and optimized gradient boosting system. The robustness of the proposed technique is analyzed using an IoT dataset named BoT-IoT. The utilization of RF approach in fog computing reduces the testing and training time. However, this approach is not suitable for large-scale network.

Vinayakumar Ravi et al [23] proposed automatic attack detection strategy to identify the randomly generated domain names and DNS homograph attacks with high detection rate. The effectiveness of the developed scheme was analyzed against three different adversarial attacks: DeepDGA, CharBot, and MaskDGA. The results of the developed mode are compared with most popular DL algorithms. This model attained greater detection rate of 97.16%. However, it is vulnerable to adversarial attacks.

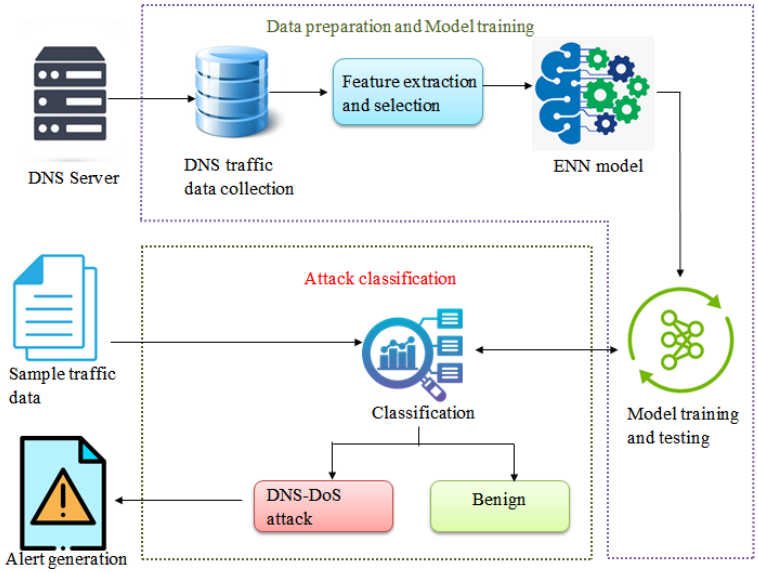


Fig. 4.1: Proposed Methodology

3. System model. The system model for DNS-DoS attack detection comprises a network of DNS servers, clients, and IDS placed in the network. The IDS monitors the DNS traffic and analyses the packets to identify any DNS-DoS attacks. The objective of the DNS-DoS attack detection system is to identify and block any DNS-DoS attacks in the network. Given a set of DNS packets, the system needs to classify each packet as either normal or anomalous. The IDS framework extracts relevant features from the DNS packets, such as DNS query rate, packet size, source IP address, destination IP address, and time of day. Then, it detects any anomalous patterns in the DNS traffic using anomaly detection algorithms such as clustering, statistical analysis, or machine learning. If an anomaly is detected, the system classifies it as a DNS-DoS attack based on a pre-defined set of rules or machine learning algorithms. Once a DNS-DoS attack is identified, the system needs to block the traffic from the identified source IP address using firewall rules or other network security measures. The system needs to send alerts to the network administrator or security team to notify them of the attack and the action taken.

4. Proposed ENN model for DNS-DoS attack detection. A novel intelligent attack detection technique was proposed using the Deep Learning (DL) algorithm. This technique utilizes the Elman neural network (ENN) to classify the DNS-DoS attacks. The presented model involves four phases namely: data collection, feature extraction and selection, ENN model training and attack classification. Initially, the network traffic data was gathered from the DNS server and fed into the system. The recursive feature elimination (RFE) approach was utilized to extract and select most relevant features from the DNS traffic dataset. Further, the ENN model was trained using the selected features for attack detection. In the classification phase, the system matches the trained attack pattern and detects the normal and DNS-DoS attack. Furthermore, the proposed model alerts the administrator to mitigate the attack. The proposed attack detection framework is demonstrated in Fig 4.1.

4.1. Data collection. The DNS-DoS attack detection mechanism begins with the data collection. In data collection phase, the network traffic data was collected from the DNS server. Capturing the network traffic data from the DNS server involves intercepting and analyzing the packets of data, which is transmitted over the network. The traffic data includes information such as source IP address, destination IP address, packet size, DNS queries, port numbers, etc. The dataset initialization is formulated in Eqn. 4.1.

$$N_{TA} = [d_{t1}, d_{t2}, d_{t3}, \cdots, d_{tk}] \tag{4.1}$$

where N_{TA} denotes the collected DNS traffic dataset, indicates the information present in the dataset, and denotes the total number of data present in the dataset.

4.2. Feature extraction and selection. Feature extraction is the process of selecting and converting the raw dataset into a sequence of meaningful features, which is utilized by the DL algorithm for attack detection. In the proposed work, the Recursive Feature Elimination (RFE) was utilized to extract and select the most relevant features. This algorithm works by recursively eliminating the least important features from the input dataset. Initially, the raw dataset was pre-processed to remove the duplicate and irrelevant information. The RFE model defines the filtered dataset as feature set to extract the meaningful features. Further, the features in the set are ranked and the feature with the lowest importance score was removed from the set. The feature extraction function is formulated in Eqn. 4.2.

$$R_{FE}(N_{TA}[d_t]) = \sum_{i=1}^{N} ||d_t(x) = d_t(\hat{x})||^2 \tag{4.2}$$

where R_{FE} denotes the RFE feature extraction function, and refers to the most relevant and least important feature present in the dataset. In this process, the most relevant DNS network traffic features like source IP address, destination IP address, protocol, request size, number of requests per second, response time, packet size, etc., are extracted. This process continues until all desired features are extracted from the dataset. The final set of features is selected for model training and testing purpose.

4.3. Elman Neural Network. The Elman neural network is a type of recurrent neural network (RNN), which is utilized in numerous applications like attack detection, time-series prediction, etc. Unlike other feed-forward neural network, which only connects the flow in one direction (input to output layer), RNN have feedback connections that permits them to utilize previous outcomes as input to the current iteration. Typically, the ENN model consists of three layers namely, an input layer, a hidden layer, and an output layer. The input layer receives the inputs to the network and the output layer provides the outcomes. The hidden layer is the main distinguishing attribute of the ENN. It consists of feedback connection from its own output to its input, enabling it to store the information from previous iterations and utilize it as context for the current iteration. The feedback connection is also termed as context layer. During training process, the input and output pairs are presented to the network and the networks output is compared to the desired output. In DNS-DoS attack detection, the ENN model was trained using the extracted relevant features of the network traffic dataset. The features like source IP address, destination IP address, protocol, request size, number of requests per second, DNS query, DNS response, etc., are fed into the input layer. To detect a DNS-DoS attack using the Elman neural network, the network takes in a sequence of feature vectors representing the network traffic over a period. The feature set fed into input layer of the ENN is expressed in Eqn. 4.3.

$$I(t) = \{f_1, f_2, f_3, \cdots, f_n\} \tag{4.3}$$

where $I(t)$ denotes the input feature set at time, indicates the extracted features, and refers to the total number of features present in the set. The network processes each feature vector and updates the hidden layer state based on the previous hidden layer state and the current input. The hidden layer of the ENN is represented in Eqn. 4.4.

$$H_i(t) = A_n(U * I(t) + W * H(t-1)) \tag{4.4}$$

where $H_i(t)$ indicates the hidden layer at time, A_n denotes the activation function, U defines the weight matrix between the input and hidden layer, and W refers to the weight matrix between the hidden layer and itself. The output layer of the ENN is expressed in Eqn. 4.5.

$$O_p(t) = R_n(V * H(t)) \tag{4.5}$$

where O_p denotes the output at time, R_n indicates the activation function, and V refers to the weight matrix between the hidden and the output layer. The training of the Elman neural network involves adjusting the

Table 5.1: Sample dataset features and its description

Feature Name	Description
query_datetime	Date and time of the DNS query
server_ip	IP address of the DNS server
client_ip	IP address of the client making the DNS query
query_name	The domain name being queried
query_type	The type of DNS query
query_class	The class of DNS query
response_code	The DNS response code
response_datetime	Date and time of the DNS response
response_ttl	The time-to-live value of the DNS response
response_address	The IP address returned in the DNS response

weight matrices U, W, and V to minimize the error between the predicted output and the actual output. The error is calculated using a cost function, such as mean squared error (MSE), and is back propagated through the network to update the weights. Thus, the system trains the model to identify the attack patterns. The classification of new incoming network traffic data is analyzed by comparing the trained attack pattern and incoming data pattern.

$$A_{CI} = \begin{cases} if(I_{dp} = T_{ap}); DNS - DoSAttack \\ else; Benign \end{cases} \tag{4.6}$$

where A_{CI} indicates the attack classification function, I_{dp} denotes the incoming network traffic data, and T_{ap} refers to the trained attack pattern. If the incoming network traffic data features match with the trained attack pattern, it is detected as attack. If the both features are not matched, it predicted as benign. Finally, an alert notification was designed alert network administrators or security personnel in real-time when an attack is detected, allowing them to take appropriate action to mitigate the attack and prevent further damage.

5. Results and discussion. A novel ENN-based attack detection framework was designed to predict the DNS-DoS attack effectively. This model utilizes the RFE approach to extract most relevant feature from the DNS traffic dataset. The ENN model was trained using the selected features to classify the traffic as normal or malicious. In addition, an alert notification module was developed to mitigate the attacks. The developed model was trained and tested with the ICS DNS dataset. The developed model was executed in MATLAB tool, version R2020a. Finally, the performances of the proposed work were analyzed and validated with a comparative analysis.

5.1. Dataset description. The ISC DNS Dataset is a publicly available dataset, which contains DNS traffic traces collected from a large number of DNS servers, including both authoritative and recursive servers. The dataset includes features like source IP, destination IP, query data time, client IP, response time, etc. The dataset was collected by the Internet Systems Consortium (ISC), a non-profit organization that supports the development of open-source software for the Internet infrastructure. The raw packet captures were collected using tcpdump and include both inbound and outbound DNS traffic. The dataset covers a period of several months and includes traffic from a variety of sources, including home networks, small and medium-sized businesses, and large enterprises. Table 5.1 tabulates sample dataset features and its description.

5.2. Comparative analysis. In comparative analysis section, the outcomes of the proposed work was compared with existing techniques like Grey wolf optimization (GWO) [24], Genetic Algorithm with Grey wolf optimization (GA_GWO) [25], Deep Belief Neural system (DBN) [26], One-Class Support Vector Machine Algorithm (OCSVM) [27], and Grasshopper Optimization (GOA) [28].

1. Accuracy. Accuracy defines the proportion of exact predictions made by the system over the total number of predictions. It measures the system capability to differentiate the normal and malicious DNS traffic

accurately. The formula for accuracy is expressed in Eqn. 5.1.

$$A'_{cq} = \frac{T'_p + T'_n}{T'_p + T'_n + F'_p + F'_n} \tag{5.1}$$

where A'_{cq} defines the system accuracy, T'_p, T'_n, F'_p, and F'_n andrefers to the true-positive, true-negative, false-positive, and false-negative, respectively.

2. Precision. Precision defines the proportion of the true positive detections made by the proposed system over the total number of positive predictions made by the system. It measures the system ability to identify the malicious DNS traffic accurately. The formula for precision calculation is represented in Eqn. 5.2

$$P'_{sc} = \frac{T'_p}{T'_p + F'_p} \tag{5.2}$$

where R'_{CL} denotes the precision.

3. Recall. Recall measures the proportion of true positive predictions over the total number of actual positive instances in the data. It is a measure system capacity to detect malicious DNS traffic. The formula for recall is expressed in Eqn. 5.3.

$$R'_{CL} = \frac{T'_p}{T'_p + F'_n} \tag{5.3}$$

where R'_{CL} denotes the recall.

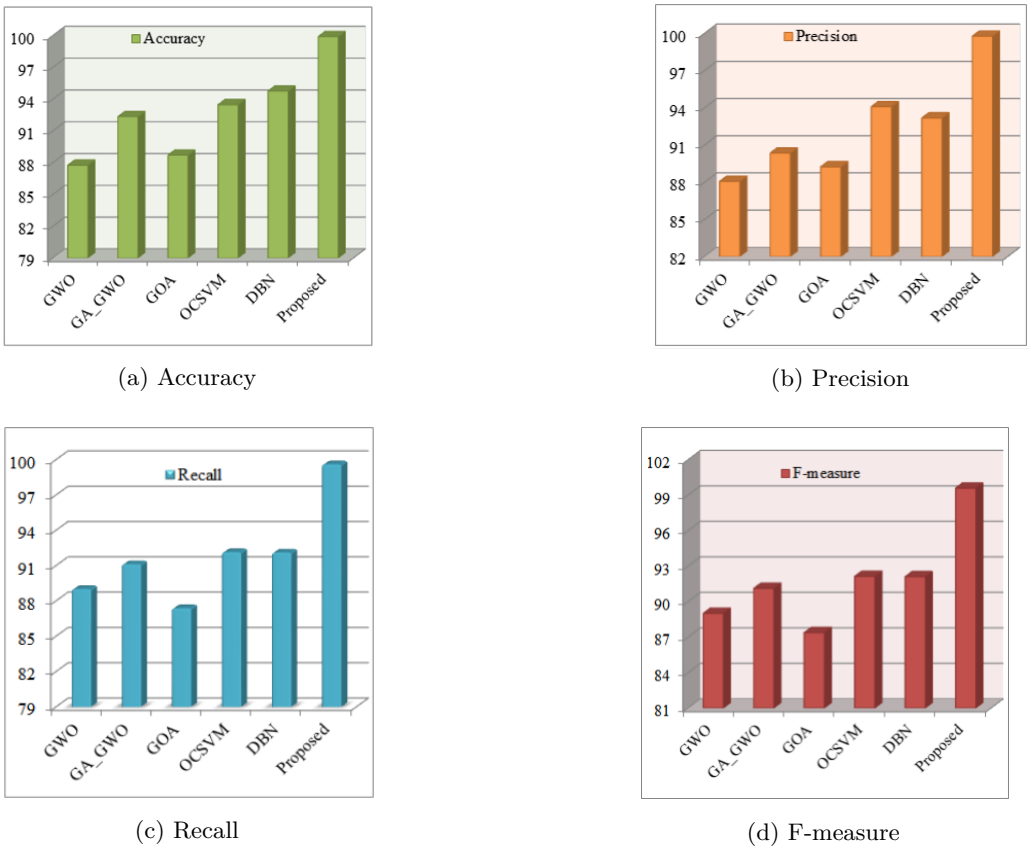
4. F-measure. F-measure represents the harmonic mean of precision and recall. It is a measure of the overall effectiveness of the system. The formula for F-measure is represented in Eqn. 5.4.

$$F_{me} = 2 \times \left(\frac{P'_{SC} + \times R'_{CL}}{P'_{SC} + R'_{CL}} \right) \tag{5.4}$$

where F_{me} refers to the F-measure.

The comparative analysis is illustrated in Fig 5.1. Here, the outcomes of the proposed technique were compared with existing techniques like GWO, GA_GWO, GOA, OCSVM, and DBN. The existing techniques outcomes are estimated by implementing it in the MATLAB tool for the same ICS DNS dataset. The comparative analysis proves that the proposed model attained greater results than the existing techniques. In addition, the performance enhancement score is determined from the comparative analysis.

6. Conclusion. This paper presents an intelligent attack classification framework to predict the DNS-DoS attack by analyzing the network traffic data. This model utilizes the ENN algorithm to classify the normal and malicious traffic. The developed model was tested and validated with the ISC DNS dataset containing network traffic information. Further, a RFE approach was applied to extract and select most relevant features from the input dataset. In the classification phase, the ENN was trained using the selected feature set to classify the malicious traffic. In addition, an alert notification module was designed to notify the detection the attack. The developed model was executed in the MATLAB tool and the outcomes are estimated. The comparative analysis demonstrates that in the proposed technique the performances like accuracy, precision, recall, and f-measure are improved by 5.13%, 5.68%, 7.46%, and 6.39%, respectively compared to the existing techniques like GWO, GA_GWO, GOA, OCSVM, and DBN.



(a) Accuracy (b) Precision

(c) Recall (d) F-measure

Fig. 5.1: Comparative Power Generation based on Solar Forecasting Power Generation based on Solar Forecasting analysis: (a) Accuracy, (b) Precision, (c) Recall (d) F-measure

REFERENCES

[1] Roopak, Monika, Gui Yun Tian, and Jonathon Chambers. "An intrusion detection system against ddos attacks in iot networks." 2020 10th annual computing and communication workshop and conference (CCWC). IEEE, 2020.

[2] Mayuranathan, M., M. Murugan, and V. Dhanakoti. "Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment." Journal of Ambient Intelligence and Humanized Computing 12 (2021): 3609-3619.

[3] Saranya, T., et al. "Performance analysis of machine learning algorithms in intrusion detection system: A review." Procedia Computer Science 171 (2020): 1251-1260.

[4] Meryem, Amar, and Bouabid EL Ouahidi. "Hybrid intrusion detection system using machine learning." Network Security 2020.5 (2020): 8-19.

[5] Sarhan, Mohanad, Siamak Layeghy, and Marius Portmann. "Towards a standard feature set for network intrusion detection system datasets." Mobile networks and applications (2022): 1-14.

[6] Verma, Abhishek, and Virender Ranga. "Machine learning based intrusion detection systems for IoT applications." Wireless Personal Communications 111 (2020): 2287-2310.

[7] Nadeem, Muhammad, et al. "Intercept the cloud network from brute force and DDoS attacks via intrusion detection and prevention system." IEEE Access 9 (2021): 152300-152309.

[8] Nimbalkar, Pushparaj, and Deepak Kshirsagar. "Feature selection for intrusion detection system in Internet-of-Things (IoT)." ICT Express 7.2 (2021): 177-181.

[9] Azeez, Nureni Ayofe, et al. "Intrusion detection and prevention systems: an updated review." Data Management, Analytics and Innovation: Proceedings of ICDMAI 2019, Volume 1 (2020): 685-696.

[10] Newaz, AKM Iqtidar, et al. "Heka: A novel intrusion detection system for attacks to personal medical devices." 2020 IEEE Conference on Communications and Network Security (CNS). IEEE, 2020.

[11] Abrar, Iram, et al. "A machine learning approach for intrusion detection system on NSL-KDD dataset." 2020 international conference on smart electronics and communication (ICOSEC). IEEE, 2020.
[12] Mendonça, Robson V., et al. "Intrusion detection system based on fast hierarchical deep convolutional neural network." IEEE Access 9 (2021): 61024-61034.
[13] Kumar, Vikash, et al. "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset." Cluster Computing 23 (2020): 1397-1418.
[14] Al-Hadhrami, Yahya, and Farookh Khadeer Hussain. "Real time dataset generation framework for intrusion detection systems in IoT." Future Generation Computer Systems 108 (2020): 414-423.
[15] Pradeep Mohan Kumar, K., et al. "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks." Concurrency and Computation: Practice and Experience 33.3 (2021): e5242.
[16] Hossain, Md Delwar, et al. "LSTM-based intrusion detection system for in-vehicle can bus communications." IEEE Access 8 (2020): 185489-185502.
[17] Elshrkawey, Mohamed, Marwa Alalfi, and Hassan Al-Mahdi. "An enhanced intrusion detection system based on multi-layer feature reduction for probe and dos attacks." Journal of Internet Services and Information Security (JISIS) 11.4 (2021): 40-57.
[18] Kasim, Ömer. "A Robust DNS flood attack detection with a hybrid deeper learning model." Computers and Electrical Engineering 100 (2022): 107883.
[19] Lyu, Minzhao, et al. "Hierarchical anomaly-based detection of distributed DNS attacks on enterprise networks." IEEE Transactions on Network and Service Management 18.1 (2021): 1031-1048.
[20] Ishikura, Naotake, et al. "DNS tunneling detection by cache-property-aware features." IEEE Transactions on Network and Service Management 18.2 (2021): 1203-1217.
[21] Zebin, Tahmina, Shahadate Rezvy, and Yuan Luo. "An explainable ai-based intrusion detection system for dns over https (doh) attacks." IEEE Transactions on Information Forensics and Security 17 (2022): 2339-2349.
[22] Kumar, Randhir, et al. "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network." Journal of Parallel and Distributed Computing 164 (2022): 55-68.
[23] Ravi, Vinayakumar, et al. "Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning." IEEE transactions on engineering management 70.1 (2021): 249-266.
[24] Almazini, Hussein, and Ku Ku-Mahamud. "Grey wolf optimization parameter control for feature selection in anomaly detection." International Journal of Intelligent Engineering and Systems 14.2 (2021): 474-483.
[25] Davahli, Azam, Mahboubeh Shamsi, and Golnoush Abaei. "Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks." Journal of Ambient Intelligence and Humanized Computing 11 (2020): 5581-5609.
[26] Manimurugan, S., et al. "Effective attack detection in internet of medical things smart environment using a deep belief neural network." IEEE Access 8 (2020): 77396-77404.
[27] Kittidachanan, Kittikun, et al. "Anomaly detection based on GS-OCSVM classification." 2020 12th International Conference on Knowledge and Smart Technology (KST). IEEE, 2020.
[28] Dwivedi, Shubhra, Manu Vardhan, and Sarsij Tripathi. "Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection." Cluster Computing (2021): 1-20.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: Jul 29, 2024
Accepted: Nov 7, 2024