I. J. Computer Network and Information Security, **2023, 2, 48-61**
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.02.04

Detecting Remote Access Network Attacks Using Supervised Machine Learning Methods

Samuel Ndichu
Maseno University, School of Computing and Informatics, Private Bag, Maseno, Kenya
E-mail: ndichu.ranji@gmail.com
ORCID iD: https://orcid.org/0000-0001-9632-2407
*Corresponding Author

Sylvester McOyowo
Maseno University, School of Computing and Informatics, Private Bag, Maseno, Kenya
E-mail: oyowosilver@gmail.com
ORCID iD: https://orcid.org/0000-0002-0183-0140

Henry Okoyo
Maseno University, School of Computing and Informatics, Private Bag, Maseno, Kenya
E-mail: okoyo.ho@gmail.com
ORCID iD: https://orcid.org/0000-0002-2669-7526

Cyrus Wekesa
University of Eldoret, School of Engineering, Eldoret, Kenya
E-mail: cyrus.wekesa@gmail.com
ORCID iD: https://orcid.org/0000-0001-8827-1005

Received: 04 March 2022; Revised: 07 July 2022; Accepted: 14 September 2022; Published: 08 April 2023

Abstract: Remote access technologies encrypt data to enforce policies and ensure protection. Attackers leverage such techniques to launch carefully crafted evasion attacks introducing malware and other unwanted traffic to the internal network. Traditional security controls such as anti-virus software, firewall, and intrusion detection systems (IDS) decrypt network traffic and employ signature and heuristic-based approaches for malware inspection. In the past, machine learning (ML) approaches have been proposed for specific malware detection and traffic type characterization. However, decryption introduces computational overheads and dilutes the privacy goal of encryption. The ML approaches employ limited features and are not objectively developed for remote access security. This paper presents a novel ML-based approach to encrypted remote access attack detection using a weighted random forest (W-RF) algorithm. Key features are determined using feature importance scores. Class weighing is used to address the imbalanced data distribution problem common in remote access network traffic where attacks comprise only a small proportion of network traffic. Results obtained during the evaluation of the approach on benign virtual private network (VPN) and attack network traffic datasets that comprise verified normal hosts and common attacks in real-world network traffic are presented. With recall and precision of 100%, the approach demonstrates effective performance. The results for k-fold cross-validation and receiver operating characteristic (ROC) mean area under the curve (AUC) demonstrate that the approach effectively detects attacks in encrypted remote access network traffic, successfully averting attackers and network intrusions.

Index Terms: Remote Access, Virtual Private Network, Encrypted Network Traffic, Network Attacks, Machine Learning.

1. Introduction

Remote access has varied advantages. For this reason, it is a popular technology and method for many applications and services. With the advent of internet of things (IoT), internet of everything (IoE), and cloud services, most organizations and individuals directly or indirectly employ remote access for data, services, and applications. The security of data and applications is an integral part of any technology. During remote access, it is important to ensure confidentiality, integrity, and availability [1] of data from the remote access point to the local resource or service. Countless conventional controls are in place to ensure data security during remote access. The security controls, including

anti-virus software, firewall, and intrusion detection systems (IDS), employ signature and heuristic-based approaches [2] for malware detection. The security controls are no match for remote access technologies like tunneling, where data is encrypted to enforce policies and ensure protection.

Even though data encryption during remote access is majorly for ensuring security, attackers leverage the same technology, which is meant to protect data, and circumvent or evade detection by perimeter security controls. The evasion techniques can introduce malware and other unwanted traffic to the internal network, such as distributed denial-of-service (DDoS) traffic. In the face of encrypted data, conventional controls yield a high rate of false positives and false negatives [3], exposing the internal network to many breaches. Other traditional controls, such as the firewall, used to analyze and scan malware in encrypted traffic do this by first decrypting the encrypted traffic for malware inspection. Then, traffic is allowed into the internal network if found free from malware and other unwanted elements. The decryption approach effectively detects malware and compromises present in encrypted traffic, but the approach has limitations: the decryption process introduces computational overheads and dilutes the primary goal of encryption to maintain data privacy.

To this end, several approaches using machine learning (ML) [4] to detect encrypted traffic have been proposed. Cha and Kim [5] differentiate between unencrypted and encrypted traffic. Other approaches have been proposed to detect encrypted traffic, focusing on tasks such as classification and characterization of virtual private network (VPN) traffic and non-VPN one [6], based on services such as secure shell (SSH), Skype, and torrents [7,8], and general malware detection in encrypted traffic [9]. The approaches have achieved good results in classifying and detecting malware in encrypted traffic. However, there are some drawbacks and limitations in that the features are manually selected and limited, can only detect specific malware and traffic types, and none of the approaches is objectively developed for remote access security.

The objective of this study is to develop and evaluate a remote access network traffic attack detection framework. This study, therefore, proposes a ML based approach to the detection of attacks in encrypted traffic during remote access using a weighted random forest (W-RF) [4] algorithm. ML algorithms learn from data and make predictions with high accuracy and less time overhead. The proposed approach employs key statistical network traffic features generated from encrypted traffic packet captures (PCAPs) without decrypting the traffic to counter the weaknesses. The approach preserves data privacy in encrypted traffic by passively inspecting for compromises with a high degree of accuracy. Generally, network traffic classification has two categories [6]; flow-based using flow bytes per second or duration per-flow and packet-based using the size and inter-packet duration. This study focuses on detecting attacks in encrypted remote access network traffic using a combination of flow-based and packet-based features.

This paper comprises four major sections: Section 2 discusses related work; Section 3 describes the methodology used in the study; Section 4 describes the experimental setup and presents the results obtained; and Section 5 presents conclusions based on the analysis of the results.

2. Related Works

The persistence of attackers and constant security breaches have necessitated advanced technologies and methods such as encryption to secure data in transit and storage. Equally, attackers have not been left behind in adopting such advanced technologies to evade detection by perimeter security controls launching carefully crafted evasion attacks. ML has gained popularity in many areas such as classification, speech recognition, and data mining, where algorithms learn from data and make accurate classification and predictions. ML has also been applied to secure networks and systems in e-mail filtering [10], anomaly detection [11], and malware detection and classification applications [12].

Recently, several approaches adopting ML have been proposed to detect encrypted traffic. Cha and Kim [5] employ ML to classify encrypted and unencrypted packets to aid IDS to avoid unnecessary computations. Attempts have been made to classify encrypted traffic services such as SSH, Skype, and bit torrent based on extracted flow-based [7] and packet-based features [8]. Draper-Gil *et. al.*, [6] uses time-related features to characterize encrypted traffic. The approaches achieve a high degree of accuracy in the classification and characterization of encrypted traffic and packets, but they do not look into the maliciousness of such packets.

Anderson and McGrew [9] use limited features and payload examination for encrypted malware traffic classification. The features used include; transport layer security (TLS) handshake metadata, domain name system (DNS) contextual flows linked to encrypted data, and hypertext transfer protocol (HTTP) headers of HTTP contextual flows. Another approach has been proposed using behavioral features, where data logs are generated from PCAP files using an IDS, and a set of thirty features is extracted from the data logs [13].

These ML-based approaches achieve good results for malware and malicious encrypted traffic detection. However, they use manually crafted and engineered features for traffic classification, and the features are limited to detecting only defined encrypted traffic characteristics, malware, and categories. The approaches would yield a high rate of false positives and false negatives when faced with undefined or variants of malicious encrypted traffic characteristics or properties, particularly for remote access encrypted traffic. Therefore, this study proposes an approach that generalizes various attack types and traffic categories used in remote access.

3. Methodology

This section describes the proposed methodology for detecting attacks in encrypted remote access network traffic. Fig. 1. presents the remote access network traffic attack detection framework. The framework comprises four main phases; remote access network traffic, feature generation, feature importance determination, and ML algorithm for prediction and classifying encrypted remote access network traffic into benign virtual private network (VPN) and attack classes. These phases are detailed next.

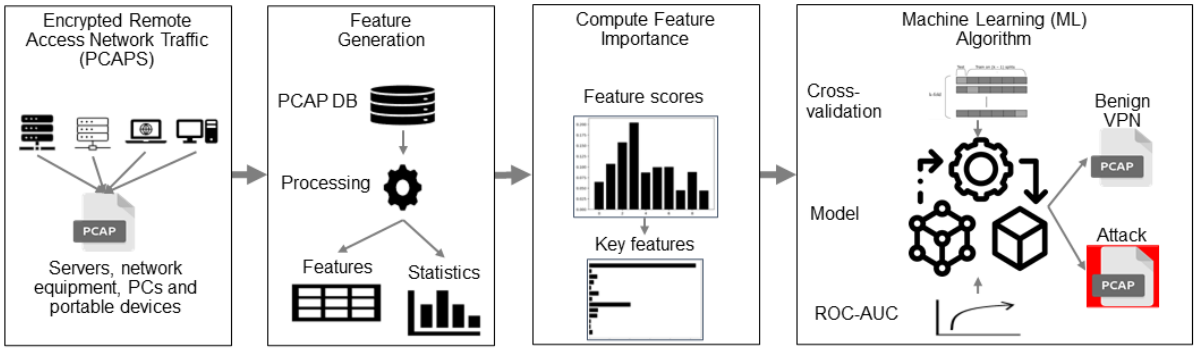

Fig.1. Remote access network traffic attack detection framework.

3.1. Remote Access Network Traffic

This phase contains remote access network traffic. It is important to consider the traffic types in a real remote access network environment, such as browsing, chat, streaming services, and e-mail, to create a representative remote access network traffic dataset that simulates an actual network environment. Therefore, the remote access network traffic dataset should include different traffic types. In addition, since the attacks perpetrated towards the remote access networks are varied, it is logical to develop a method that detects different attack types. This approach will enhance the generalization performance. The common network attacks include:

- Denial-of-service (DoS) – is an attack meant to shut down or disrupt service, machine, or network, making it unavailable. The attack is achieved by flooding the target resource or infrastructure with Internet traffic or transmitting information that triggers a crash.
- A distributed denial-of-service (DDoS) – is a DoS attack that uses multiple computers or machines to flood a targeted infrastructure or resource.
- Port Scan – is a technique used to discover vulnerabilities, open ports, weak points, or back doors in a network. Attackers can also probe for the availability of active security devices such as firewalls in a network.
- Bot attack – leverages remotely controlled malware-infected devices to launch automated web requests to a website, application, application programming interface (API), or end-users for manipulation or service disruption.
- Brute Force – is a hacking method that comprises trial and error to crack login credentials, passphrases, passwords, or encryption keys using automated methods.
- Cross-Site Scripting (XSS) – is an attack vector that utilizes web application vulnerabilities to inject malicious code into benign and legitimate websites.
- SQL Injection (SQLi) – is a web security vulnerability that facilitates the execution of malicious SQL code or statements, allowing an attacker to interfere with the queries an application makes to its database for backend database manipulation.

3.2. Network Traffic Feature Generation

The CICFlowMeter [14,15], an application used for network traffic flow generation and analysis, is adopted to generate network traffic features from attack and benign VPN PCAP files. CICFlowMeter generates bidirectional flows, where the first packet determines the forward and backward directions; source to destination and destination to source, respectively. The application can calculate over eighty statistical network traffic features, including flow duration, the average time between two flows, maximum flow length, the average size of a packet, and the number of packets with; FIN, SYN, RST, ACK, among others. The application outputs a comma-separated values (CSV) file with columns representing the features for network traffic analysis. This application uses FIN packet and flow timeout to terminate transmission control protocol (TCP) flows, and user datagram protocol (UDP) flows.

3.3. Feature Importance Determination

The network traffic features calculated by the CICFlowMeter result in a high-dimensional dataset. Therefore, it is

crucial to verify the usefulness of each feature. The feature importance method from RF computes the average node impurity decrease from all decision trees (DT) in a forest [16]. In other words, the methods assign a score to network traffic features based on their value at predicting the feature label; attack or benign VPN. A RF algorithm is fitted to compute network traffic feature importance to determine relevant features for attack detection in encrypted remote access network traffic. To compute importances over a set of network traffic features, the importance of a node j in a single DT is computed as:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (1)$$

where ni_j is the importance of node j, w_j is the weighted number of feature samples reaching node j as a fraction of the total weighted number of samples, C_j is the impurity value of node j, $left(j)$ is the child node from left split on node j, and $right(j)$ is the right split's child node on node j.

Then feature importance for feature i is then computed as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{j \in all\ nodes} ni_j} \quad (2)$$

where fi_i is the importance for feature i, and ni_j is the importance of node j. Feature importance is computed by averaging all tree's feature importances.

3.4. Machine Learning Algorithms

For the classification task of the encrypted remote access network traffic, using the selected network traffic features, five ML algorithms are used in the experiments [17,18], i.e., naive Bayes (NB) [19], linear regression (LR) [20], support vector machine (SVM) [21], k-nearest neighbors (k-NN) [22], and RF. The best algorithm is selected based on the evaluation results for k-fold cross-validation and receiver operating characteristic (ROC) area under the curve (AUC).

NB is a generative or informative statistical algorithm. The algorithm estimates joint probability from the training data for a given feature x and the label y. The algorithm converges quickly hence needs less training data and can learn interaction between features. NB assumes features are conditionally independent, hence poor performance for dependent features. Given a class, NB assumes the conditional probability of a feature is independent of the conditional probabilities of other features in that class. Therefore, it does not use feature combinations for prediction [23]. The algorithm fits feature weights independently and works best with less training data. Gaussian NB is used to classify network traffic as traffic flows generate negative values, and multinomial NB fails to classify negative values.

LR is a discriminative classifier algorithm that estimates the probability of y or x directly from the training data by minimizing error. The algorithm splits feature space linearly and produces good performance even for correlated variables but may overfit on a small dataset. LR accounts for correlation among features and models some event's probability of occurring as a linear function of a set of predictor variables [23].

SVM is a discriminative algorithm that transforms features to higher dimensions using nonlinear mapping. The algorithm uses support vectors and margins to find the hyperplane, a decision boundary separating features in different classes. An optimal hyperplane divides the training features into respective classes without committing any misclassification errors. SVMs can model complex nonlinear decision boundaries, although the algorithm is slow on training and testing. SVMs are less likely to overfit when compared to other algorithms [24].

k-NN is a supervised machine learning algorithm that finds the distances between a query and data samples. The algorithm selects the particular number of examples K closest to the query then votes for the most frequent label. As the size of data increases, the algorithm tends to be slow.

RF is a supervised algorithm that comprises an ensemble of DTs. It fits or builds several DTs on various data subsamples during training. Predictions from all trees are averaged to make the final prediction using the mode of the classes for classification. RF is trained using the bagging method, and averaging improves predictive accuracy and controls overfitting. Algorithm 1. presents a RF pseudocode.

3.5. Weighted Machine Learning Algorithms

RF curbs overfitting in DT and results in improved accuracy for most classification tasks. Therefore, this study proposes using the RF algorithm to detect encrypted remote access network traffic attacks. The other algorithms, NB, LR, SVM, and k-NN, are used for performance comparison. We use a weighted RF algorithm with the inverse of the class distribution to account for the unbalanced data distribution common in remote access network traffic. The *class_weight* parameter, a dictionary defining each class weight in the form {*class_label: weight*}, is used to specify the weights. It is envisioned that RF with class-weight values would perform better than the other algorithms.

Algorithm 1: Pseudocode for random forest (RF) algorithm

To generate c classifiers:
for $i = 1$ to c **do**
    Randomly sample the training encrypted remote access network traffic D with replacement to produce D_i
    Create a root node N_i containing D_i
    Call BuildTree(N_i)
end for

BuildTree(N):
if N contains instances of only one class, **then**
    return
else
    Randomly select x% of the possible splitting network traffic features in N
    Select the traffic feature F with the highest information gain to split on
    Create f child nodes of $N, N_1, \ldots, N_f$, where F has f possible values $(F_1, \ldots, F_f)$
    for $i = 1$ to f **do**
        Set the contents of N_i to D_i, where D_i are all instances in N that match F_i
        Call BuildTree(N_i)
    end for
end if

4. Experiments

This section presents the dataset, preprocessing, data distribution visualization, evaluation metrics, and analysis and discussion of results.

4.1. Dataset

The dataset used in the experiments contains network traffic PCAPs obtained from two different sources. The encrypted network attack traffic was obtained from CICIDS2017 [25], an intrusion detection evaluation dataset that comprises common attacks in actual network traffic. The encrypted benign VPN network traffic was obtained from ISCXVPN2016 [6], a VPN-nonVPN dataset comprising verified normal hosts and representing real-world remote access network traffic. We combine the two datasets to create a customized remote access dataset with remote access network characteristics and attack features.

4.2. Dataset Preprocessing

The encrypted attack and benign VPN network traffic were analyzed using CICFlowMeter to generate CSV files of labeled flows based on the time stamp, source, and destination internet protocols (IP), source and destination ports, protocols, and attack.

The encrypted benign VPN network traffic comprises eighteen traffic categories, as shown in Table 1.

The traffic ranges from traffic content, web browsing, e-mail, chat, streaming, file transfer, voice over internet protocol (VoIP), and peer-to-peer (P2P). Hangout's audio and chat categories make up the majority of the benign VPN traffic.

The encrypted attack network traffic comprises fourteen common attack types, as shown in Table 2. The attacks range among brute force file transfer protocol (FTP), brute force SSH, denial-of-service (DoS), Heartbleed, web attack, infiltration, botnet, and DDoS. Generally, traffic data in a remote access network [26] contains more benign samples than the attack ones. Therefore, to simulate an actual network environment, we sample each attack type so that the experiment dataset has more benign samples than the attack ones. This process results in 5,568 attack samples and 23,225 benign samples and is the dataset used in the experiments, making attack samples 19.338% of the total dataset.

Using CICFlowMeter, we generated eighty-three network traffic features. For each feature, we compute feature importance obtained from a fitted RF attribute *feature_importances_* that gives the relative importance scores for each input encrypted network traffic feature. Feature importance facilitates the determination of key features for encrypted remote access attack detection and dimensionality reduction. We select twenty-six features with a feature importance score of and above 0.005, as shown in Fig. 2. The features are described in Table 3.

Table 1. Statistics of benign VPN network traffic.

No.	Benign VPN traffic	#Traffic flows
1	Hangouts audio	9,332
2	Hangouts chat	2,904
3	Skype audio	1,961
4	Facebook audio	1,568
5	Skype files	1,507
6	Facebook chat	1,230
7	Voipbuster	1,127
8	Spotify	597
9	Netflix	577
10	Youtube	553
11	Bittorrent	487
12	Vimeo	453
13	E-mail	331
14	File transfer protocol secure (FTPS)	171
15	Skype chat	165
16	AOL instant messenger (AIM) chat	100
17	I seek you (ICQ) chat	96
18	Secure file transfer protocol (SFTP)	66
	Total	23,225

Table 2. Statistics of attack network traffic.

No.	Attack type	#Unique attacks	#Sampled attacks
1	DoS Hulk	172,849	500
2	DDoS	128,016	500
3	PortScan	90,819	500
4	DoS GoldenEye	10,286	500
5	FTP-Patator	5,933	500
6	DoS slowloris	5,385	500
7	DoS Slowhttptest	5,228	500
8	SSH-Patator	3,219	500
9	Bot	1,953	500
10	Web Attack - Brute Force	1,470	500
11	Web Attack - XSS	652	500
12	Infiltration	36	36
13	Web Attack - SQL Injection	21	21
14	Heartbleed	11	11
	Total	425,878	5,568

4.3. Data Distribution Visualization

To examine the distribution of the benign VPN and attack network traffic dataset, we used the principal component analysis (PCA) [27] for network traffic data visualization in a two-dimensional space. PCA is a method for data dimensionality reduction, which adds interpretability and minimizes information loss. The method preserves as much variability or statistical information as possible [28].

A PCA projection of benign VPN and attack encrypted network traffic data is shown in Fig. 3. Label '0' in black represents the benign VPN traffic, and '1' in bright brown represents the attack traffic. The benign VPN traffic is tightly clustered in the same embedding space, whereas the attack traffic is clustered across the space. The projection demonstrates that the benign VPN traffic is separable from the attack one and implies that the ML algorithms are likely to achieve a high classification performance on the dataset. In addition, we also analyze the attack data separately, as shown in Fig. 4. Different colors in the embedding space represent the attack traffic types. The attack data is projected in the same embedding space except for the boat traffic, and similar attack types form tight clusters.

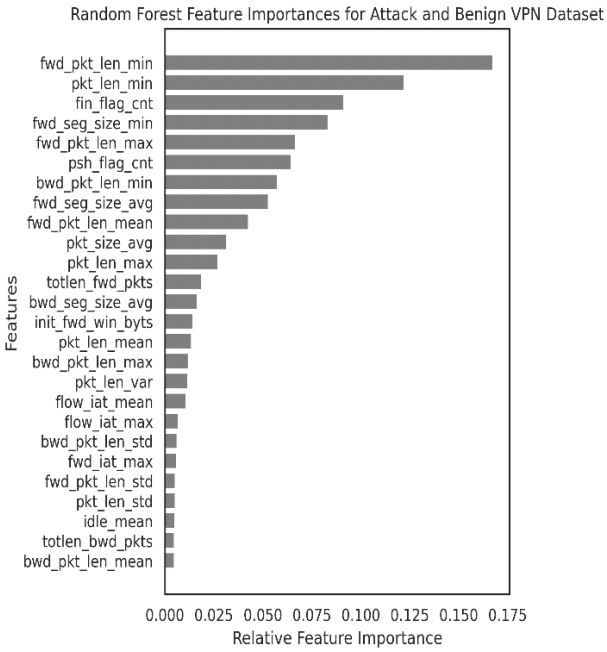

Fig.2. Remote access network traffic feature importance.

Table 3. Description of remote access network traffic attack detection features.

No.	Feature name	Description
1	fwd_pkt_len_min	Minimum size of the packet in the forward direction
2	pkt_len_min	Minimum length of a flow
3	fin_flag_cnt	Number of packets with FIN
4	fwd_seg_size_min	Minimum segment size observed in the forward direction
5	fwd_pkt_len_max	Maximum size of the packet in the forward direction
6	psh_flag_cnt	Number of packets with PUSH
7	bwd_pkt_len_min	Minimum size of the packet in the backward direction
8	fwd_seg_size_avg	Average size observed in the forward direction
9	fwd_pkt_len_mean	Mean size of the packet in the forward direction
10	pkt_size_avg	The average size of the packet
11	pkt_len_max	The maximum length of a flow
12	totlen_fwd_pkts	The total size of the packet in the forward direction
13	bwd_seg_size_avg	Average size observed in the backward direction
14	init_fwd_win_byts	Number of bytes sent in the initial window in the forward direction
15	pkt_len_mean	Mean length of a flow
16	bwd_pkt_len_max	Maximum size of the packet in the backward direction
17	pkt_len_var	Minimum inter-arrival time of packet
18	flow_iat_mean	The average time between two flows
19	flow_iat_max	Maximum time between two flows
20	bwd_pkt_len_std	Standard deviation size of the packet in the backward direction
21	fwd_iat_max	Maximum time between two packets sent in the forward direction
22	fwd_pkt_len_std	Standard deviation size of the packet in the forward direction
23	pkt_len_std	Standard deviation length of a flow
24	idle_mean	The mean time a flow was idle before becoming active
25	totlen_bwd_pkts	The total size of the packet in the backward direction
26	bwd_pkt_len_mean	Mean size of the packet in the backward direction

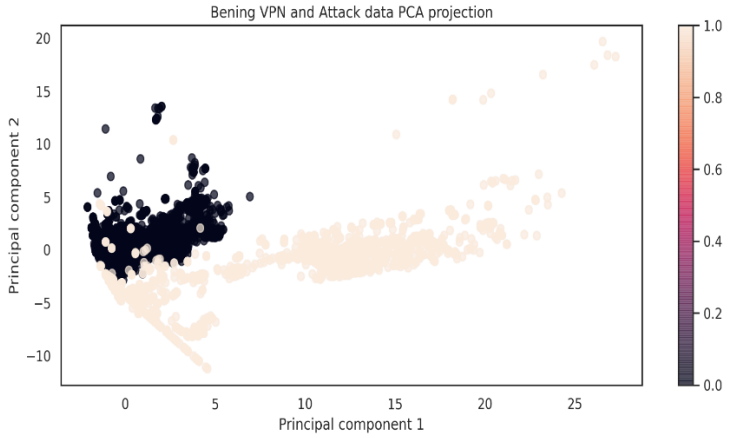

Fig.3. Remote access network traffic visualization using principal component analysis (PCA).

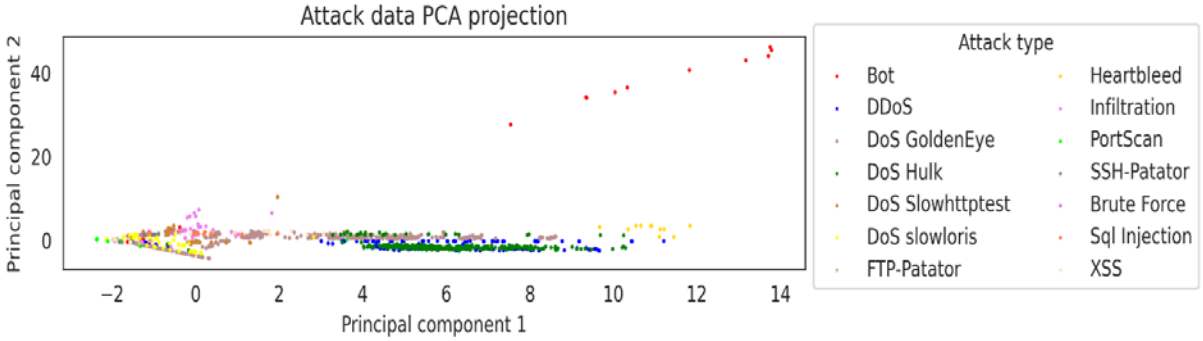

Fig.4. Network attack traffic visualization using principal component analysis (PCA).

4.4. Evaluation Metrics

Several ML evaluation metrics were used to evaluate the performance of the algorithms on detection of attacks in encrypted remote access network traffic; accuracy, precision, recall, and f1-score. In the experiments, true positive (TP) represents the attack traffic correctly classified as an attack, true negative (TN) represents the benign VPN traffic correctly classified as benign, false positive (FP) represent the benign VPN traffic misclassified as an attack, and false-negative (FN) represents the attack traffic misclassified as benign.

Accuracy is derived from the proportion of correctly classified network traffic,

$$Accuracy = \frac{TP+TN}{P+N} \tag{3}$$

where P and N represent the positive (attack) and negative (benign VPN) traffic, respectively.

Precision is the proportion of correctly classified positive traffic,

$$recision = \frac{TP}{TP+FP} \tag{4}$$

Recall is the proportion of correctly classified attack traffic,

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

F1-score can be described as the harmonic mean of precision and recall given by,

$$F1\text{—}score = 2\frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

True and false-positive rates (TPR and FPR) were plotted using ROC graphs and computed mean AUC to visualize the algorithm's performance. K-fold cross-validation with $k = 10$ was used in all the experiments. Cross-validation eliminates biases of overfitting or under-fitting, resulting in a generalizable algorithm. During cross-validation, network traffic D is divided into ten folds, $d_1, d_2, \ldots, d_{10}$. For ten iterations, nine folds $D - d_k$ are used for algorithm training and one-fold d_k for performance evaluation, and estimated metric E_i is computed for each iteration. The estimated

performance E_i for each fold is then used to compute the estimated average performance E for each algorithm, as shown in Algorithm 2.

Algorithm 2: 10-fold cross-validation for remote access network traffic

Input: Encrypted remote access network traffic $D = d_1, d_2, \dots, d_{10}$

Output: Evaluation metric E_i for each iteration and average evaluation metric E for all iterations.

1: **for** $k = 1$ to 10 **do**
2: $Train = D - d_k$
3: $Test = d_k$
4: Classifier = *clf*
5: clf.$Train(D - d_k)$
6: clf.$Test(d_k)$
7: Compute metric E_i for each iteration.
8: **end for**
9: Compute average metric E for the ten iterations,

$$E = \frac{1}{10}\sum_{i=1}^{10} E_i$$

4.5. Results

We conducted a performance comparison for the five algorithms using the preprocessed network traffic data. Accuracy, precision, recall, and f1-score obtained by 10-fold cross-validation are computed for the five algorithms, as shown in Table 4.

Table 4. Performance comparison.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NB	83.267 (±0.0057)	61.100 (±0.0252)	37.068 (±0.0237)	46.115 (±0.0234)
SVM	92.470 (±0.0042)	86.539 (±0.0122)	72.323 (±0.0197)	78.779 (±0.0136)
LR	84.774 (±0.0585)	57.726 (±0.0733)	93.121 (±0.0100)	70.991 (±0.0647)
k-NN	99.309 (±0.0012)	98.034 (±0.0046)	98.402 (±0.0046)	98.216 (±0.0031)
RF	**99.910** (±0.0005)	**100.000** (±0.0000)	**99.533** (±0.0028)	**99.766** (±0.0014)

The best performing algorithms are k-NN and RF, with over 98.034% for all the four-evaluation metrics. RF outperforms all the other algorithms in the four metrics with 99.910% accuracy, 100% precision, 99.533% recall, and 99.766% f1-score. These scores demonstrate the RF ability to detect attacks in encrypted remote access network traffic effectively. However, the performance in recall metric demonstrates that 62.932%, 27.677%, 6.879%, 1.598%, and 0.467% of attacks in encrypted remote access network traffic would go unnoticed for NB, SVM, LR, k-NN, and RF, respectively.

A missed attack translates to undesired effects such as network disruption, loss of financial information, and reputation damage. On the other hand, misclassified benign VPN traffic leads to annoying false alarms that unnecessarily consume the security analysts' time to investigate them. Therefore, an effective detection system for attacks in encrypted remote access network traffic would have as few as possible false negatives and false positives. We conduct experiments with weighted versions of the four algorithms to achieve such a system. NB is not used for experiments with weighted algorithms as it does not have a *class_weight* parameter.

The network traffic dataset used in the experiments has a class distribution of 19.338:80.662. Therefore, we assign weights as the inverse of class distribution. For benign VPN traffic class, which comprises the majority traffic, a weight of 19.338 is used, and for the attack traffic class, representing the minority traffic, a weight of 80.662 is used in the form {*0:19.338, 1:80.662*}, where '0' and '1' represents benign VPN and attack labels respectively. The class weights make the penalty for wrong prediction of attack traffic class 80.662 times more severe than a wrong prediction of benign VPN traffic class. This dictionary is used for LR, SVM, and RF algorithms. K-NN is assigned '*uniform'* as the weight parameter.

Table 5. shows the accuracy, precision, recall, and f1-score obtained using cross-validation for the four algorithms with a weighting parameter. The table shows that the recall performance for the four algorithms is improved, indicating an improvement in the detection of attacks in remote access network traffic for the four algorithms with 4.221%, 1.868%, 0.718%, and 0.467% for W-LR, W-SVM, W-k-NN, and W-RF respectively. However, the precision for W-LR, W-SVM,

and W-k-NN decline, indicating that the three algorithms would yield many false alarms leading to time wastage by security analysts tasked with investigating such alarms. The best two algorithms are W-k-NN and W-RF. W-RF outperforms W-k-NN in all metrics with 0.382% accuracy, 1.09% precision, 0.88% recall, and 0.986% f1-score. With a score of 100% in all the metrics, W-RF effectively detects attacks in encrypted remote access network traffic, successfully averting attackers' ill intentions such as network disruption, loss of financial information, and reputation damage. In addition, the algorithm yields zero false alarms indicating that the security analysts can concentrate on other aspects of network security. Therefore, this approach gives promising results to avert remote access network traffic attacks successfully.

Table 5. Performance comparison with weighted algorithms.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
W-LR	40.319 (±0.0062)	24.137 (±0.0017)	97.342 (±0.0062)	38.682 (±0.0022)
W-SVM	92.391 (±0.0039)	84.567 (±0.0096)	74.191 (±0.0180)	79.030 (±0.0124)
W-k-NN	99.618 (±0.0009)	98.910 (±0.0047)	99.120 (±0.0035)	99.014 (±0.0023)
W-RF	**100.000** (±0.0000)	**100.000** (±0.0000)	**100.000** (±0.0000)	**100.000** (±0.0000)

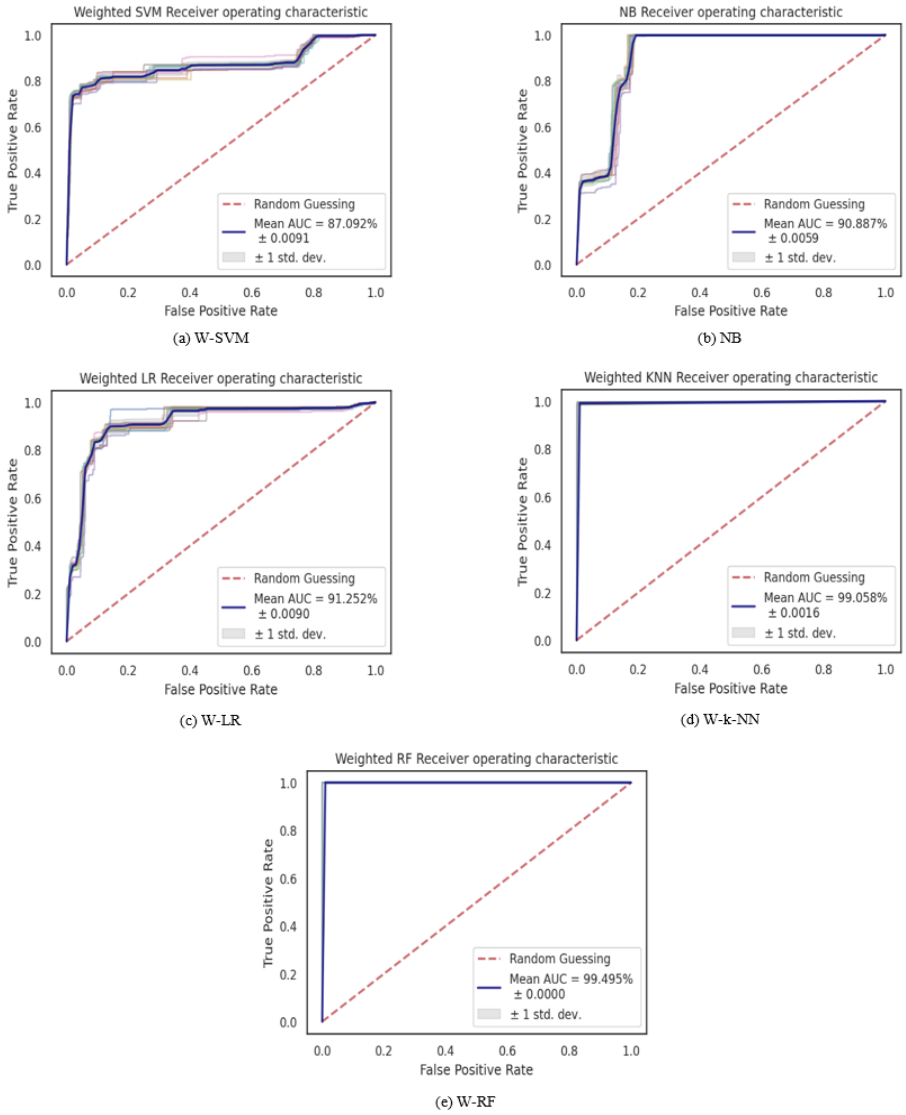

Fig.5. Receiver Operating Characteristic.

Fig. 5. shows the ROC graphs and mean AUC for each algorithm obtained using 10-fold cross-validation where AUC for each iteration is computed. The dashed red line indicates the threshold for the algorithm's performance. A perfect algorithm would have a TPR of 1 and an FPR of 0 and vice versa. All the algorithms perform above the threshold

for attack detection in encrypted remote access network traffic, yielding a mean AUC of 87.092%, 90.887%, 91.252%, 99.058%, and 99.495% for W-SVM (a), NB (b), W-LR (c), W-k-NN (d) and W-RF (e) respectively. As demonstrated using the other evaluation metrics, W-RF is the best performing algorithm achieving the highest AUC, affirming its effectiveness in detecting attacks in encrypted remote access network traffic.

We perform parameter tuning using GridSearchCV [17,18], a library function that performs an exhaustive search over specified parameter values for algorithms selection. It loops through given parameters and fits the algorithms on encrypted remote access network traffic, selecting the best parameters for detecting attacks in encrypted remote access network traffic. The selected parameters for k-NN are *leaf_size*=10, *n_neighbors*= 1, and *p*=1. The selected parameters for RF are *criterion*='entropy', *max_depth*=8, *min_samples_leaf*=1, *min_samples_split*=0.05. We found these parameters optimal for the two algorithms to detect attacks in encrypted remote access network traffic.

4.6. Performance Comparison

This section presents the performance comparison of the proposed approach to related approaches for detecting network attacks. Network traffic, attack type, and machine learning algorithms are presented. Table 6. shows the performance comparison. We did not find any related work on detecting attacks in remote access network traffic in the literature surveyed. We, therefore, make a comparison with works analyzing network traffic. The comparison is sufficient to evaluate the obtained results even though the experimental scenarios and datasets differed.

Table 6. Performance comparison with related approaches.

Network Traffic	Attack type	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Software-as-a-service (SaaS) security [29]	botnets	Deep Belief Network (DBN) with Median Fitness oriented Sea Lion Optimization algorithm (MFSLnO)	86.000	82.000	82.000	-
General network security [30]	fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms	1D-Convolutional Neural Network (1D-CNN) and Feed Forward Network	98.990	98.130	100.000	98.990
General network security [31]	DoS/DDoS	Random forest (RF)	99.936	99.900	96.500	99.900
Source Side in Cloud security [32]	DDoS	SVM Linear Kernel	99.730	99.940	99.560	99.750
General network security [33]	DoS, Remote to Local Attack (R2L), User to Root Attack (U2R), Probing Attack (Probe)	Fuzzy C-Means	82.100	-	84.600	-
Proposed approach - remote access network security	14 Attack types [Table 2]	Weighted Random Forest (W-RF)	100.000 (±0.0000)	100.000 (±0.0000)	100.000 (±0.0000)	100.000 (±0.0000)

The works [29-32] employ supervised machine learning algorithms, whereas [33] use unsupervised ones. Some researchers concentrate on detecting one network attack, botnets [29] and DDoS [31,32]. In contrast, the proposed approach detects fourteen network attack types enhancing generalization performance. Francisco *et. al.* [31] employs an RF algorithm for attack detection, the same as the proposed approach. However, they do not consider the imbalanced nature of network traffic data. Approaches using RF [31] and SVM linear kernel [32] perform better than the rest. However, the proposed approach with W-RF outperforms all related approaches in all the evaluation metrics. The unsupervised approach [33] presents the worst performance. The results indicate that supervised machine learning methods are better suited for detecting network traffic attacks, specifically remote access network traffic attacks.

5. Conclusions

This study presented an approach to detect attacks in encrypted remote access network traffic. The proposed approach uses CICFlowMeter to generate network traffic features from attack and benign VPN PCAP files. Then, a RF algorithm is fitted to compute network traffic feature importance to determine key features for attack detection in encrypted remote access network traffic. Finally, to account for the unbalanced data distribution common in remote access network traffic, a weighted RF with *class_weight* parameter of the inverse of the class distribution is used to classify encrypted remote access network traffic into benign VPN and attack classes. The results for k-fold cross-validation and ROC mean AUC demonstrate that the proposed remote access network traffic attack detection framework effectively detects attacks in encrypted remote access network traffic, successfully averting attackers' ill intentions such as network disruption, loss of financial information, and reputation damage. Therefore, the proposed approach can be

employed for a comprehensive security solution for remote access networks. Other approaches to handle imbalanced class distribution in encrypted remote access network traffic, such as oversampling and undersampling methods, can be evaluated in future work.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

Acknowledgments

The research was achieved and supported in part by the Higher Education Loans Board, Kenya (HELB, Kenya) through Scholarship HELB/45/003/VL.1 of Postgraduate Award 2014/2015.

References

[1] Yan, F., Jian-Wen, Y. and Lin, C. (2015). Computer Network Security and Technology Research, Seventh International Conference on Measuring Technology and Mechatronics Automation, PP. 293-296, DOI: 10.1109/ICMTMA.2015.77.

[2] Ndichu, S., McOyowo, S. and Wekesa, C. (2016). A Review of Security Vulnerabilities, Controls and Models in Networked Environments, *International Journal of Latest Research in Engineering and Technology (IJLRET)*, ISSN: 2454-503, Volume 02, Issue 08, August 2016, PP. 06-14.

[3] Stefan, A. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information System Security*, Volume 3, Issue 3 (Aug. 2000), PP. 186–205, DOI: https://doi.org/10.1145/357830.357849.

[4] Breiman, L. (2001). Random Forests, *Machine Learning*, Volume 45, PP. 5–32 (2001). https://doi.org/10.1023/A:1010933404324.

[5] Cha, S. and Kim, H. (2016). Detecting Encrypted Traffic: A Machine Learning Approach, *International Workshop on Information Security Applications*, WISA 2016. PP. 54-65.

[6] Draper-Gil, G., Lashkari, A., Mamun, M. and Ghorbani, A. (2016). Characterization of Encrypted and VPN Traffic Using Time-related Features, *In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP'16)*, PP. 407-414. http://www.unb.ca/cic/datasets/vpn.html.

[7] Alshammari, R. and Zincir-Heywood A. N. (2009). Machine Learning-Based Encrypted Traffic Classification: Identifying SSH and Skype, *Computational Intelligence for Security and Defense Applications (CISDA)*, IEEE.

[8] Tabatabaei, T. S., Adel, M., Karray, F. and Kamel, M. (2012). Machine Learning-Based Classification of Encrypted Internet Traffic, *In: Perner P. (eds) Machine Learning and Data Mining in Pattern Recognition, MLDM 2012, Lecture Notes in Computer Science*, Volume 7376, Springer, Berlin, Heidelberg, PP. 578-592, https://doi.org/10.1007/978-3-642-31537-4_45.

[9] Anderson, B. and McGrew, D. (2017). Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-Stationarity, *KDD'17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, PP. 1723-1732.

[10] Emmanuel, G. D., Joseph, S. B., Haruna, C., Shafi'i M. A., Adebayo, O. A. and Opeyemi E. A. (2019). Machine learning for e-mail spam filtering: review, approaches and open research problems, *Heliyon*, Volume 5, Issue 6, 2019, e01802, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2019.e01802, https://www.sciencedirect.com/science/article/pii/S2405844018353404.

[11] Nassif, A. B., Talib, M. A., Nasir Q. and Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review, *in IEEE Access*, Volume. 9, PP. 78658-78700, DOI: 10.1109/ACCESS.2021.3083060.

[12] Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L. (2009). Malware detection using machine learning, *International Multiconference on Computer Science and Information Technology*, PP. 735-741, DOI: 10.1109/IMCSIT.2009.5352759.

[13] Bru con. (2017). Detecting Malware even when it is Encrypted – Machine Learning for Network HTTPS Analysis, *Bru con Security Conference*.

[14] Lashkari, H., Arash. (2018). CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyzer for anomaly detection. https://github.com/ISCX/CICFlowMeter. 10.13140/RG.2.2.13827.20003.

[15] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I. and Ghorbani, A A. (2017). Characterization of Tor Traffic Using Time-Based Features, *In the proceeding of the 3rd International Conference on Information System Security and Privacy*, SCITEPRESS, Porto, Portugal, 2017.

[16] Forest of trees-based ensemble methods. (Accessed October 2021). Those methods include random forests and extremely randomized trees, Compute the importance of each feature, https://github.com/scikit-learn/scikit-learn/blob/0abd95f742efea826df82458458fcbc0f9dafcb2/sklearn/ensemble/forest.py#L360.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A. et. al. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, Volume 12, PP. 2825–2830.

[18] Buitinck, L., Louppe, G., Blondel, M. et. al. (2013). API design for machine learning software: experiences from the scikit-learn project, *In ECMLPKDD Workshop: Languages for Data Mining and Machine Learning*, PP. 108–122.

[19] Zhang H. and Li, D. (2007). Naïve Bayes Text Classifier, *IEEE International Conference on Granular Computing (GRC 2007)*, PP. 708-708, DOI: 10.1109/GrC.2007.40.

[20] Sammut C., Webb G.I. (2011). Logistic Regression, *Encyclopedia of Machine Learning*, *Springer*, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_493.

[21] Cortes, C., and Vapnik, V. (1995). Support-vector networks, *Machine Learning*, Volume 20, Issue 3, PP. 273–297, https://doi.org/10.1007/BF00994018.

[22] Cunningham, P. and Delany, S. J. (2021). K-Nearest Neighbour Classifiers - A Tutorial, *ACM Computing Surveys, Association for Computing Machinery (ACM),* Number 6, Volume 54, PP. 1–25, DOI 10.1145/3459665, http://dx.doi.org/10.1145/3459665.

[23] Entezari-Maleki, R., Rezaei, A. and Minaei-Bidgoli, B. (2009). Comparison of Classification Methods Based on the Type of Attributes and Sample Size, *Journal of Convergence Information Technology*, Volume 4, Number 3, September, PP. 94-102.
[24] Han, J., Kamber, M. and Pei, J. (2011). Data Mining: Concepts and Techniques (3rd. ed.), *Morgan Kaufmann Publishers Inc*., San Francisco, CA, USA.
[25] Sharafaldin, I., Lashkari, A. H. and Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal.
[26] Samuel Ndichu, Sylvester McOyowo, Henry Okoyo, Cyrus Wekesa, "A Remote Access Security Model based on Vulnerability Management", International Journal of Information Technology and Computer Science, Vol.12, No.5, pp.38-51, 2020.
[27] Jolliffe I. (2011). Principal Component Analysis, *In: Lovric M. (eds) International Encyclopedia of Statistical Science*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_455.
[28] Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374: 20150202, http://doi.org/10.1098/rsta.2015.0202.
[29] SaiSindhuTheja, R. and Gopal, K. S. (2020). A machine learning-based attack detection and mitigation using a secure SaaS framework, Journal of King Saud University - Computer and Information Sciences, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2020.10.005.
[30] De Lucia, M. J., Maxwell, P. E., Bastian, N. D., Swami, A., Jalaian, B. and Leslie, N. (2021). Machine learning raw network traffic detection, Proc. SPIE 11746, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III, 117460V, https://doi.org/10.1117/12.2586114.
[31] Francisco, F., Frederico, S., Agostinho, J., Genoveva, v s. and Luiz S. (2019). Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning, Security and Communication Networks, PP. 1-15. 10.1155/2019/1574749.
[32] He, Z., Zhang, T. and Lee, R. B. (2017). Machine Learning-Based DDoS Attack Detection from Source Side in Cloud, IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), PP. 114-120, doi: 10.1109/CSCloud.2017.58.
[33] Kumar, A., Glisson, W. and Hyuk, C. (2020). Network Attack Detection Using an Unsupervised Machine Learning Algorithm, 10.24251/HICSS.2020.795.

Authors' Profiles

Samuel Ndichu is a Ph.D. candidate in the School of Computing and Informatics, Maseno University, Kenya. His current research interests include remote access and computer network security.

Sylvester McOyowo holds a Ph.D. degree in Computer Science from the Peoples' Friendship University. He is the Dean of, School of Computing and Informatics and a lecturer at the Department of Computer Science, Maseno University, Kenya. His main teaching and research interests include Digital Signal Processing, Computer Architecture, Fibre Optics, Digital and Analog Electronics.

Henry Okoyo holds a Ph.D. degree in Computer Science from the University of Manchester, an MSc degree in Microprocessor Engineering and Digital Electronics from the former University of Manchester Institute of Science and Technology (UMIST), and a BSc degree from the University of Nairobi, Kenya. He is a lecturer at the Department of Computer Science, School of Computing and Informatics, Maseno University, Kenya. His main teaching and research interests include artificial intelligence.

Cyrus Wekesa holds a Ph.D. degree in Electrical Engineering from University of Tokushima, Japan, an MSc, and a BSc in Electrical Engineering from University of Nairobi, Kenya. He is an associate professor in the school of Engineering, University of Eldoret, Kenya. His teaching and research interests include telecommunications, computer networks, information security, computer architecture, electronics, and distributed systems.

How to cite this paper: Samuel Ndichu, Sylvester McOyowo, Henry Okoyo, Cyrus Wekesa, "Detecting Remote Access Network Attacks Using Supervised Machine Learning Methods", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.2, pp.48-61, 2023. DOI:10.5815/ijcnis.2023.02.04