Journal of Computer Science

Original Research Paper

Improved Intrusion Detection System to Alleviate Attacks on DNS Service

[1]Hani Mahmoud Al-Mimi, [2]Nesreen Adnan Hamad, [3]Mosleh Mohammad Abualhaj, [3]Sumaya Nabil Al-Khatib and [3]Mohammad Osama Hiari

[1]Department of Cybersecurity, Faculty of Science and Information Technology, Al-Zaytooanh University of Jordan, Amman, Jordan
[2]Department of Artificial Intelligence, Faculty of Science and Information Technology, Al-Zaytooanh University of Jordan, Amman, Jordan
[3]Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan

Article history
Received: 26-06-2023
Revised: 08-09-2023
Accepted: 16-09-2023

Corresponding Author:
Hani Mahmoud Al-Mimi
Department of Cybersecurity, Faculty of Science and Information Technology, Al-Zaytooanh University of Jordan, Amman, Jordan
Email: hani.mimi@zuj.edu.jo

Abstract: Cybercriminals continuously devise new and more sophisticated ways to attack their targets' security and cyberattacks are on the rise. One of the earliest and most vulnerable network services is the Domain Name System (DNS), which has had several security issues that have been repeatedly exploited over time. Building a strong Intrusion Detection System (IDS) that guards against unwanted access to network resources is essential to identify DNS attacks in the network and safeguard data. Recently, a number of interesting approaches have been developed as a cure-all for intrusion detection, but constructing a safe DNS system remains difficult because attackers frequently alter their tactics to move around the system's security measures. In this study, we provide a self-learning model that detects the new attacks on DNS using machine learning classifiers. Support Vector Machine (SVM), K-nearest neighbor, Naive Bayes, and Decision Tree are used in the proposed model to classify data as intrusive or normal. The UNSW_NB15 dataset is used to assess the model performance. Data are pre-processed to eliminate irrelevant attributes from the dataset given that the dimensions of the data affect the success of an IDS. Empirical findings show that SVM and Decision Tree have the best performance for all the classifiers, with an accuracy rate of 99.99%. The performance of Naive Bayes is 99.89% for all attack types, which is the lowest of all the classifiers.

Keywords: Machine Learning, DNS Attacks, IDS Systems, UNSW_NB15 Dataset

Introduction

The physical world is increasingly changing into a digital one, which faces a huge number of security threats by so-called hackers. Statistics show that the global number of malware attacks in 2019 was approximately 10 billion (Abualhaj *et al*., 2022; Petrosyan, 2023). As such, cybersecurity has become a necessity for all companies. Cybersecurity is the practice of protecting computer systems, networks, and programs from digital attacks (Dawson *et al*., 2022). In 2019, spending in the cybersecurity business reached approximately 40.8 billion dollars in the United States (Statista, 2023).

Hackers target mostly the main services of the digital world, such as the Domain Name System (DNS). Users enter domain names and the DNS servers' role is to find an IP address that corresponds to each domain. DNS is one of the first and most susceptible network services, with various security flaws that have been regularly exploited throughout time (Satoshi and Hiroyuki, 2020). In 2021, 72% of organizations were hit by DNS attacks, which resulted in enormous financial damages (Help Net Security, 2021). Thus, the timely detection and classification of malicious activity on DNS is essential. Several types of attacks can be done on DNS servers, including DNS spoofing, DNS tunneling, DNS flood attack, NXDOMAIN attack, and DNS amplification. A specific technique is used for mitigating each of these attacks. However, many other new types of attacks on DNS are devised and are continuously increasing (Singh and Roy, 2020; Li *et al*., 2021), leading to difficulties in finding mitigation techniques and coping with new attack types.

Recently, Machine Learning (ML) science has helped to enhance the exposure capability of attacks. ML techniques have been used with smart security tools, such as Intrusion Detection and Prevention Systems (IDPS), to stop DNS service attacks. IDPS monitors and analyzes the system events to detect and alert users of any unauthorized attempts to access the system resources in actual or close to real-time (Alharbi *et al*., 2021; Maabreh *et al*., 2022; Abdel-Fattah *et al*., 2021). In this study, we provide a self-learning model that detects new attacks on DNS using ML techniques. The Supervised ML (SML) methods of ML are utilized in the suggested model. These algorithms function on structured and labeled data, comparable with that utilized by the IDPS. The proposed model implements several SML algorithms to choose the best suited for the detection of DNS systems attack and thus the proposed model is called SML-DNS.

Background

This section elaborates on a few of the topics that make the paper more understandable for the reader. These topics include the UNSW_NB15 dataset, SML algorithms, the Min-max scaler technique, and the K-Fold cross-validation technique.

UNSW_NB15 Dataset

Unsw-Nb15 is a network intrusion dataset. The well-known TCPDUMP tool is utilized to collect a massive amount of raw traffic in the cyber range laboratory of the Australian Centre for cyber security. The training dataset has 175,341 records and the testing dataset has 82,332 records of attack and normal traffic. Attack traffic has nine dissimilar kinds: Generic, worms, fuzzers, backdoors, analysis, DoS, exploits, shellcode, and reconnaissance (Kasongo and Sun, 2020; Moustafa and Slay, 2015). Table 1 shows the number of records of each attack type. These attacks are distributed over several services: POP3, FTP, SMTP, RADIUS, HTTP, IRC, SNMP, SSH, DNS, SSL, DHCP, and others (-) (several services are unknown). Several tools and algorithms are used to generate a total of 42 features, which produces a well-designed dataset for evaluating network anomaly detection systems (Moustafa and Slay, 2015; Zoghi and Serpen, 2021). Table 2 describes the features of the UNSW-NB15 dataset.

Table 1: Number of records for each attack in UNSW-NB15

Attack type	Number of records
Reconnaissance	13988
Backdoors	2329
DoS	16353
Analysis	2677
Exploits	44525
Generic	58871
Shellcode	1511
Worms	174
Fuzzers	24247
Normal (No attack)	92998
Total	257673

Table 2: The features of the UNSW-NB15 dataset

#	Feature	Type	Description	Min. value	Max. value
1	dur	Numeric	Duration	0	59.996017
2	Proto	Nominal	Transport protocol	N/A	N/A
3	Service	Nominal	DNS, FTP-data, HTTP, SSH, IRC, FTP, SMTP, and (-) if not a much-used service	N/A	N/A
4	State	Nominal	Refers to the state and it's any relying protocols, e.g., PAR, TXD, ECO, RST, ECR, MAS, ACC, REQ, URN, CON, CLO, FIN, TST, INT, URH and (-) (No state)	N/A	N/A
5	spkts	Numeric	Number of packets to destination (DEST)	1	512
6	dpkts	Numeric	Number of packets to source (SRC)	0	800
7	sbytes	Numeric	Number of bytes to DEST	65	44196
8	dbytes	Numeric	Number of bytes to SRC	0	60800
9	Rate	Numeric	Ethernet data rates transmitted and received	0	1000000
10	STTL	Numeric	Time to live (TTL) to DEST	31	255
11	dttl	Numeric	TTL to SRC	0	254
12	sload	Numeric	SRC BPS	0	2.224E+09
13	dload	Numeric	DEST BPS	0	818390.81
14	sloss	Numeric	Lost or resent packets at SRC	0	2
15	dloss	Numeric	Lost or resent packets at DEST	0	6
16	sinpkt	Numeric	Interpacket incoming time at SRC (MSEC)	0	13992.212
17	dinpkt	Numeric	Interpacket incoming time at DEST (MSEC)	0	13992.285
18	sjit	Numeric	Jitter at SRC (MSEC)	0	19787.972

Table 2: Continue

19	djit	Numeric	Jitter at DEST (MSEC)	0	19788
20	swin	Numeric	TCP window Length (Len) advertised by SRC	0	255
21	stcpb	Numeric	Seq. No. of TCP at SRC	0	4277446941
22	dtcpb	Numeric	Seq. No. of TCP at DEST	0	4088422545
23	dwin	Numeric	TCP window Len advertised by DEST	0	255
24	tcprtt	Numeric	TCP session initiation round-trip time ('synack' + 'ackdat')	0	0.265426
25	synack	Numeric	TCP session initiation duration, the duration of the SYN and the SYN_ACK	0	0.176175
26	ackdat	Numeric	TCP session initiation time, the time between the SYN_ACK and the ACK	0	0.136995
27	smean	Numeric	Mean of the connection packet Len send by the SRC	50	834
28	dmean	Numeric	Mean of the connection packet Len send by the DEST	0	865
29	trans_depth	Numeric	Symbolizes the pipelined depth into the flow of http request/response traffic	0	131
30	response_body_ len	Numeric	The Len of uncompressed data sent by the http server	0	5242880
31	ct_srv_src	Numeric	No. of HTTP sessions that hold the exact SRC address in 100 sessions	1	59
32	ct_state_ttl	Numeric	In a particular range of values, the number of each state (4) for SRC/DEST TTL (10) (11)	0	6
33	ct_dst_ltm	Numeric	The ratio of sessions with the identical destination address out of 100 sessions	1	59
34	ct_src_dport_ ltm	Numeric	The ratio of sessions with the identical SRC IP address and DEST port number out of 100 sessions	1	59
35	ct_dst_sport_ ltm	Numeric	The ratio of sessions with the identical DEST IP address and SRC port number out of 100 sessions	1 1	38 59
36	ct_dst_src_ ltm	Numeric	The ratio of sessions with the identical SRC and DEST IP address out of 100 sessions		
37	is_ftp_login	Numeric	If the user and password are utilized to log in the FTP session, then 1; else, 0	0	1
38	ct_ftp_cmd	Numeric	The number of flows that contain a command in an FTP connection	0	2
39	ct_flw_http_ mthd	Numeric	The number of flows that contain methods such as as Get and Post in http connection	0	16
40	ct_src_ltm	Numeric	The ratio of sessions of the identical SRC address in 100 connections according to the last time	1	60
41	ct_srv_dst	Numeric	The ratio of HTTP sessions that has the identical DEST IP address in 100 sessions	1	59
42	is_sm_ips_ ports	Numeric	If the SRC IP is the same as the DEST IP and port numbers are also the same, then 1, else 0	0	1

This study is interested only in attacks on the DNS service. Therefore, the DNS service records are extracted from the UNSW-NB15 dataset (for both training and testing) and all other records of the other services are removed. Thus, a new sub-dataset of the UNSW-NB15 dataset, called DNS-UNSW-NB15, is created and used in this study. The DNS-UNSW-NB15 training dataset has 21367 records and the testing dataset has 47294 records of attack and normal traffic. These two datasets have been combined as one, for training and testing, yielding 68,661 records of attack and normal traffic. The attack traffic, in DNS-UNSW-NB15, contains five different types: Fuzzers, DoS, Exploits, Generic, and Reconnaissance. Table 3 displays the number of records of each attack kind.

Table 3: Number of records for each attack in DNS-UNSW-NB15

Attack type	Number of records
Fuzzers	375
DoS	147
Exploits	253
Generic	57278
Reconnaissance	47

ML Techniques Used in this Article

Naive Bayes

The Naive Bayes method is a classification method that exploits Bayes' Theorem and operates on the premise that predictors are independent of one another. A Naive

Bayes classifier, in simplest terms, is predicated on the idea that the existence of one characteristic in a class is unlinked to that of any other characteristic. In spite of the fact that these characteristics are dependent either on one another or on the existence of other characteristics, they all contribute to the probability of the object's nature, which is why it is considered "naïve". The Naive Bayes model is straightforward to develop and excel when applied to large data sets in particular. This technique is recognized for its ability to outperform even the most complex classification systems, despite its straightforward nature (Çavuşoğlu, 2019; Ma *et al*., 2020). The probabilistic expressions that are utilized in Bayes' Theorem are shown in Eq. 1:

$$R(M|S) = \frac{R(S|M)R(m)}{R(S)} \quad (1)$$

K-Nearest Neighbors (K-NN)

The K-NN algorithm is a non-parametric, supervised learning classifier that uses vicinity to categorize or forecast the grouping of a single data point. K-NN is frequently used as a classification method and relies on the notion that similar points can be found close together. For classification problems, K-NN attempts to forecast the true class for the test data by finding its distance from all the training points, using, for example, the Euclidean algorithm (Çavuşoğlu, 2019; Wang *et al*., 2021) the formula of which is displayed in Eq. 2:

$$Euclidean\ Distance\ between\ L\ and\ K = \sqrt{(R_2 - R_1)^2 + (T_2 - T_1)^2} \quad (2)$$

Decision Tree

The decision tree classification method is among the most straightforward and common approaches that can be used in the case of regression and classification problems. Utilizing a decision tree serves the objective of developing a training model eligible for forecasting the class label of a target point by acquiring fundamental decision-making skills from previously collected data (training data). When using decision trees, we start at the base to determine the class label to assign to a record and then check to see if the values of the root attributes are the same as those of the record by comparison. In light of the comparison, we move on to the subsequent node after taking the path indicated by the branch matching the value in question (Çavuşoğlu, 2019; Elaidi *et al*., 2018). Figure 1 clarifies the decision tree algorithms.

SVM

For problems involving classification and regression, the Support Vector Machine (SVM) is among the SML algorithms that are exploited most frequently. However, its primary function is to overcome difficulties associated with classification. The objective of the SVM method is to pinpoint the best decision line for grouping n-dimensional space into classes to facilitate the straightforward assignment of new data points in the appropriate group in the future. The border of the best choice is at times referred to as a hyperplane. SVM is exploited to select the extreme points and vectors that are used in the creation of the hyperplane. Finding the hyperplane that most clearly differentiates the two classes is the first step in the classification (Çavuşoğlu, 2019; Ma *et al*., 2020). Figure 2 clarifies the SVM Algorithms.

MinMaxScaler

When numerical input variables are scaled to a standard range, SML algorithms perform better. SML algorithms learn how to translate input variables to output variables. The former may have various units, which means that their scales may differ, which increases the difficulty of modeling. MinMaxScaler is a continuous variable scaling method (Ahsan *et al*., 2021). The input variables are between the minimum of 0 and maximum of 1, using Eq. 3, where S_{new} is a newly derived value, S is the original value, and $S_{\min}$ and $S_{\max}$ are the minimum and maximum values of the feature, respectively:

$$S_{new} = (S - S_{\min})/(S_{\max} - S_{\min}) \quad (3)$$

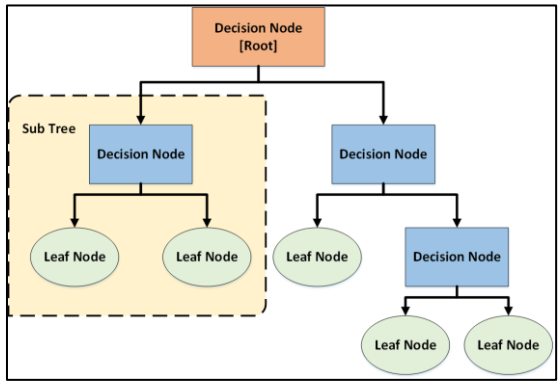

Fig. 1: Decision tree technique scheme

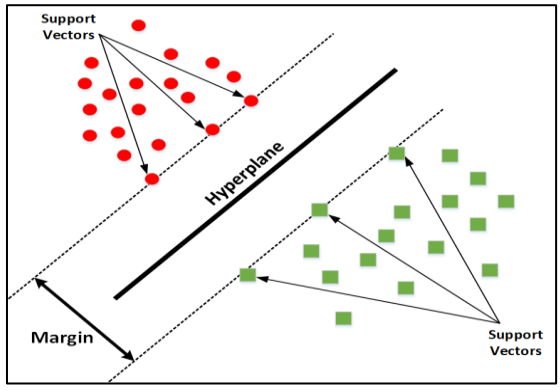

Fig. 2: SVM technique scheme

K-Fold Cross-Validation

A resampling technique recognized as k-fold cross-validation is exploited to assess ML models by exploiting a partial volume of data. The technique has a single variable known by the letter k. This variable determines how many groups a particular data sample needs to be divided into. If a particular value for k is given, then that value can be placed for K in the assessed model. For example, k = 5 means 5-fold cross-validation (Wong and Yang, 2017).

Related Works

DNS attacks have always been an area of great concern for cybersecurity research. This section discusses a few studies on mitigating DNS attacks.

DNS Anomaly Detection Visual Platform (DNS-ADVP) was created by Trejo *et al*. (2019) as an integrated platform with the goal of quickly detecting and mitigating an ongoing DNS DDoS attack. DNS-ADVP has a one-class classifier that attracts attention to the situation if the DNS server engages in anomalous behavior and offers a visual model that can be used to interpret the present traffic condition in an authoritative DNS server. An in-depth analysis of well-known ways that mitigate DNS DDoS attacks is examined to accomplish this objective. These approaches include a UDP rule to restrict the rate of requests done by the same IP, the well-known Response Rate Limit (RRL), and an existing methodology. The DNS-ADVP approach is used to establish the degree to which these strategies are important and how well-suited them for use in a production setting. To construct DNS-ADVP, Trejo *et al*. (2019) have presented a unique visual model to rapidly evaluate the current DNS traffic and to flag any irregularities in a timely manner using visual semaphores. In conjunction with this, they create a new classifier based on K-NN, which is specifically designed to evolve according to the dynamic nature of the traffic seen by authoritative DNS servers. When the classifier is put through its paces using simulated attacks in the trials, it attained an Area Under the Curve (AUC) score of 83% (Trejo *et al*., 2019).

Lyu *et al*. (2021) proposed a technique for spotting distributed DNS attacks. The proposed method exploits a hierarchical graph structure to monitor DNS data at three levels of the host, subnet, and Autonomous System (AS), paired with ML that recognizes aberrant behaviors at several hierarchical levels. The approach being suggested can identify distributed attacks even with small rates and covert patterns. Three different contributions may be made using the suggested strategy. First, monitoring real DNS traffic from the edges of two large industrial networks over the course of a week (almost 400 million packets) to highlight the different kinds of inbound DNS inquiries and the behavior of malicious entities creating query scans and floods. Second, creating a hierarchical graph structure to monitor DNS activity, identifying essential properties, training, tuning, and evaluating anomaly detection models at various levels of the hierarchy to achieve an accuracy of over 99% at each hierarchical level. Third, the scheme is applied to a month's worth of DNS data from the two enterprises and compared the outcomes in contrast to blacklists and firewall logs to demonstrate the scheme's capability of spotting distributed attacks that may be missed by legacy techniques while still maintaining decent real-time performance (Lyu *et al*., 2021).

Jin *et al*. (2019) suggested a unique detection technique for defending against DNS cache poisoning threats using ML methods. Jin *et al*. (2019) seeks to add a significant number of additional characteristics to the suggested technique besides the fundamental 5-tuple information of a DNS packet that has been retrieved depending on the ordinary DNS protocol and the heuristic aspects. These characteristics are "characteristics associated with Time", "characteristics associated with GeoIP" and "trigger of cached DNS data". This part allows for easier recognition of the DNS response packets that are exploited in cache poisoning attacks, particularly those that come from hacked authoritative DNS servers. The suggested technique, which is still a work in progress, is described along with the core perception of the experimental environment and desired network structure. This is done while the prototype is being implemented, training data is being prepared and the model is being created (Jin *et al*., 2019).

Chowdhary *et al*. (2021) proposed two distinct approaches for spotting the DNS Tunneling query. Later, these methods are merged to develop a DNS tunneling attack detector, which has the capability of informing the user about a possible attack taking place in real-time. The first approach makes use of cache misses in a DNS cache server while the second approach uses the techniques from ML to categorize a specific DNS query. In the classification of DNS tunneling data, KNN had the greatest accuracy at 93.955%, beating out all of the other traditional ML and ensemble models. If the amount of time the model takes is considered, then the Decision Tree algorithm achieves the highest level of accuracy, reaching 91.025%. Accuracies have been reached to clarify that entropy is based on the hostname is a useful feature for DNS tunneling detection. This outcome can be seen depending on the obtained results from Chowdhary *et al*. (2021).

To detect malicious behavior at the DNS level in a DNS over HTTPS (DoH) context, Singh and Roy (2020) utilized various ML classifiers such as (i) Naive Bayes, (ii) Logistic Regression, (iii) Random Forest, (iv) K-NN and (v) Gradient Boosting. These classifiers are tested on a new benchmark, known as the MoH dataset, which is

freely accessible MontazeriShatoori *et al.* (2020). The performance study reveals a distinct classification of two types of traffic: Benign and malicious DoH requests. According to the findings, both the Random Forest and the Gradient Boosting have achieved a maximum accuracy of 100% and an F1 measure in both types of traffic. KNN and Logistic Regression accuracy rates for malicious DoH are 99 and 98%, respectively, which also results in improved performance. When compared to the other four classifiers, the Naive Bayes classifier demonstrates an inferior performance. Considering the results of the performance study, we can deduce that ensemble learning-based classifiers such as Random Forest and Gradient Boosting are the most effective solutions for dealing with issues of this nature (Singh and Roy, 2020).

Moubayed *et al*. (2018) devised a solution that was based on ML to combat the typo-squatting issue. To accomplish this goal, exploratory data analytics are first applied to obtain a deeper comprehension of the patterns that are spotted in eight domain name-based extracted attributes. In addition, a majority of voting-based ensemble learning classifier that is constructed using five different classification algorithms has been offered as a method that is capable of accurately detecting questionable domains. In addition, the patterns that are discovered are corroborated by researching the same characteristics in an unlabeled dataset using the K-means clustering approach and by applying the built ensemble learning classifier. According to the findings, legitimate domains have shorter names with fewer distinct characters and a shorter overall length. In addition, the constructed ensemble learning classifier has improved performance in terms of accuracy, precision, and F-score. In addition, when clustering is applied, comparable tendencies are seen as a result. Despite this, a significant number of domains have been flagged as possibly malicious. As a consequence, the ensemble learning classifier has been used, the results reveal that the number of domains that have been classified to be possibly suspicious decreased by almost a factor of five while still keeping the same trends in terms of the statistics of features (Moubayed *et al*., 2018).

As evident from prior studies, some have been tailored to address specific DNS attacks and have not adequately tackled the latest and most prevalent threats. In contrast, the DNS-UNSW-NB15 dataset encompasses a range of attack types that can target DNS systems, offering a more comprehensive evaluation platform. Furthermore, there is room for improvement in performance results. While (Singh and Roy, 2020) achieved a remarkable 100% accuracy, it's crucial to note that this achievement pertains to a specific type of attack, unlike our model, which is designed to handle multiple attack types. We will employ four robust classifiers (Naive Bayes, K-NN, Decision Tree, and SVM) to assess performance. We will evaluate them using four key metrics: Accuracy, Recall, Precision, and Matthews Correlation Coefficient (MCC).

Proposed SML-DNS Attacks Detection Model

This section discusses the model of detecting the attacks on DNS service, including pre-processing of the DNS_UNSW_NB15 dataset to be ready for training and testing the proposed SML-DNS model, and the DNS attack detection model is presented in detail.

DNS_UNSW_NB15 Dataset Preprocessing

Data transformation and normalization operations must be made on the DNS_UNSW_NB15 dataset to be ready for applying the SML algorithms.

Manual Features Filtration

As mentioned earlier, the DNS_UNSW_NB15 dataset has been filtered to include only the DNS records. As a consequence, several features must be removed because they are highly unrelated to the DNS service. Table 4 summarizes the excluded features.

Table 4: Excluded features

#	Feature	Description	Reason
3	Service	DNS, FTP-data, http, SSH, IRC, FTP, SMTP and (-) if not a much-used service	Contains only DNS service, all other services were removed
29	Trans_depth	Symbolizes the pipelined depth into the the flow of http request/response traffic	Related HTTP service
30	response_body_len	The Len of uncompressed data sent by the http server	Related to HTTP service
37	is_ftp_login	If the user and password are utilized to login the FTP session, then 1; else, 0.	Related to FTP service
38	ct_ftp_cmd	The number of flows that contain a command in an FTP connection	Related to FTP service
39	ct_flw_http_mthd	The number of flows that contain methods such as Get and Post in the http connection	Related to HTTP service
42	is_sm_ips_ports	If the SRC IP is the same as the DEST IP and port numbers are also the same, then 1, else 0	All values are equal to zero with the DNS service

Data Transformation

Data transformation is a method exploited to transform the raw data into an appropriate form that eases retrieving strategic information. One of the mandatory transformations is converting non-numeric features into numeric ones. The DNS_UNSW_NB15 dataset contains several non-numeric features that need transformation into numeric. Label Encoding is one of the well-known methods used by SML to transform the labels into a numeric format to transform them into a machine-readable format (Abualhaj *et al*., 2022; Dirin and Saballe, 2022; Jia and Zhang, 2021). Table 5 shows the features that have been converted into numeric and their new values. Tables 6-7 show samples of the DNS_UNSW_NB15 dataset before and after the label encoding, respectively. In addition, the target of this study is to classify the attack or normal traffic, without showing the attack type. Therefore, all attack types have been labeled only in the output column. Thus, the output column contains now only two labels, Normal and Attack, which have been transformed into 0 and 1, respectively, as shown in Tables 6-7.

Table 5: Features transformation

#	Feature	Before transformation	After transformation
2	Proto	TCP	0
		UDP	1
4	State	INT	0
		FIN	1
		CON	2
		REQ	3

Table 6: Before the transformation

No.	Instances	Label
1	4.490101, UDP, INT, 116, 0, 17078, 0, 25.611897, 254, 0, 30165.91406,0, 0, 0, 39.044355, 0, 53.99923, 0, 0, 0, 0, 0, 0, 0, 0, 147, 0, 2, 2, 2, 2, 2, 2, 4, 2	Fuzzers
2	0.008666, UDP, CON, 2, 2, 126, 240, 346.180462, 62, 252, 58158.31641, 110777.75, 0, 0, 0.009, 0.008, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 63, 120, 1, 3, 1, 1, 1, 1, 1, 1	DoS
3	0.344168, TCP, FIN, 10, 14, 546, 10068, 66.827827, 254, 252, 11436.27539, 217312.4688, 2, 5, 38.240889, 19.374538, 2014.262464, 1516.292252, 255, 4060329867, 311844409, 255, 0.120871, 0.055742, 0.065129, 55, 719, 1, 1, 1, 1, 1, 2, 2, 1	Exploits
4	13.337146, UDP, REQ, 12, 0, 936, 0, 0.824764, 254, 0, 514.652832, 0, 0, 0, 1340.393875, 0, 1414.228875, 0, 0, 0, 0, 0, 0, 0, 0, 78, 0, 10, 6, 5, 5, 5, 5, 5, 5	Fuzzers
5	0.000004, UDP, INT, 2, 0, 130, 0, 250000.0006, 31, 0, 130000000, 0, 0, 0, 0.004, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 65, 0, 3, 0, 3, 3, 1, 1, 4, 5	Normal

Table 7: After transformation

No.	Instances	Label
1	4.490101, 1, 0, 116, 0, 17078, 0, 25.611897, 254, 0, 30165.91406,0, 0, 0, 39.044355, 0, 53.99923, 0, 0, 0, 0, 0, 0, 0, 0, 147, 0, 2, 2, 2, 2, 2, 2, 4, 2	1
2	0.008666, 1, 2, 2, 2, 126, 240, 346.180462, 62, 252, 58158.31641, 110777.75, 0, 0, 0.009, 0.008, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 63, 120, 1, 3, 1, 1, 1, 1, 1, 1	1
3	0.344168, 0, 1, 10, 14, 546, 10068, 66.827827, 254, 252, 11436.27539, 217312.4688, 2, 5, 38.240889, 19.374538, 2014.262464, 1516.292252, 255, 4060329867, 311844409, 255, 0.120871, 0.055742, 0.065129, 55, 719, 1, 1, 1, 1, 1, 2, 2, 1	1
4	13.337146, 1, 3, 12, 0, 936, 0, 0.824764, 254, 0, 514.652832, 0, 0, 0, 1340.393875, 0, 1414.228875, 0, 0, 0, 0, 0, 0, 0, 0, 78, 0, 10, 6, 5, 5, 5, 5, 5, 5	1
5	0.000004, 1, 0, 2, 0, 130, 0, 250000.0006, 31, 0, 130000000, 0, 0, 0, 0.004, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 65, 0, 3, 0, 3, 3, 1, 1, 4, 5	0

Table 8: After normalization

No.	Instances	Label
1	0.074839985, 1, 0, 0.225048924 0, 0.385511319, 0, 2.56E-05, 0.995535714, 0, 1.36E-05, 0, 0, 0.002790435, 0, 0.002728892, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.12372449, 0, 0.017241379, 0.333333333, 0.017241379, 0.017241379, 0.027027027, 0.017241379, 0.050847458, 0.0172413790.000144443,	1
2	1, 0.666666667, 0.001956947, 0.0025, 0.001382248, 0.003947368, 0.00034618, 0.138392857, 0.992125984, 2.62E-05, 0.135360452, 0, 0, 6.43E-07, 5.72E-07, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.016581633, 0.138728324, 0, 0.5, 0, 0, 0, 0, 0, 0	1
3	0.000956413, 1, 0.666666667, 0.001956947, 0.0025, 0.00120097, 0.005164474, 5.23E-05, 0.995535714, 0.236220472, 3.70E-06, 0.026746121, 0, 0, 2.86E-07, 5.00E-07, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.011479592, 0.18150289, 0.017241379, 0, 0.017241379, 0.017241379, 0.027027027, 0.034482759, 0.016949153, 0.017241379	1
4	5.00E-08, 1, 0, 0.001956947, 0, 0.003421631, 0, 0.333333321, 0.995535714, 0, 0.129496403, 0, 0, 0, 2.14E-07, 0, 0, 2.14E-07, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.073979592, 0, 0, 0.333333333, 0.034482759, 0, 0, 0.034482759, 0.050847458, 0	1

Data Normalization

Data normalization is the procedure of transforming the columns in a dataset to the same scale. Thus, the SML techniques perform better, because the large value features do not dominate the small value features (Çavuşoğlu, 2019; Ahsan *et al*., 2021). The aforementioned MinMaxScaler is used by the proposed SML-DNS method to normalize the DNS_UNSW_NB15 dataset. Table 8 shows a sample of the DNS_UNSW_NB15 dataset after normalization. Figure 3 illustrates the UNSW_NB15 dataset data preprocessing steps.

Proposed SML-DNS Model

The core objective of the SML-DNS model is to discover attacks on the DNS service. For this, the SML-DNS model trains several ML techniques on the DNS_UNSW_NB15 dataset and tests the performance of these techniques. Four SML techniques are used by the SML-DNS model: Naive Bayes, K-NN, Decision Tree, and SVM, as discussed above. In stage 1, the UNSW_NB15 dataset prepares for the SML-DNS model, and in stage 2, the proposed SML-DNS model trains and tests the aforementioned SML techniques to achieve the intended goal. In stage 1, the UNSW_NB15 dataset is first filtered to contain the DNS service records, only to produce the DNS_UNSW_NB15 dataset. Then, the features that are unrelated to the DNS service are manually removed from the DNS-UNSW_NB15 dataset. This step is a result of the previous step. After that, all types of attacks in the output columns have been labeled as "Attack". This is because the SML-DNS model aims to determine whether the DNS traffic is attack or normal, regardless of the type of attack. All the non-numeric data in the DNS-UNSW_NB15 dataset has been changed into numeric (using the label encoding algorithm) to suit the SML methods. Finally, the data in the DNS_UNSW_NB15 dataset has been scaled using the MinMaxScaler algorithm. Figure 4 shows the steps performed in Stage 1. Now, the DNS_UNSW_NB15 dataset is ready to perform stage 2. First, the four SML algorithms (Naive Bayes, K-NN, Decision Tree, and SVM) used by the SML-DNS model are trained on the DNS_UNSW_NB15 dataset. The performance of these algorithms is tested to choose the best. The training and testing are achieved with the help of the K-fold technique (discussed above), the DNS_UNSW_NB15 dataset is divided into 10 groups, nine of them are used for training and the remaining are used for testing. The training and testing groups keep changing until all of them are used for training and testing. Thus, the SML-DNS model is evaluated better and the result has no bias. Finally, the SML-DNS model responds with either normal or attack traffic. Figure 4 shows the steps performed in stage 2.

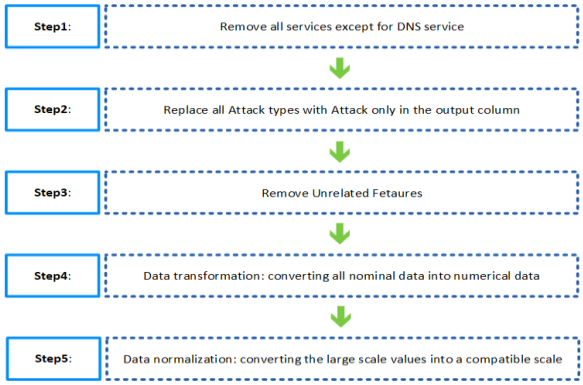

Fig. 3: UNSW_NB15 dataset preprocessing steps

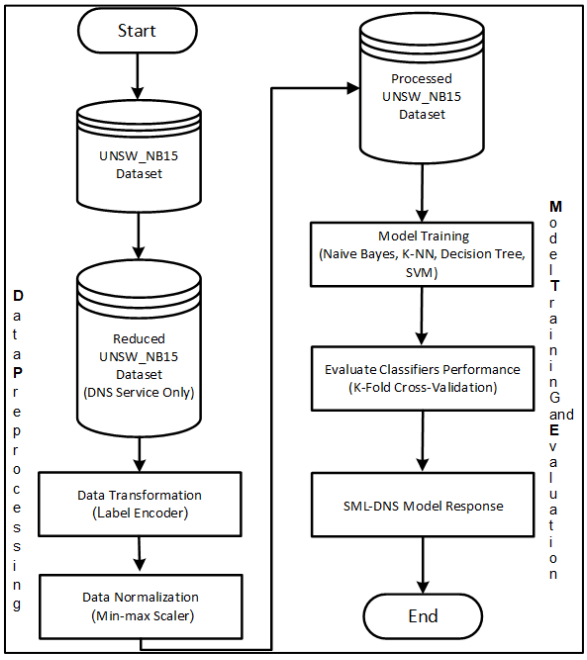

Fig. 4: SML-DNS model (Stage 1 and Stage 2)

Results and Discussion

In this section, the suggested model (SML-DNS) is evaluated along with showing the implementation environment. The proposed SML-DNS model is implemented using Python, a powerful programming language that supports multiple paradigms. In 2022, Python ranked number one among the currently on-the-rise programming languages. Python supports all the packages that are needed to implement and evaluate the proposed SML-DNS model. A few of the used packages to implement the proposed SML-DNS model include "sklearn" and "pandas". The implantation environment is as follows: Python 3.10.4 programming language, Intel Core i7-9750H CPU, 32GB RAM, and 64-bit MS Windows operating system.

Typically, the elements of the confusion matrix are used to evaluate a proposed SML model. The elements of the confusion matrix are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP means that the model successfully classifies an attack. TN means that the model also successfully classifies no attack. FP means that the model has unsuccessfully classified an attack. FN means that the model has unsuccessfully classified that there is no attack (Pangaliman *et al*., 2018; Ajithkumar *et al*., 2017). Figure 5 shows the confusion matrix. Several metrics can be calculated from the confusion matrix to evaluate a proposed SML model, a few of which are Accuracy (Acc), Recall (Re), Precision (Pr), and Matthews Correlation Coefficients (MCC). Acc is considered one of the best metrics to evaluate an SML model, in our proposed SML-DNS model, it is the number of correct DNS attack predictions to the total number (correct and wrong) of DNS attack predictions. Equation 4 can be used to calculate the accuracy of the proposed SML-DNS model. Another important metric to evaluate an SML model is Re, which in our proposed model is the number of correct predictions of DNS attacks to the total number (correct and wrong) of actual DNS attacks. Equation 5 can be used to calculate the proposed SML-DNS model Re. Pr is also considered a key metric to evaluate an SML model. In our proposed model, the number of correct actual positive DNS attack predictions to the total number (correct and wrong) of positive DNS attack predictions. Equation 6 can be used to calculate the proposed SML-DNS model Pr. The last metric that must be used to evaluate an SML model is MCC, which is utilized to evaluate the quality of a binary classification model, similar to our proposed SML-DNS model. Equation 7 can be used to calculate the proposed SML-DNS model MCC (Abualhaj *et al*., 2022; Jia and Zhang, 2021; Pangaliman *et al*., 2018). Note that the SML-DNS is abbreviated as *SD* in the equations:

$$SD_{Acc} = \frac{(SD_{TP} + SD_{TN})}{(SD_{TP} + SD_{TN} + SD_{FP} + SD_{FN})} \quad (4)$$

$$SD_{Re} = \frac{(SD_{TP})}{(SD_{TP} + SD_{FN})} \quad (5)$$

$$SD_{Pr} = \frac{(SD_{TP})}{(SD_{TP} + SD_{FP})} \quad (6)$$

$$SD_{MCC} = SD_{MCC} = \frac{((SD_{TP} * SD_{TN}) - (SD_{FP} * SD_{FN}))}{\sqrt{(SD_{TP} + SD_{FP}) * (SD_{TP} + SD_{FN}) * (SD_{TN} + SD_{FP}) * (SD_{TN} + SD_{FN})}} \quad (7)$$

Figures 6-9, respectively, show the accuracy, recall, precision, and MCC of the suggested SML-DNS model with the four examined methods: Decision tree, KNN, Naive Bayes, and SVM. Figure 6 shows that the Naive Bayes attained the lowermost accuracy (99.89%) among all four methods, while the other three methods achieved the same accuracy (99.99%). Figure 7 shows that the four methods attained the same recall (100%). Figure 8 shows that the SVM and decision tree methods achieved the uppermost precision (99.99%), while the Naive Bayes attained the lowermost precision (99.87%). Figure 9 shows that the SVM and decision tree methods attained the uppermost MCC (99.97%), while the Naive Bayes method attained the lowermost MCC (99.58%).

In general, all methods perform well with the suggested SML-DNS model. However, the SVM and Decision Tree methods can be considered the best among the four methods because they outperform the others in all four measures. In addition, the SVM and Decision Tree techniques outperform the other techniques with the Accuracy metric, which is considered one of the most reliable among the four metrics in the proposed SML-DNS model and the used dataset.

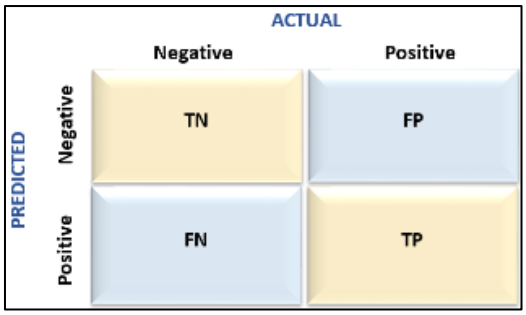

Fig. 5: Confusion matrix

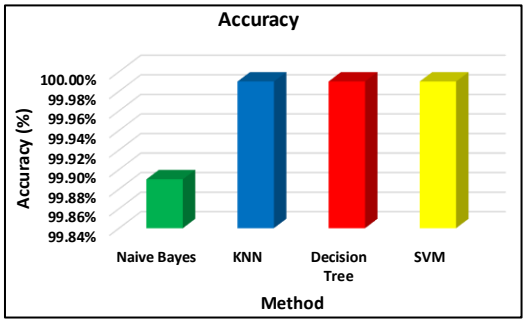

Fig. 6: Accuracy of the SML-DNS

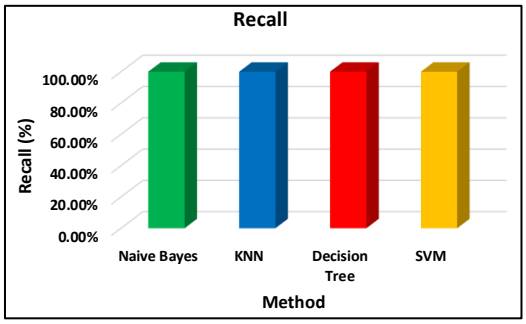

Fig. 7: Recall of the SML-DNS

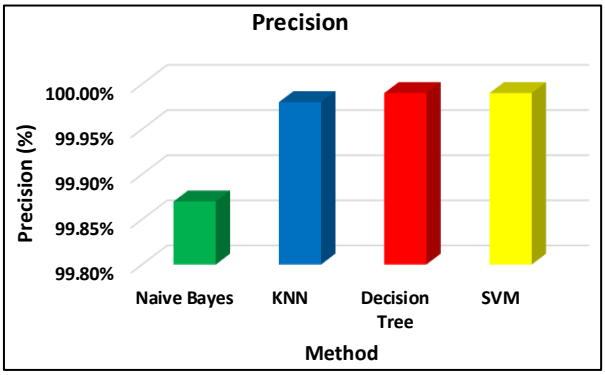

Fig. 8: Precision of the SML-DNS

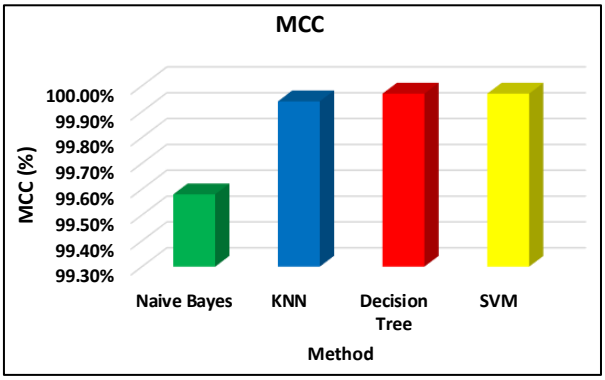

Fig. 9: MCC of the SML-DNS

Conclusion

As a result of a rise in unwanted access and exploitation of network resources, security has emerged as a major concern in recent years. Given that DNS attacks have become such a problematic issue, identifying them as soon as possible is essential such that the system network may sustain the least amount of harm possible. In recent years, ML classifiers have seen widespread use in IDS as a result of their versatility, power of generalization, and resilient nature. In the current research, an inclusive empirical study utilizing ML classifiers, specifically, SVM, KNN, Naive Bayes, and Decision Tree is carried out to detect network intrusion, and the performance of these classifiers is evaluated on UNSW_NB15. The purpose of this study is to determine how well these classifiers can detect network intrusion. After the dataset is originally preprocessed, the model is trained and evaluated in accordance with the relevant attributes that are identified. Based on the empirical findings, all of the classifiers clearly produce promising outcomes with regard to DNS attacks. However, of all of the classifiers, the best performers are the SVM and Decision Tree, both with accuracy rates of 99.99%. By contrast, the Naive Bayes shows the lowest performance at 99.89% across the board for all forms of attacks. In future works, more machine learning classifiers will be used to assess the performance of the proposed model. In addition, more DNS datasets will be used to validate the achieved results.

Acknowledgment

I highly appreciate the researchers who supported this study with all the required references. I thank the editorial board and reviewers for spending their valuable time reviewing our article and giving valuable suggestions.

Funding Information

Funded by Al-Zaytoonah University of Jordan, Project number 17/10/2022-2023.

Author's Contributions

Hani Mahmoud Al-Mimi and Mosleh Mohammad Abualhaj: Participated in all experiments, coordinated the data-analysis and contributed to the written of the manuscript.

Nesreen Adnan Hamad: Contributed to the written of the manuscript. Coordinated the mouse work.

Sumaya Nabil Al-Khatib and Mohammad Osama Hiari: Participated in all experiments.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

References

Abdel-Fattah, F., AlTamimi, F., & Farhan, K. A. (2021, July). Machine Learning and Data Mining in Cybersecurty. In *2021 International Conference on Information Technology (ICIT)* (pp. 952-956). IEEE. https://doi.org/10.1109/ICIT52682.2021.9491749

Abualhaj, M. M., Abu-Shareha, A. A., Hiari, M. O., Alrabanah, Y., Al-Zyoud, M., & Alsharaiah, M. A. (2022). A Paradigm for DoS Attack Disclosure using Machine Learning Techniques. *International Journal of Advanced Computer Science and Applications*, *13*(3). https://doi.org/10.14569/IJACSA.2022.0130325

Ahsan, M. M., Mahmud, M. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, *9*(3), 52. https://doi.org/10.3390/technologies9030052

Ajithkumar, N., Aswathi, P., & Bhavani, R. R. (2017, April). Identification of an effective learning approach to landmine detection. In *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)* (pp. 1-5). IEEE. https://doi.org/10.1109/IEMENTECH.2017.8077018

Alharbi, H. A., Alshaya, H. I., Alsheail, M. M., & Koujan, M. H. (2021). Analyzing Graduation Project Ideas by using Machine Learning. *International Journal of Interactive Mobile Technologies*, *15*(23). https://doi.org/10.3991/ijim.v15i23.27707

Çavuşoğlu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence*, *49*, 2735-2761. https://doi.org/10.1007/s10489-018-01408-x

Chowdhary, A., Bhowmik, M., & Rudra, B. (2021, May). DNS tunneling detection using machine learning and cache miss properties. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1225-1229). IEEE. https://doi.org/10.1109/ICICCS51141.2021.9432279

Dawson, M., Tabona, O., & Maupong, T. (Eds.). (2022). *Cybersecurity Capabilities in Developing Nations and its Impact on Global Security*. IGI Global. https://doi.org/10.4018/978-1-7998-8693-8

Dirin, A., & Saballe, C. A. (2022). Machine Learning Models to Predict Students' Study Path Selection. *IJIM*, *16*(01), 159. https://doi.org/10.3991/ijim.v16i01.20121

Elaidi, H., Elhaddar, Y., Benabbou, Z., & Abbar, H. (2018, April). An idea of a clustering algorithm using support vector machines based on binary decision tree. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1-5). IEEE. https://doi.org/10.1109/ISACV.2018.8354024

Help Net Security. (2021). 72% of organizations hit by DNS attacks in the past year. https://www.helpnetsecurity.com/2021/10/26/organizations-dns-attacks

Jia, B. B., & Zhang, M. L. (2021). Multi-dimensional classification via decomposed label encoding. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2021.3100436

Jin, Y., Tomoishi, M., & Matsuura, S. (2019, September). A detection method against DNS cache poisoning attacks using machine learning techniques: Work in progress. In *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)* (pp. 1-3). IEEE. https://doi.org/10.1109/NCA.2019.8935025

Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, *7*, 1-20. https://doi.org/10.1186/s40537-020-00379-6

Li, Z., Gao, S., Peng, Z., Guo, S., Yang, Y., & Xiao, B. (2021). B-DNS: A secure and efficient DNS based on the blockchain technology. *IEEE Transactions on Network Science and Engineering*, *8*(2), 1674-1686. https://doi.org/10.1109/TNSE.2021.3068788

Lyu, M., Gharakheili, H. H., Russell, C., & Sivaraman, V. (2021). Hierarchical anomaly-based detection of distributed DNS attacks on enterprise networks. *IEEE Transactions on Network and Service Management*, *18*(1), 1031-1048. https://doi.org/10.1109/TNSM.2021.3050091

Ma, T. M., Yamamori, K., & Thida, A. (2020, October). A comparative approach to Naïve Bayes classifier and support vector machine for email spam classification. In *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)* (pp. 324-326). IEEE. https://doi.org/10.1109/GCCE50665.2020.9291921

Maabreh, M., Obeidat, I., Elsoud, E. A., Alnajjar, A., Alzyoud, R., & Darwish, O. (2022). Towards Data-Driven Network Intrusion Detection Systems: Features Dimensionality Reduction and Machine Learning. *International Journal of Interactive Mobile Technologies*, *17*(14). https://doi.org/10.3991/ijim.v16i14.30197

MontazeriShatoori, M., Davidson, L., Kaur, G., & Lashkari, A. H. (2020, August). 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)* (pp. 63-70). IEEE. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026

Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H. (2018, December). DNS typo-squatting domain detection: A data analytics & machine learning based approach. In *2018 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-7). IEEE. https://doi.org/10.1109/GLOCOM.2018.8647679

Moustafa, N., & Slay, J. (2015, November). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Pangaliman, M. M. S., Cruz, F. R. G., & Amado, T. M. (2018, November). Machine learning predictive models for improved acoustic disdrometer. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)* (pp. 1-5). IEEE. https://doi.org/10.1109/HNICEM.2018.8666256

Petrosyan, A. (2023). Annual number of malware attacks worldwide from 2015 to 2022. https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide

Satoshi, K. I. M. U. R. A., & Hiroyuki, I. N. A. B. A. (2020, September). Proposal of Anomaly Detection for DNS Attacks Based on Packets Prediction Using LSTM. In *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)* (pp. 90-95). IEEE. https://doi.org/10.1109/IIAI-AAI50415.2020.00028

Singh, S. K., & Roy, P. K. (2020, December). Detecting malicious DNS over https traffic using machine learning. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (pp. 1-6). IEEE. https://doi.org/10.1109/3ICT51146.2020.9312004

Statista. (2023). Spending on cybersecurity worldwide from 2017 to 2022. *Technology & Telecommunications*. https://www.statista.com/statistics/991304/worldwide-cybersecurity-spending

Trejo, L. A., Ferman, V., Medina-Pérez, M. A., Giacinti, F. M. A., Monroy, R., & Ramirez-Marquez, J. E. (2019). DNS-ADVP: A machine learning anomaly detection and visual platform to protect top-level domain name servers against DDoS attacks. *IEEE Access*, *7*, 116358-116369. https://doi.org/10.1109/ACCESS.2019.2924633

Wang, P., Zhang, Y., & Jiang, W. (2021, June). Application of K-Nearest Neighbor (KNN) Algorithm for Human Action Recognition. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (Vol. 4, pp. 492-496). IEEE. https://doi.org/10.1109/IMCEC51613.2021.9482165

Wong, T. T., & Yang, N. Y. (2017). Dependency analysis of accuracy estimates in k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, *29*(11), 2417-2427. https://doi.org/10.1109/TKDE.2017.2740926

Zoghi, Z., & Serpen, G. (2021). Unsw-nb15 computer security dataset: Analysis through visualization. *arXiv Preprint arXiv:2101.05067*. https://doi.org/10.48550/arXiv.2101.05067