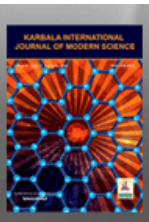

Karbala International Journal of Modern Science

Volume 7 | Issue 4 Article 4

Detecting Malicious DNS Queries Over Encrypted Tunnels Using Statistical Analysis and Bi-Directional Recurrent Neural Networks

Mohammad Al-Fawa'reh
Yarmouk University, School of Information Technology and Computer Science, Jordan, fawareh@yu.edu.jo

Zain Ashi
Princess Sumaya University for Technology ,King Hussein School for Computing Sciences, Jordan

Mousa Tayseer Jafar
Princess Sumaya University for Technology ,King Hussein School for Computing Sciences, Jordan

Detecting Malicious DNS Queries Over Encrypted Tunnels Using Statistical Analysis and Bi-Directional Recurrent Neural Networks

Abstract

The exponential rise in the number of malicious threats targeting computer networks and digital services puts network infrastructure in jeopardy. Domain name protocol attacks are one of the most pervasive network attacks posing a threat to networks, whereby attackers send harmful information to the network; this type of threat is identified as DNS tunneling. The DNS protocol has recently gained increased attention from cyber-attackers, targeting organizations with a web presence or reliance on e-commerce businesses. Cyber-attackers can subtly exploit the contents of encrypted DNS packets that are sent across covert network tunnels, which are difficult for firewalls and blacklist detection methods to detect. Therefore, efficient methods for detecting DNS intrusions in the network are required. Machine learning (ML), deep learning (DL), and computational intelligence models have proved to be increasingly effective in dealing with these cyber-attacks, especially when using an appropriate dataset. This paper proposes an intrusion detection model to detect malicious DNS over HTTPS (DoH) queries among network covert tunnels, using statistical analysis and Bi-directional Recurrent Neural Network (BRNN) techniques, based on the flow level of the network traffic. The proposed approach was tested and evaluated based on a realistic dataset called CIRA-CIC-DoHBrw-2020, provided by the Canadian Institute for Cybersecurity. Experiments have shown that the robustness of the model is strong, with a detection rate of 100%. Furthermore, the proposed model achieved high performance in terms of the accuracy rate in detecting malicious DoH queries, with low false-negative and false-positive rates. Furthermore, the number of features used is fewer than other approaches, making it perform faster in the training and testing phases.

Keywords

Intrusion Detection, DNS Tunneling, Network Security, Threat Detection, Anomaly Detection, DNS attacks

Cover Page Footnote

Acknowledgment: We are very grateful to the reviewers for their valuable comments that helped improve the paper. We wish to express our gratitude to all members of our colleges, Yarmouk University and Princess Sumaya University for Technology (PSUT), for their support.

1. Introduction

DNS protocol has been the core of the Internet infrastructure since 1985. It is a query-response protocol to help users reach host servers by resolving hostnames and corresponding IP addresses Bilge et al. [1]. DNS provides services through the Internet depending on domain names and their IP addresses, such as email services, load balancing, and distribution networks, etc. Torabi et al. [2]. DNS traffic flows unencrypted through the organization's network, which enables the organization's administrators to monitor how insiders browse websites and broadcast media consumption, and monitor network traffic against any malicious activities Ashi et al. [3]. This seems beneficial, but at the same time it is against the principle of protecting user privacy, which is always an essential consideration. Additionally, transmitting unencrypted traffic is a vulnerability that can be exploited by cyber-attackers, compromising the confidentiality, integrity, and availability of such data.

Popularizing protocols to encrypt DNS traffic is the best solution to improve the confidentiality and integrity of user privacy. Such protocols include DNS over HTTPS (DoH) and DNS-over-TLS (DoT), which are double-edged swords; while they protect user privacy, they create challenges for organizations trying to protect their environments from outsider malware or malicious insiders Stevanovic et al. [4]. These protocols have made monitoring the network traffic more complex, and have facilitated the malicious activities of attackers by hiding their malicious malware within DoH queries, to impose dominance on command-and-control servers and thereby launch attacks Stevanovic et al. [4].

The “IDC 2020 Global DNS Threat Report” announced that in 2020 DNS attacks were launched against 83% of the Internet service providers, and 79% of all telecommunications providers, affecting the availability of their services and causing losses of over USD 5 million in losses.

The more creative cyber attackers become, the more efforts researchers in the field must undertake. In recent times, detecting DNS attacks has become the primary focus of many researchers, and awareness has increased that the traditional blacklisted intrusion detection methods are no longer working effectively. Although blacklisted systems have low false detection alarms, they are unable to detect zero-day attacks Jafar et al. [5]. Furthermore, keeping them up-to-date with the evolution of DNS attacks is incredibly expensive Satoh et al. [6].

Therefore, increasing attention has been redirected towards careful network traffic analysis, to improve accuracy and effectiveness in distinguishing between malicious and benign DoH queries Almashhadani et al. [7]. Related approaches extract the intelligence of Big Data, computational intelligence, and ML technologies, all of which can build Intrusion Detection Systems (IDS) that provide organizational capabilities to monitor networks and prevent violations of user privacy Torabi et al. [2]. Moreover, researchers have attempted to provide more development by looking for new trends, including adding a preprocessing layer to extract the most significant features in the dataset, and combining multiple classifiers to improve performance Kailas et al. [8].

Despite that, IDS to detect malicious DoH queries based on conventional ML and Big Data technologies has challenging issues, like processing large volumes of network traffic promptly and accurately, and the ability to generalize training predictions for new and different types of attacks Yang et al. [9].

With the revolutionary growth of computational power, DL techniques have demonstrated remarkable progress and have been able to overcome many challenges and limitations Roy and Cheung [10]. DL neural networks - such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) - have delivered progress in a wide range of applications, including self-driving vehicles, voice/image recognition, and traffic prediction Rusk [11]. Moreover, new algorithms recently introduced for CNN and RNN can extract a sequential connection between features, to solve more difficult issues that were traditionally addressed by less efficient conventional methods Sherstinsky [12].

Additionally, the importance of statistical feature analysis cannot be overlooked, as it has been observed experimentally that these features can analyze network traffic flow while removing any redundant information, thereby providing an effective classification of benign and malicious DoH queries Moustafa et al. [13].

These accumulated challenges and the available new DL trends highlight the motivation for this paper, which proposes an IDS model to detect malicious DoH queries in covert tunnels by conducting statistical analysis on the network traffic flow level using BRNN

https://doi.org/10.33640/2405-609X.3155

model. This paper looks for answers to the research question of how IDS model performance can be enhanced in terms of accuracy and error rates.

This paper evaluates the performance of the BRNN model based on the accuracy and error rates to build IDS. For future work, we aim to optimize the model to be implemented in real-time as Intrusion Prevention System (IPS) to prevent any malicious DoH queries within an organizational network.

This paper's structure is as follows: Section 2 reviews previous work in this area, while some background information is explained in Section 3. Section 4 explains the CIRA–CIC–DoHBrw-2020 dataset, and the methodology is described in Section 5. Finally, Section 6 concludes this paper.

2. Literature review

The misuse of DoH queries poses an escalating threat to the Internet's infrastructure by paving the way for attackers to launch their attacks. A variety of IDS have been implemented over the past decade to eliminate this threat, depending on the trend of ML, DL, and data mining technologies. In order to achieve the desired goals, researchers are trying to maintain the highest accuracy rates in detecting malicious DoH queries. Therefore, researchers have increasingly focused on improving the performance of supervised and unsupervised ML techniques by increasing interest in manipulating the dataset, trying to optimize features' dimensions, and selecting the most appropriate solutions to achieve the desired outcomes. To categorize these features, four groups can be used:

1. Relational: relational features are related to predefined malware, predefined malicious IP addresses, and zone/location of the malicious domain names and IP addresses. Defining the relation provides intuition to classify and cluster the potential malicious IP addresses and domain names Torabi et al. [2].
2. Statistical: statistical features are calculated by performing statistical data analysis to distinguish between malicious and benign network traffic without any redundant information Moustafa et al. [13]. These include analyzing the frequencies of the packets in particular network traffic or analyzing the TTL values related to domain activity to identify that domain Torabi et al. [2].
3. Text-based: text-based features can identify the domain name according to its meaning, the randomness degree of its alphabets, and alphabets' similarities. These features are useful to identify patterns of malicious domain names which are generated automatically by a Domain Name Generating Algorithm (DGA) Almashhadani et al. [7].
4. Time-series: time-series features represent the nature of the network traffic, in which TLS protocol encrypts the traffic in a series of packets flowing in a time-periodic manner Montazeri Shatoori et al. [14]. This provides the ability to monitor packets' behavior during a period of time and detect anomalous behavior. An example of a time-series feature is request/response time Al-Fawa'reh and Al-Fayoumiy [15].

Numerous DNS intrusion detections techniques have been explored in the literature in relation to these feature types. Almashhadani et al. [16] called their detection system "MaldomDetector", to detect communications to Command and Control (C&C) servers using malicious DGA-based domain names. The "MaldomDetector" system used the RMA algorithm to select text-based features responsible to determine if there is a random relationship between the characters of the tested domain names in the "DGArchive" dataset. They relied on the randomness of characters, as attackers used random algorithms to generate malicious domain names. They trained and evaluate their detection system with Decision Tree (DT), Support Vector Machine (SVM), Ensemble, Naïve Bayes (NB), and K-Nearest Neighbor (K-NN) ($k = 5$) classifiers. "MaldomDetector" achieved 98% accuracy rate in detecting malicious domain names by depending on the relation between the name characters.

Satoh et al. [6] built their blacklist-based approach to detect the malicious DNS queries by classifying them according to their causes. They noted that administrators cannot rely on ordinary blacklist-based approaches, as their results are mostly unreliable. Their detection approach depended on numerical analysis of benign DNS queries before and after malicious queries, to estimate their causes. Using the blacklist concept to reduce the number of investigated malicious queries, they utilized ML techniques to consistently update their dependent blacklists and increase the reliability of results for the benefit of administrators.

To improve the detection capability of this method, several studies have attempted to automatically update blacklist entries by using ML techniques. They have adopted Word2Vec and several Gaussian ML models for the numerical analyses of 388 entries in their blacklist.

Nguyen et al. [17] suggested a new approach to detect malicious DNS queries, combining two unsupervised ML algorithms: "Density-Based Spatial Clustering of Applications of Noise Clustering" (DBSC) and K-NN. Their approach was based on behavioral analysis of DNS queries, such as the number of exchanged bytes and the time series of the sent messages. They achieved 100% detection accuracy rate, and a 98% ROC-AUC score.

Liu and Gou [18] proposed a DNS detection model depending on load-based and traffic statistical analysis of DNS tunnels to extract the most significant features. For classification, they used the Random Forest (RF) classifier. For evaluation, they used the ROC-AUC score, and they obtained average accuracy of 99.6%.

The efficiency of IDS is enhanced when appropriate datasets and corresponding ML techniques are employed. However, researchers continually attempt to provide ever-more improved solutions. Recently, many studies have incorporated additional processing for dataset features to improve their act during ML model training.

Al Messabi et al. [19] suggested an approach based on the behavioral analysis of suspicious DNS, with marking prior to user visits. Depending on the previous approaches that they had studied, they chose text-based features such as the number of characters, dots, digits, and hyphens in the domain name, the order of the words, and other textual characteristics. They then sorted these features according to their rank rate (using the WEKA tool), and excluded all with a low-ranking rate. They collected their data from malicious malware websites, Google search engines, and Alexa. Their model was trained by the Decision Tree classifier, and achieved 77.52% accuracy.

Almashhadani et al. [7] discussed new cyber-attacks in which attackers target business infrastructures for extortion. One of these attacks attempts to encrypt the victims' data and block their computers, utilizing C&C communication protocols such as DNS, HTTP, and NBNS, to inject their encryption public key. The researchers followed a methodology to analyze the network traffic in both packet and flow levels using the "MCFP" dataset, and built their IDS using independent classifiers. While analyzing the DNS traffic, they found that informative features can be extracted by error statistics in domain names, as well as by differences between illegitimate and legitimate traffic time. Behavioral features can be extracted from a number of DNS queries for meaningless domain names, employing Shannon Entropy metrics, programmer-defined Python functions, and Weka tools. All extracted features were used to train two classifiers, one based on packet level and the other on flow level. Both classifiers built two accurate IDS, with accuracy rates of 97.92% and 97.08%, respectively.

Alsaadi et al. [20] built IDS by implementing Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms to extract features, and three classification algorithms: K-NN, SVM, and NB. Their datasets were KDD cup and NSL-KDD, which contain several network attacks collected at the packet level, such as Denial of Service, Probe, User to Root, and Remote to Local. The accuracy rate achieved by the K-NN to detect those attacks in both datasets on average was 99.53%, while the NB achieved 91%, and SVM achieved 85.6%.

Damodaram [21] proposed a framework to identify malicious botnets according to their DNS queries. They used a genetic algorithm to extract the appropriate features and hyper-parameters as a pre-processing step for the TI-2016 DNS dataset. RF was used to train the framework, which achieved 94.71% accuracy.

Many other researchers have adopted DL techniques to seek further improvement in the IDS domain. DL implements the ML process using a hierarchal artificial neural network (NN), including CNN and RNN. The main characteristic of such techniques is generalization, and the ability to recognize new patterns of malicious NN traffic on which the model was not trained Lecun [22].

The IDS model presented by Fawcett [23] identified malicious network traffic using one class dataset, whereby the whole dataset had only normal traffic data to avoid manual labeling. They used CNN and support vectors (SV) as unsupervised DL techniques and evaluated their IDS model by comparing the CNN and SV performance, finding that SV had the highest accuracy rate of 96% in detecting malicious network traffic.

Kwon et al. [24], introduced an IDS model for an "IEEE based power system network" using BRNN. They analyzed the network traffic of the IEEE cyber-physical system and separated the header and payload for each packet. The BRNN-model training phase was performed. First, they used the packet header data to learn the pattern of the NN communication flow, then they used the packet payload to validate the power system data and detect anomalies. The proposed model could fully detect advanced power system attacks by unauthorized commands and identify NN communication problems.

Yuan et al. [25] proposed an approach to detect DDoS attacks in network traffic using the BRNN technique. In the data preprocessing phase, they extracted text, Boolean, and numerical features. In the

experiment, they implemented one conventional ML technique, RF, and many DL techniques, including CNN and the BRNN with different window sizes. Comparative evaluation between all the implemented techniques revealed that BRNN outperformed all other techniques in terms of generalization and error rate reduction, with an accuracy rate of 98.410%.

Bouzar-Benlabiod et al. [26] built a model to detect NN attacks using Principle Component Analysis (PCA) for feature dimensionality reduction, and LSTM-RNN for training the model. After the evaluation, the achieved accuracy rate was 98.85%, with a low FP rate (4.86%).

Wang et al. [27] attempted to provide enhanced algorithms for building efficient IDS by optimizing the "single-layer hidden forward neural network" and "extreme machine learning" algorithms, to increase learning speed and performance efficiency in classification models. They proposed a "self-adaptive extreme learning machine (SaELM)" algorithm, which adapts itself automatically by selecting the adequate number of neurons in the hidden layer to build the optimal neural network. To evaluate the proposed SaELM they compared its performance against the Italian wine and Iris classification problems, with the performance of BP, ELM, and "general regression neural network (GRNN)" algorithms. SaELM was effective in speed training and automatically finding the optimal number of neurons without adjusting the weight in the training phase.

Cui et al. [28] built their malware detection approach based on an innovative method by utilizing the ability of deep learning in image recognition. They stated that malware code poses serious threats against Internet infrastructure, entailing significant risks in IoT and mobility devices. Their approach, based on image visualization techniques, converted malware codes into greyscale images to extract features automatically. CNN is advantageous in such situations as its structure reduces neural network complexity, especially with greyscale images as multidimensional input. They used CNN to extract the malware images features and classify them. They obtained 25 types of malwares represented by 9342 greyscale images for their dataset. They used a bat resampling algorithm to overcome imbalanced data and achieve higher accuracy rates. Comparing their experimental results with those of previous relevant researches, they achieved a higher accuracy rate of 94.5%, a precision score of 94.6%, and a recall score of 94.5%. Despite the promising results, the CNN algorithm is only used with fixed-size images, and this was a limitation of the proposed approach.

In 1980 a mathematical and analytical method called the wavelet transformation theory was proposed. Wavelet transformation theory proves the ability in time domain localization and focal features. Wavelet Neural network (WNN) combines the advantages of the wavelet algorithm and the robustness of the neural networks to solve classification problems, but it is hampered in finding a suitable activation function. Wang et al. [29] proposed a threat assessment model, considering it the mainstay to prevent cooperation attacks. They developed an algorithm that can be leveraged to determine the optimal activation function of the WNN, which they called "Multiple Wavelet Function Wavelet Neural Networks". The results showed that "Morlet" is the optimal activation function that outfits the proposed problem.

"Probabilistic Neural Network PNN" was used by Yi et al. [30] to diagnose faults in the mechanical equipment. Their contribution was derived from striving to exploring more models with accurate and real-time predictions at low computational costs. They proposed a self-adaptive PNN (SaPNN), which automatically supplied the original PNN with the most suitable parameters. To prove the performance of SaPNN when implemented with a fault detection problem, it was compared to the performance of BP, ELM, GRNN, and SaELM algorithms. Experimental results proved that SaPNN achieved higher accuracy rates with lower computational costs and better generalization.

3. Background

A. IDS vs. Prevention Systems

Confidentiality, integrity, and availability (CIA) constitute the security objectives triangle in any network, and any measure to penetrate this triangle is called an intrusion, such as viruses, worms, and attacks. Network intrusions are a primary concern for cyber defenders, who are always trying to create systems to detect and prevent these intrusions Ashoor and Gore [31], as explored in this research. IDS identifies intrusions by monitoring network traffic, preserving network logs, and reporting intrusions to the network administrator IDS. IDS copies network traffic for read-only analysis to detect any intrusions that have already been launched, and notify the administrators about what is going on so they can apply manual interventions. IDS is implemented outbound of the network line without affecting the network data flow Ashoor and Gore [32].

IDS plays a critical role in network security, but after-occur responses and the manual interactions

delays, which range from a few seconds to minutes Ashoor and Gore [32], may enable intrusion attacks to succeed. With the increasing development of network intrusions, a small worm can exploit a vulnerable server within 30 s Ashoor and Gore [31].

Therefore, IPS automates responses to detected intrusions to prevent them before any negative consequences, by blocking, dropping, or terminating their network traffic Wu et al. [33]. To do this, IPS is implemented within the network line (as shown in Fig. 1), so high processing and storage capabilities are required to cope with the high network traffic volume without any delay Ashoor and Gore [31].

B. DNS Protocol Potential Risks

DNS infrastructure is a popular target for cyber-attacks, targeting organizations that have a web presence or which rely on e-commerce. Attackers can abuse DNS infrastructure to collect users' identifying credentials, such as IDs and passwords. For example, DDoS attacks against DNS servers can be executed by numerous compromised devices (as in botnet attacks), to overwhelm these servers with bogus queries until they become unable to respond to legitimate queries Hassan [34].

Two main techniques help to constitute sophisticated DNS attacks:

- IP-Flux/Fast-Flux Technique

  Repeatedly random and short time-to-live (TTL) IP addresses are assigned to a specific domain name by "Domain Change Algorithm" Stevanovic et al. [4]. Every 3 min, the DNS server can generate a different IP address for a specific domain name to ensure service continuity, and to support more traffic volume for that domain name Cochran and Cannady [35]. Attackers abuse the IP-Flux technique to distribute their attacks across the wide networks, including DDOS, phishing, data theft, and other attacks Stevanovic et al. [4]. Taking advantage of changeable and short TLL IP addresses, attackers can avoid any IDS and hide the existence of their malicious servers Torabi et al. [2]. An attacker can create a hidden layer between the compromised user and the desired web server.

  In normal situations, the user gets the IP address in response to a DNS query for the requested website, but when an attacker appears in the middle, the user will get an IP address for a malicious website Cochran and Cannady [35]. For example, when a user sends a DNS query requesting the IP address of his bank website to a compromised DNS server, he will get back an IP address of a malicious website, which looks exactly like the bank web site. When the user enters his user's name and password, he gets an error message and an instruction to try again; for the retry, the malicious site redirects the user to the real bank website, to continue without any awareness that his username and password have been taken over by the hackers.
- Domain-Flux

  A botnet generates an enormous number of short life domain names using Domain Generation Algorithm (DGA), sending them one after another to Control & Command (C&C) servers, depending on DNS queries to locate one of them. When attackers allocate a C&C server, they open a secret tunnel to control a botnet through it Torabi et al. [2]. To block and detect these malicious domain names, the covert tunnels between the attacker and the C&C server must be detected before any successful malicious activity occurs Khare et al. [36].

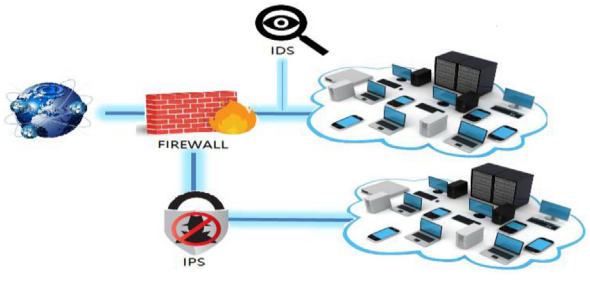

Fig. 1. Intrusion Detection System vs. Intrusion Prevention System.

C. Deep Learning (DL) Techniques

DL neural network is an advanced form of ML that uses hierarchical data representation. Features in the higher level, having more abstractions, are extracted from the lower level, using various activation functions Rusk [11]. The complicity of the hierarchical structure of the DL models can improve the performance of classification and regression problems by increasing the achieved accuracy and reducing error rates Kamilaris and Prenafeta-Boldú [37]. DL neural networks (CNN and RNN) have delivered an artificial revolution in a wide range of applications, such as self-driving vehicles, voice/image recognition, and traffic prediction, etc. Rusk [11].

Supervised DL model manipulates large labeled datasets, just like the dataset of this study is labeled with benign or malicious DoH queries (y). At each time point in the training phase, a training set of features (X_1,X_2,X_3,

$X_4, ..., X_n$) is presented to the DL model with initial weights ($W_1, W_2, W_3, W_4, ..., W_n$). In a hidden layer, predictions ($\hat{y}$) are produced by activation functions. The error rate between the actual (y) and the predicted ($\hat{y}$) values is then calculated to adjust the weights, until the error rate reaches the minimal, as shown in Fig. 2. After the training phase, the model performance is evaluated, to determine its ability to output a reasonable prediction for new inputs Lecun [22].

D. Recurrent Neural Network (RNN)

RNN, one of the models that demonstrated proficiency in DL neural networks, depends on the input sequence connection $\left(x^{(t-1)}, x^{(t)}, x^{(t+1)}\right)$ Zhong et al. [38]. The hidden layer in the RNN consists of sequential activation function nodes $\left(h^{(t-1)}, h^{(t)}, h^{(t+1)}\right)$. The sequence goes in forward or backward propagation operations, whereby the input for the next h depends on the output of the previous h. In other words, the output of the previous h participates in the production of the output of the next h during a time series $\left(O^{(t-1)}, O^{(t)}, O^{(t+1)}\right)$ Zhong et al. [38], as shown in Fig. 3.

Despite the stability of RNN models, they suffer from limitations in keeping long input sequences during elongated time series situations. This problem is called "vanishing or exploding gradients" Sherstinsky [12].

Long Short-Term Memory RNN (LSTM) models are designed to avoid the vanishing gradients problem Al-Emadi et al. [39]. With long time series, LSTM

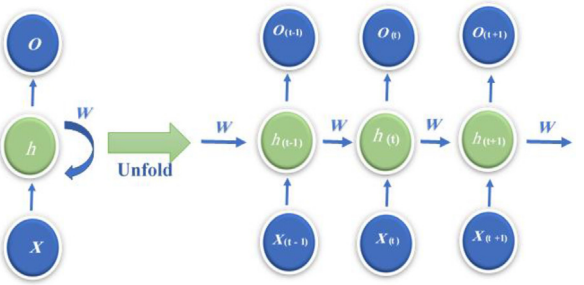

Fig. 3. Forward RNN model.

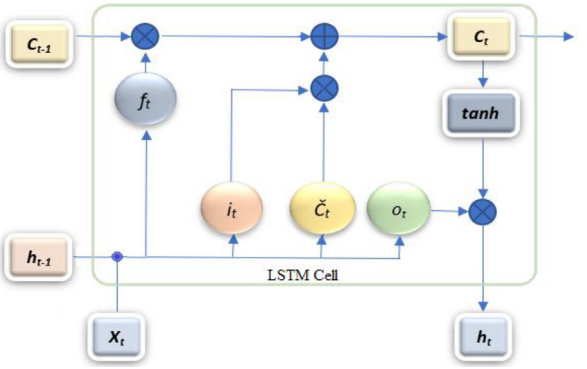

Fig. 4. Inside LSTM cell.

cells are employed in the RNN nodes to learn the long sequential connections of Meng et al. [40]. Besides the activation function, each LSTM cell has three gates Sherstinsky [12]:

1. Input gate (i_t) determines which Xi from the previous cell will remain in the long-term memory.

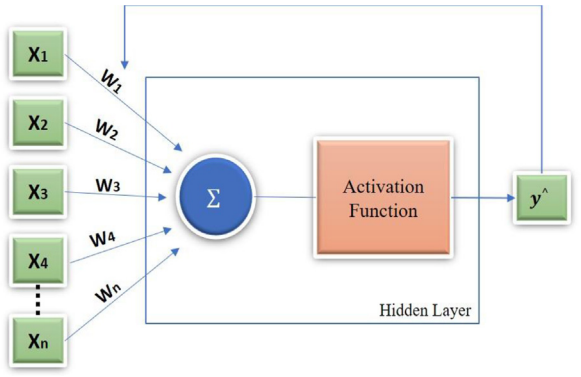

Fig. 2. DL model architecture.

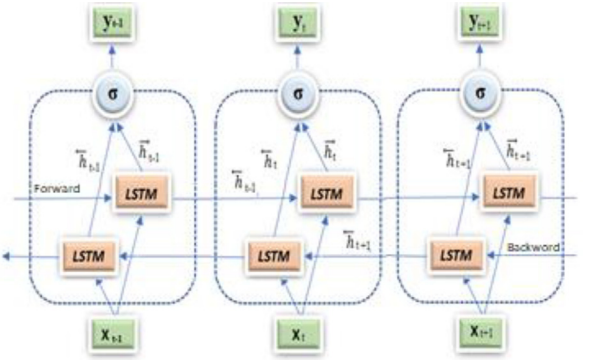

Fig. 5. BRNN architecture.

Equations (1) and (2) calculate the input gate and the cell state, respectively.

2. Forget gate (f_t): determines which X_t from the previous cell will be forgotten from the long-term memory. Equations (3) and (4) calculate the forget gate and the cell state, respectively.
3. Output gate (o_t): determines which outputs will be passed to the next LSTM cell. Equations (5) and (6) calculate the output gate and the vector of the hidden sequence, respectively.

Fig. 4 shows the LSTM cell in each RNN node.

$$i_t = \sigma(w_i.[h_{t-1}, X_t] + b_i) \quad (1)$$

$$\check{C}_t = tanh(w_c.[h_{t-1}, X_t] + b_c) \quad (2)$$

$$f_t = \sigma\left(w_f.[h_{t-1}, X_t] + b_f\right) \quad (3)$$

$$C_t = f_t.C_{t-1} + i_t.\check{C}_t \quad (4)$$

$$O_t = \sigma(W_o.[h_{t-1}, X_t] + b_o) \quad (5)$$

$$h_t = O_t.tanh(C_t) \quad (6)$$

where b is bias, C_t, $\check{C}$ are the memory new state, and h_t is the sequence of hidden vectors.

E. Bi-Directional RNN (BRNN)

BRNN models employ two layers of LSTM cells in a hidden layer, to be able to process the sequential input in forward and backward directions. This reinforces the BRNN to build its predictions in a Bi-directional way Kwon et al. [24]. In Fig. 5, ($\overleftarrow{h_t}$) is the backward output sequence, ($\overrightarrow{h_t}$) is the forward output sequence, and the output Y is calculated by combining both ($\overleftarrow{h_t}$) and ($\overrightarrow{h_t}$) as in equation (7).

$$Y = \sigma\left(\overleftarrow{h_t}, \overrightarrow{h_t}\right) \quad (7)$$

where σ is concatenation, multiplication, sum, or average.

4. CIRA–CIC–DoHBrw-2020 dataset

DNS protocols have several vulnerabilities that have been continuously exploited. DNS misuses have always attracted researchers looking for the best protection methods to preserve the privacy of DNS queries across the network. HTTPS protocol encrypts DNS queries and associates a secret tunnel for their transmission between the client and the DNS server, to protect them from potential security attacks and protect users' privacy. It is impossible for firewalls to discover these tunnels, but attackers can utilize them to send many malicious DNS queries in the same connection, reducing the potentiality to detect them MontazeriShatoori et al. [14].

CIRA–CIC–DoHBrw-2020 dataset was created due to the importance of the dataset quality to enhance the performance of ML techniques MontazeriShatoori et al. [14]. It was subsequently verified by the Canadian Institute for Cybersecurity (CIC). CIRA–CIC–DoHBrw-2020 analyzes DoH traffic within an application to distinguish benign from malicious DoH queries Banadaki [41]. In this study, after the network traffic was generated, all its packets were filtered and aggregated to get rid of the impurities and to represent the collected data succinctly.

Fig. 6 illustrates the process used to capture simulated network traffic, in which:

- 10,000 Alexa websites were accessed to generate HTTPS (non-DoH-related) traffic.
- Mozilla Firefox and Google Chrome web browsers were used for benign DoH traffic.
- DNS tunneling tools were utilized to generate malicious DoH traffic, creating covert tunnels to send encapsulated DNS queries using encrypted HTTPS requests to DoH servers Banadaki [41].

Two-layer systematic approach was implemented to capture malicious and benign network DoH traffic. At layer one the traffic was classified into DoH-related traffic and non-DoH-related traffic, using statistical features. At layer two, the collected data was used to train time-series classifier to label DoH-related traffic into benign and malicious DoH, as shown in Fig. 6. Table 1 shows more detailed information about the data capturing process.

Python programmer defined tools were created for the following purposes:

- "DoH Data Collector" tool was used to capture the HTTPS traffic.
- "DoHLyzer" tool was used to read pcap files (network packets), and to generate and then analyze the flow of DoH-related traffic.
- "DoHMeter" tool was used to produce the output csv file with labelled data, by extracting 28 features (statistical and time-series) from pcap files on the flow level, as shown in Table 2.

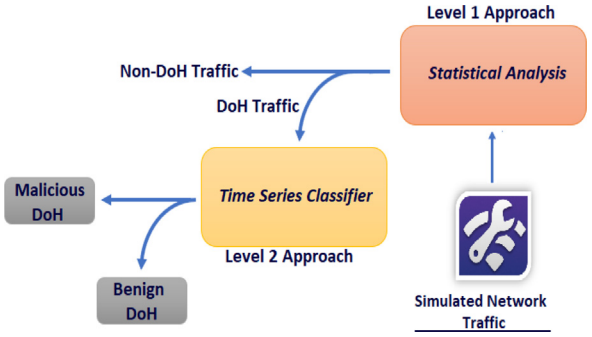

Fig. 6. The proposed two layer approach.

5. Methodology

This section demonstrates the implemented methodology (shown in Fig. 7) to build the proposed BRNN-IDS model, which distinguishes between benign and malicious DoH queries, using Python 3 programming utilities and libraries (Skilearn, Pandas, Keras, etc.).

A. Data Selection Phase

Extracting the statistical features on the network flow level, CIRA–CIC–DoHBrw-2020 Dataset's pcap files were used. 34 features were collected in csv files, to be used by the ML learning classifiers. In this implemented methodology, only statistical features were used, therefore the six non-statistical features were excluded. Table 2 shows a sample of features.

B. Data Pre-processing Phase

Choosing the best statistical features to improve the performance of the classifier is not sufficient if the dataset is not properly prepared. In this phase, the dataset was pre-processed by:

Table 2
Sample of features.

Description of features	
F1	# of FBS[a]
F2	Rate of FBS
F3	# of FBR[a]
F4	Rate of FBR
F5	Mn[b] pkt len
F6	Mdn[b] pKt len
F7	Md[b] pkt len
F8	Variance of pkt len
F9	STD[b] of pkt len
F10	Coefficient of pkt len
F11	Skew of Mdn pkt len
F12	Skew of Md pkt len
F13	Mn pkt time
F14	Mdn pkt time
F15	Md pkt time
F16	Variance of pkt time
F17	STD pkt time
F18	Coefficient of pkt time
F19	Skew of Mdn pkt time
F20	Skew of Md pkt time
F21	Mn Req/Res time
F22	Mdn Req/Res time
F23	Md Req/Res time
F24	Variance of Req/Res time
F25	STD of Req/Res time
F26	Coefficient of Req/Res time
F27	Skew of Mdn Req/Res time
F28	Skew of Md Req/Res time

STD: Standard deviation.
[a] FBS: Flow bytes sent –FBR: Flow bytes received.
[b] Mn: Mean, Mdn: Median, Md: Mode.

1. Deleting All Redundant Values and Treating Missing Values.

   This is to reduce the computational overhead and the training time; null or missing values could be replaced with the median of other corresponding values or zero, or easily be deleted with their entire row. In this methodology the redundant and the missing values were deleted.
2. Normalizing

   All the values were normalized within a specific

Table 1
Data capturing process.

SERVERS→	AD Guard		Cloudflare		Google DNS		Quad9	
	Packets	Flows	Packets	Flows	Packets	Flows	Packets	Flows
HTTPS (DNS-unrelated-Benign DNS queries)								
Google Chrome	5609 K	105,141	6117 K	132,552	58,578 K	108,680	10,737 K	199,090
Mozilla Firefox	4943 K	50,485	4299 K	90,260	6413 K	138,422	4956 K	92,670
Malicious DNS queries								
DNS to TCP	128 K	5459	3694 K	6045	28,711 K	17,423	8750 K	138,588
DNSCAT2	1301 K	5369	12,346 K	9230	48,069 K	11,915	9309 K	9108
LODINE	3938 K	11,336	5932 K	14,110	73,459 K	12,192	22,668 K	8975

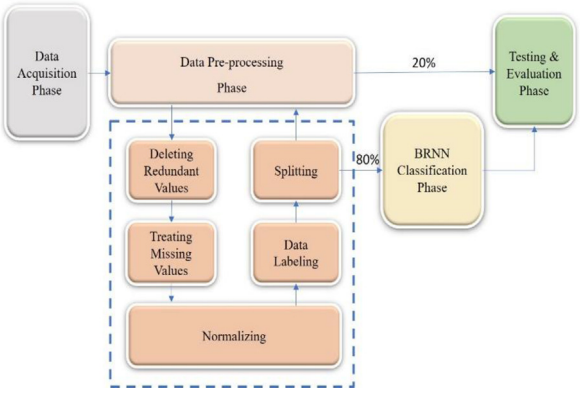

Fig. 7. The proposed methodology.

range, to improve the classification algorithms' calculations and to avoid any overflow due to particularly high- or low-ranging (non-normal) values Al-Fawa'reh and Al-Fayoumiy [15].

3. Data Labeling

   All the values of benign and malicious DNS queries in the label column were represented by 0 and 1 values, respectively.

4. Splitting

   The data set was split: 80% to the training set, and 20% to the testing set.

C. BRNN Classification Phase

The BRNN model is built so as to concatenate the output of bi-directional LSTM layers. The Sigmoid function (σ) shown in equation (8) was selected to be the output activation function, which indicates the final prediction ($\hat{y}$):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

The BRRN model is used in two phases:

1. The BRNN training phase, in which the BRNN model learns how to identify benign and malicious DoH queries patterns, utilizing the sequence of statistical features in the 80% training set.
2. The BRNN testing phase, in which the BRNN model is fed with new inputs from the 20% testing set, to test its generalization ability.

As demonstrated in Fig. 8 the selected BRNN consists of three layers of hidden units, made up of 64 units for the first hidden layer and 128 for the second and third hidden layers. The activation function on the first layer is tanh, where ReLU in the second layer, and sigmoid activation function for the last output layer. The architecture was implemented using Keras v2.2.4 and TensorFlow.

D. Testing and Evaluation Phase

For evaluation, the BRNN training and testing phases were compared in terms of their performance accuracy and error rates, as shown in Figs. 9 and 10. The error rate between them was undetected, and the performance of the BRNN model displayed a rapid improvement after a few iterations of training. The accuracy of both training and testing phases was completely achieved.

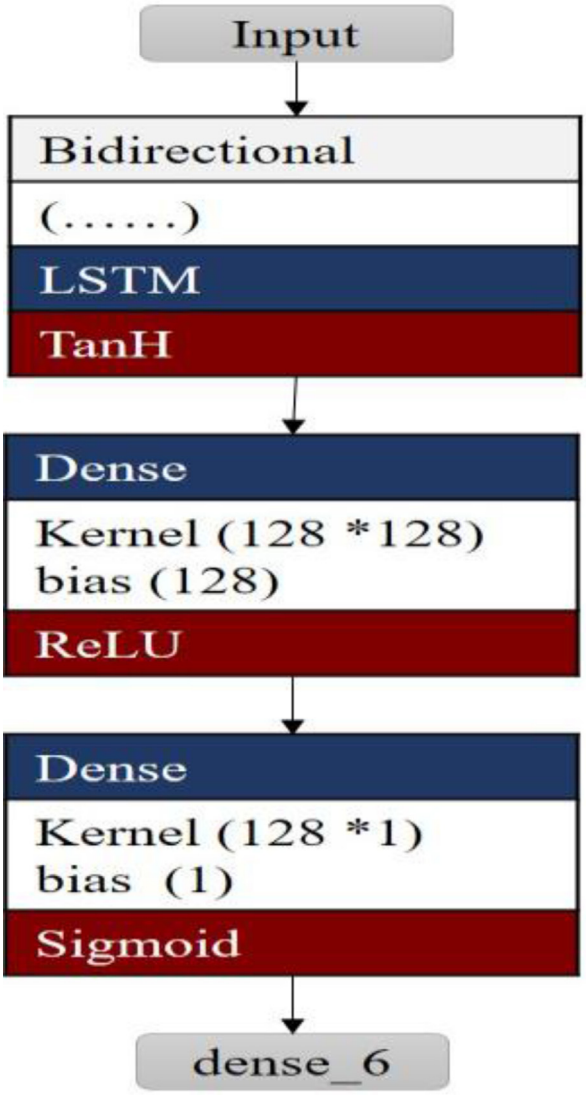

Fig. 8. BRRN model.

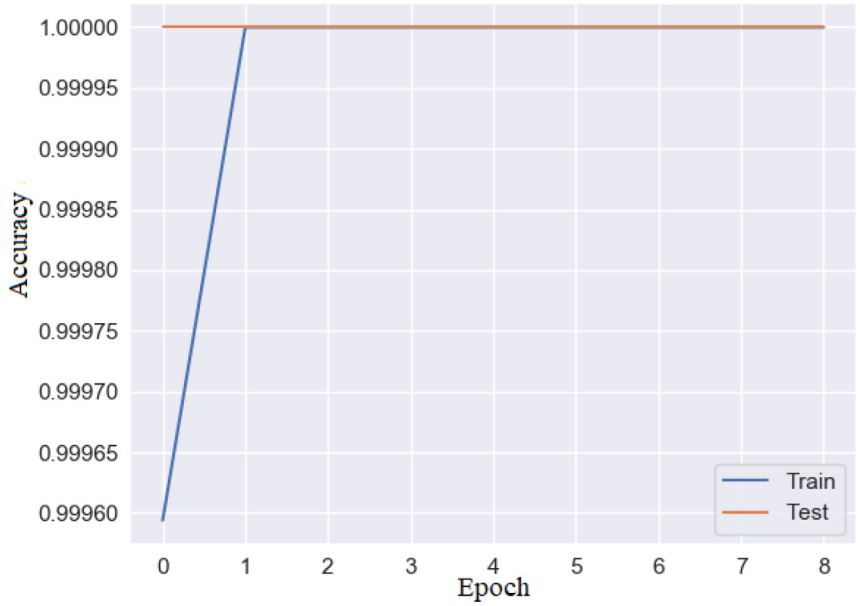

Fig. 9. BRNN model accuracy rate.

The achieved accuracy rate for the BRNN model was 100%. In comparison with previous studies (as shown in Table 3), the proposed BRNN model can distinguish between benign and malicious DoH queries more efficiently, without any FP and FN rates. In conclusion, the proposed BRNN model is adequate to build DoH IDS that organizations can rely on to protect their environment.

As shown in Table 3, most ML studies used old datasets to test their work, whereas only two recent studies have used realistic network traffic to test their systems. Experiments have shown that our model achieves the state of arts using only statistical features.

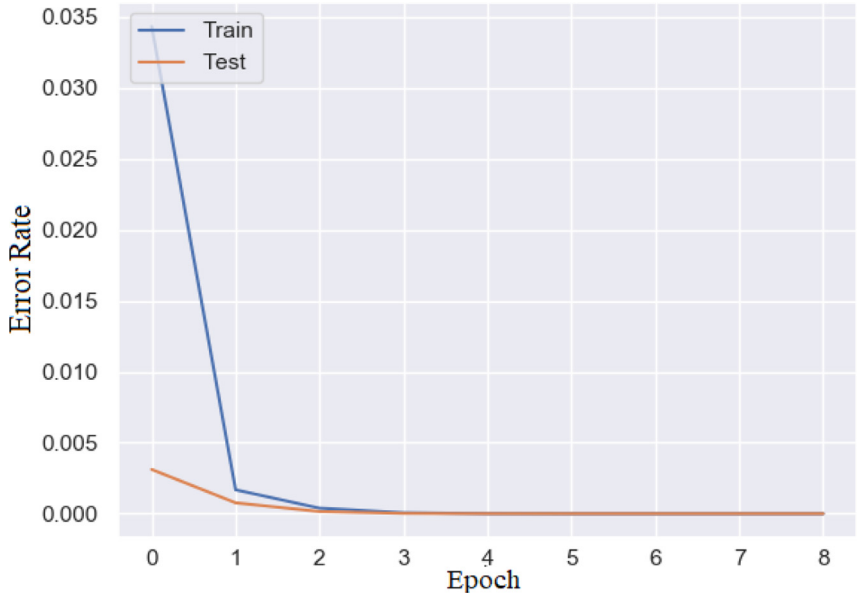

Fig. 10. BRNN model error rate.

Table 3
The achieved accuracy.

Study	Feature extraction algorithm	Classification algorithms		Highest achieved accuracy
		ML	DL	
Jafar et al. [5]	✓	RF, SVM, DT, K-NN	×	99.9%
Almashhadani et al. [16]	×	DT, SVM, NB, K-NN	×	98%
Nguyen et al. [17]	×	DBSC, K-NN	×	100%
Al Messabi et al. [19]	×	DT	×	77.52%
Alsaadi et al. [20]	✓	K-NN, SVM and NB	×	99.53%
Damodaram [21]	✓	RF	×	94.71%
Fawcett [23]	×	×	CNN, SV	96%
Yuan et al. [25]	×	×	BRNN	98.410%
Bouzar-Benlabiod et al. [26]	×	×	LSTM-RNN	98.85%
Banadaki [41]	✓	LGBM and XG Boost	×	100%
This Model	×	×	BRNN	100%

6. Conclusions

DNS abuse is one of the most difficult threats cybersecurity experts seek to combat. However, since many DNS tunneling tools are available online and do not require a high level of skill to use, businesses must be aware of this potential security violation. This paper proposes an IDS model to detect malicious DoH queries among the network's covert tunnels using statistical analysis and DL techniques by applying a bi-directional recurrent neural network model. Comparing the achieved accuracy and error rates with previous studies, the proposed IDS model can distinguish between benign and malicious DoH queries more efficiently. This model can be used to detect any malicious DNS queries or DNS tunneling type in the real world. One of the most important concerns to consider is the size of the dataset, since the most innovative DL has a large dataset. The overall conclusion is that combining the CIRA–CIC–DoHBrw-2020 dataset with DL techniques is a highly effective method for detecting DNS tunneling, with a distinct impact on internet security.

The only limitation or constraint that was experienced was the size of the dataset, since the state-of-the-art DL has a large dataset. In future work, advanced methods will be applied for hyperparameter optimization and feature selection, such as Monarch Butterfly Optimization (MBO), Earthworm Optimization Algorithm (EWA), Elephant Herding Optimization (EHO), Moth Search (MS) algorithm, Slime Mold Algorithm (SMA), and Harris's Hawk's Optimization (HHO). Furthermore, distributed IDS deployment using different ML and DL methods will be implemented on a real network, to enhance the design and increase its throughput and performance, and to strengthen effort to prevent malicious DoH queries within organizational networks.

Acknowledgment

We are very grateful to the reviewers for their valuable comments that helped improve the paper. We wish to express our gratitude to all members of our colleges, Yarmouk University and Princess Sumaya University for Technology (PSUT), for their support.

References

[1] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, C. Kruegel, Exposure, ACM Trans. Inf. Syst. Secur. 16 (2014) 1–28, https://doi.org/10.1145/2584679.

[2] S. Torabi, A. Boukhtouta, C. Assi, M. Debbabi, Detecting internet abuse by analyzing passive DNS traffic: a survey of implemented systems, IEEE Commun. Surv. Tutorial 20 (2018) 3389–3415, https://doi.org/10.1109/comst.2018.2849614.

[3] Z. Ashi, L. Aburashed, M. Al-Fawa'reh, M. Qasaimeh, Fast and reliable DDoS detection using dimensionality reduction and machine learning, in: 15th International Conference for Internet Technology and Secured Transactions, ICITST, 2020, pp. 1–10, https://doi.org/10.23919/ICITST51030.2020.9351347.

[4] M. Stevanovic, J.M. Pedersen, A. D'Alconzo, S. Ruehrup, A. Berger, On the ground truth problem of malicious DNS traffic analysis, Comput. Secur. 55 (2015) 142–158, https://doi.org/10.1016/j.cose.2015.09.004.

[5] M.T. Jafar, M. Al-Fawa'reh, Z. Al-Hrahsheh, S.T. Jafar, Analysis and Investigation of Malicious DNS Queries Using CIRA-CIC-DoHBrw-2020 Dataset, 2, 2021, https://doi.org/10.1109/access.2021.2908675, 65–70.

[6] A. Satoh, Y. Nakamura, Y. Fukuda, K. Sasai, G. Kitagata, A cause-based classification approach for malicious DNS queries detected through blacklists, IEEE Access 7 (2019)

142991–143001, https://doi.org/10.1109/access.2019.2944203.
[7] A.O. Almashhadani, M. Kaiiali, S. Sezer, P. O'Kane, A multi-classifier network-based crypto ransomware detection system: a case study of locky ransomware, IEEE Access 7 (2019) 47053–47067, https://doi.org/10.1109/access.2019.2907485.
[8] P. Kailas, K. Chandrasekaran, Recursive harmony search based classifier Ensemble reduction, in: Second International Conference on Intelligent Computing and Control Systems, ICICCS, 2018, pp. 1612–1618, https://doi.org/10.1109/ICCONS.2018.8663245.
[9] H. Yang, F. Wang, Wireless network intrusion detection based on improved convolutional neural network, IEEE Access 7 (2019) 64366–64374, https://doi.org/10.1109/access.2019.2917299.
[10] B. Roy, H. Cheung, A deep learning approach for intrusion detection in internet of things using Bi-directional long short-term memory recurrent neural network, in: 28th International Telecommunication Networks and Applications Conference, ITNAC, 2018, pp. 1–6, https://doi.org/10.1109/ATNAC.2018.8615294.
[11] Y. Bengio, A. Courville, Deep learning of representations, in: International Conference on Statistical Language and Speech Processing, 2013, pp. 1–37, https://doi.org/10.1007/978-3-642-36657-4_1.
[12] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Phys. Nonlinear Phenom. 404 (2020) 132306, https://doi.org/10.1016/j.physd.2019.132306.
[13] N. Moustafa, B. Turnbull, K.-K.R. Choo, An Ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet Thing J. 6 (2019) 4815–4830, https://doi.org/10.1109/jiot.2018.2871719.
[14] M. MontazeriShatoori, L. Davidson, G. Kaur, A. Habibi Lashkari, Detection of DoH tunnels using time-series classification of encrypted traffic, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/ PiCom/ CBDCom/CyberSciTech, 2020, pp. 63–70, https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.
[15] M. Al-Fawa'reh, M. Al Fayoumi, Detecting stealth-based attacks in large campus networks, Int. J. Adv. Trends Comput. Sci. Eng. 9 (2020) 4262–4277, https://doi.org/10.30534/ijatcse/2020/15942020.
[16] A.O. Almashhadani, M. Kaiiali, D. Carlin, S. Sezer, MaldomDetector: a system for detecting algorithmically generated domain names with machine learning, Comput. Secur. 93 (2020) 101787, https://doi.org/10.1016/j.cose.2020.101787.
[17] T.Q. Nguyen, R. Laborde, A. Benzekri, B. Qu'hen, Detecting abnormal DNS traffic using unsupervised machine learning, in: 4th Cyber Security in Networking Conference, CSNet, 2020, pp. 1–8, https://doi.org/10.1109/CSNet50428.2020.9265466.
[18] Y. Liu, X. Gou, Research on application of feature analysis method in DNS tunnel detection, J. Phys.: Conf. Ser. 1575 (2020), 012069, https://doi.org/10.1088/1742-6596/1575/1/012069.
[19] K.A. Messabi, M. Aldwairi, A.A. Yousif, A. Thoban, F. Belqasmi, Malware detection using DNS records and domain name features, in: Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 2018, pp. 1–7, https://doi.org/10.1145/3231053.3231082.
[20] H.I. Alsaad, R.M. Almuttairi, O. Bayat, O.N. Ucani, Computational Intelligence Algorithms to Handle Dimensionality Reduction for Enhancing Intrusion Detection System, 36, 2020, https://doi.org/10.6688/JISE.202003_36(2).0009, 293–308.
[21] R. Damodaram, Fake website detection : association classification algorithm with Ant Colony optimization technique, in: International Journal of Advanced Research in Computer Science, 2011, pp. 568–577, https://doi.org/10.1007/978-3-642-343883.
[22] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.
[23] T. Fawcett, ROC graphs with instance-varying costs, Pattern Recogn. Lett. 27 (2006) 882–891, https://doi.org/10.1016/j.patrec.2005.10.012.
[24] S. Kwon, H. Yoo, T. Shon, IEEE 1815.1-based power system security with bidirectional RNN-based network anomalous attack detection for cyber-physical system, IEEE Access 8 (2020) 77572–77586, https://doi.org/10.1109/access.2020.2989770.
[25] X. Yuan, C. Li, X. Li, DeepDefense: identifying DDoS attack via deep learning, in: 2017 IEEE International Conference on Smart Computing, SMARTCOMP, 2017, pp. 1–8, https://doi.org/10.1109/SMARTCOMP.2017.7946998.
[26] L. Bouzar-Benlabiod, S.H. Rubin, K. Belaidi, N.E. Haddar, RNN-VED for reducing false positive alerts in host-based anomaly detection systems, in: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science, IRI, 2020, pp. 17–24, https://doi.org/10.1109/IRI49571.2020.00011.
[27] G.-G. Wang, M. Lu, Y.-Q. Dong, X.-J. Zhao, Self-adaptive extreme learning machine, Neural Comput. Appl. 27 (2016) 291–303, https://doi.org/10.1007/s00521-015-1874-3.
[28] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, J. Chen, Detection of malicious code variants based on deep learning, IEEE Trans. Ind. Inf. 14 (2018) 3187–3196, https://doi.org/10.1109/tii.2018.2822680.
[29] G. Wang, L. Guo, H. Duan, Wavelet neural network using multiple wavelet functions in target threat assessment, Sci. World J. 2013 (2013) 1–7, https://doi.org/10.1155/2013/632437.
[30] J.-H. Yi, J. Wang, G.-G. Wang, Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem, Advances in Mechanical Engineering, in: Proceedings of the International Conference on Applied and Engineering Mathematics, ICAEM, 2016, pp. 1–13, https://doi.org/10.1177/1687814015624832.
[31] A.S. Ashoor, S. Gore, Difference between intrusion detection system (IDS) and intrusion prevention system (IPS), in: Fifth International Conference on Risks and Security of Internet and Systems, CRiSIS, 2011, pp. 497–501, https://doi.org/10.1007/978-3-642-22540-6_48.
[32] A.S. Ashoor, S. Gore, Intrusion detection system (IDS) &intrusion prevention system (IPS), Case Study 2 (2011) 1–3.
[33] Z. Wu, D. Xiao, H. Xu, X. Peng, X. Zhuang, Virtual inline: a technique of combining IDS and IPS together in response intrusion, in: First International Workshop on Education Technology and Computer Science, vol. 1, 2009, pp. 1118–1121, https://doi.org/10.1109/ETCS.2009.255.

[34] A. Hassan, S. Tahir, A.I. Baig, Unsupervised machine learning for malicious network activities, in: International Conference on Applied and Engineering Mathematics, ICAEM, 2019, pp. 151−156, https://doi.org/10.1109/ICAEM.2019.8853788.

[35] T.O. Cochran, J. Cannady, Not so fast flux networks for concealing scam servers, in: 2010 Fifth International Conference on Risks and Security of Internet and Systems, CRiSIS, 2010, pp. 1−8, https://doi.org/10.1109/CRISIS.2010.5764914.

[36] N. Khare, P. Devan, C. Chowdhary, S. Bhattacharya, G. Singh, S. Singh, B. Yoon, Smo-Dnn, Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection, Electronics 9 (2020) 692, https://doi.org/10.3390/electronics9040692.

[37] A. Kamilaris, F.X. Prenafeta-Boldú, Deep learning in agriculture: a survey, Comput. Electron. Agric. 147 (2018) 70−90, https://doi.org/10.1016/j.compag.2018.02.016.

[38] W. Zhong, N. Yu, C. Ai, Applying big data based deep learning system to intrusion detection, Big Data Min. Anal. 3 (2020) 181−195, https://doi.org/10.26599/bdma.2020.9020003.

[39] S. Al-Emadi, A. Al-Mohannadi, F. Al-Senaid, Using deep learning techniques for network intrusion detection, in: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT, 2020, pp. 171−176, https://doi.org/10.1109/ICIoT48696.2020.9089524.

[40] F. Meng, Y. Fu, F. Lou, Z. Chen, An effective network attack detection method based on kernel PCA and LSTM-RNN, in: International Conference on Computer Systems, Electronics and Control, ICCSEC, 2017, pp. 568−572, https://doi.org/10.1109/ICCSEC.2017.8447022.

[41] Y.M. Banadaki, Detecting malicious DNS over HTTPS traffic in domain name system using machine learning classifiers, J. Chem. Soc. A 8 (2020) 46−55, https://doi.org/10.12691/jcsa-8-2-2.