Detecting Network Intrusion through Anomalous Packet Identification

Tanjim Munir Dipon [1], Md. Shohrab Hossain[1], Husnu S. Narman[2]
[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[2]Weisberg Division of Computer Science, Marshall University, Huntington, WV, USA
Email: tanjim.dipon@gmail.com, mshohrabhossain@cse.buet.ac.bd, narman@marshall.edu

***Abstract*—Rule based intrusion detection depends on the attack signature database which has to be constantly updated, requiring time and efforts. Anomaly based intrusion detection through unsupervised methods does not require comparing with attack signatures. However, detecting anomalous behaviour is a complex task. In this paper, we have proposed an unsupervised approach for anomalous network traffic identification by combining dimensionality reduction with sub-space clustering. Our approach takes the attribute values from network traffics as input, performs principal component analysis on them, and then applies density-based clustering on each possible three dimensional sub-spaces to rank the outliers. Results show that our proposed approach detects a wide range of anomalous network session which included instances of intrusive sessions too. The evaluation of this approach showed significant accuracy and faster detection with a zero false negative rate, implying that no instance of the listed attacks went undetected.**

***Index Terms*—NIDS, Unsupervised learning, PCA, Density-based clustering.**

I. Introduction

Network based intrusion detection (NIDS) represents detecting intrusive packets or sessions through the analysis of network traffics. In this modern era, all sectors exchange information through the Internet. There have been intrusions and session hijacking related security threats [1] using botnets [2] that are controlled by hackers in remote locations. However, various intruder groups try to disrupt the activities by launching different attacks over the Internet. Hence, the role of Network Intrusion Detection System (NIDS) is crucial to detect these malicious activities and subsequently report those activities to the appropriate administrator for possible mitigation. Such detection is done in two ways: rule based and anomaly based detection. Rule based detection matches network traffic against known attack patterns; hence, it is unable to detect newly crafted attack pattern. Anomaly based detection plays a vital role in detecting newly launched attacks by marking the network traffics as anomalous which are over the given threshold behaviour.

There have been several works for the detection of intrusive traffic in networks. Few of the works [3]–[6] used clustering based on density and distance. Few other works [7], [8] used K-Means clustering. Chen et al. [9] made a comparison between density and K-means clustering. In recent works, Chen et al. used multi scale PCA [10] and four different clustering methods [11] based on voting for anomaly detection. Odiathevar et al. [12] used both supervised and unsupervised approach on labelled data to represent a hybrid anomaly detection model. None of the above mentioned works used dimensionality reduction in combination with clustering which can extract the data from network traffic extensively and finally mark the outliers concerning anomaly count.

The *contributions* of this work are:

- proposing an efficient network anomaly detection technique by combining dimensionality reduction and clustering,
- utilizing all possible features from network traffic for precise detection of a wide range of anomalies,
- accurately detect anomalies by combining the outlier counts from all possible 3-dimensional sub-spaces.

Our proposed model first extracts network attributes from a packet capture file which are recorded inside a session. Next, this data is provided for dimensionality reduction to avoid dependency between attributes and to drop the nonessential attributes. Then, clustering is used in all of the 3-dimensional sub-spaces. Finally, an outlier count is derived for all of the sessions and sessions having the most counts are represented as anomalous, assuming that most of the network sessions are normal and a few sessions will have intrusive behaviours.

Results show that our proposed model has achieved good accuracy, best true positive rate and zero false negative rates, leaving no instance of attack undetected. Our proposed detection method can work in parallel with rule based detection to detect any session (of newly generated attack), resulting in high accuracy and leading to faster intrusion detection.

The rest of the paper is organized as follows. Section II lists the related works. Section III shows the methodology used. Section IV shows the results achieved from our experiment. Finally, Section V has the concluding remarks.

II. Related works and Gap Analysis

There have been a number of works performed for network anomaly detection. Casas et al. [3] applied density-based clustering (DBC) for all of the two-dimensional sub-spaces. Dromard et al. [4] used discrete time sliding window and density-based incremental grid clustering algorithm to detect anomalies in real-time using all possible transport layer features. Zhao et al. [6] used abnormality weight matrix to separate the outliers from the data clusters. Chen et al. [11] used density-based clustering, one-SVM, agglomerative clustering and expectation maximization in parallel. Chen et al. [10] used

maximal information coefficient matrix from the attributes of network data and then PCA was applied on different scales depending on different attributes. Savvas et al. [5] used PCA and density-based clustering for anomaly detection in railway traffic data. However, the majority of these works did not consider which attributes were not contributing to the variations of the data that much. Some works considered necessary attributes but did not accumulate the evidence on anomaly detection from different combinations of the attributes.

Some of the works used K-Means Clustering-based approaches. Chen et al. [9] showed a comparison between k-means clustering and DBC based on system call numbers of the running processes extracted from audit files of the host machine. Turab et al. [13] also used process monitoring. Alom et al. [7] used autoencoder and restricted boltzmann machine algorithm for dimensionality reduction and then used K-Means clustering. Terzi et al. [8] used attributes specific to botnet attacks from netflow data and applied K-Means clustering to cluster the data. K-means clustering could reveal the presence of several clusters in the data set. However, they could not detect the anomalous data points as they are included inside their closest clusters. Odiathevar et al. [12] used both offline and online technique for detection where the offline system worked on labelled data and in case of its failure, the online portion used support vector machine to detect anomalies.

Our work differs from previous works in the following ways:

- We have extracted all possible attributes from different layers of network traffic.
- We have performed dimensionality reduction through PCA before clustering.
- We have performed DBC inside all possible 3-dimensional sub-spaces of the principal components.

III. Proposed System

The proposed system is composed of four fundamental modules.

- Data preprocessing from Network Packets
- Apply Principal Component Analysis (PCA) to the preprocessed data
- Apply density-based clustering (DBC) in all possible 3-dimensional sub-spaces.
- Accumulate anomaly count to detect outliers.

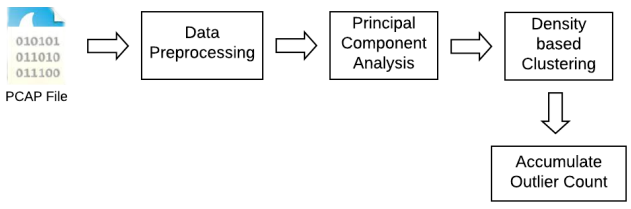

Fig. 1. Overview of the proposed system

Fig. 1 shows the workflow that we used in our proposed system. The detailed description is listed below:

A. Data Preprocessing

Preprocessing is the most fundamental part of our proposed approach. As the network packets belong to different services, thus it is not possible to detect network anomalies by inspecting a few of the attributes of these packets. Works done previously did not use such extensive attribute extraction process from network packets. Extracting attributes from all possible layers leads to the detection of a wide range of anomalies which helps to detect intrusion accurately. Without considerable preprocessing, the unsupervised learning module will not be able to detect the outliers properly. Preprocessing is divided into two portions. The first portion is different for different layers of network traffic. The second portion is the same for all.

1) Network Layer: Services having fields up to network layer (e.g. ICMP) are processed under this module. The extracted fields are: source and destination IP addresses, don't fragment (DF) and more fragment (MF) rate, type of ICMP packet, checksum status of ICMP packet, average data length of ICMP packets in a session.

2) Transport Layer: Only fields related to TCP are extracted in this layer so far. These fields are listed below:

- Source and destination port numbers
- Average TCP segment length from client to server
- Average time to live value of SYN flagged packets
- Percentage of SYN, SYN-ACK, PUSH, URG, RST and FIN flags
- RST and FIN flag count from both client to server and from server to client

3) Application Layer: So far, in the application layer, data has been extracted from three different protocols.

- HTTP: HTTP request name, user agent name and response code is extracted. Average request value, successful and non-successful responses are recorded from these attributes in a session.
- SMTP: SMTP request command, response code and data length transmitted in that SMTP session are extracted. Based on this information; request, response and error response percentage, average data length, number of packets containing full capacity data (for TCP segment, it is 1460 bytes), session QUIT percentage and session closing confirmation percentage are recorded.
- FTP: Current working directory, response code, request command and argument are recorded in a session.

4) Common Preprocessing: This is the module of preprocessing which is applied for all attributes extracted in the first module. In this module, first, any type of encoding is performed if needed (e.g. source and destination IP address). Next scaling is used to standardize the data so that the data can be properly used without any bias in the PCA module.

B. Principal Component Analysis (PCA)

The preprocessed data from network packets are sent to this module for further analysis. PCA is used mainly for dimensionality reduction. In a vector space of n-dimensions,

most often happens that some dimensions do not contribute to the variance of data that much. Thus dropping out those unnecessary dimensions helps to determine the key sub-spaces in outlier detection by dropping the sub-spaces which can be derived from the combination of unnecessary dimensions. Thus, PCA helps to detect outliers efficiently. In PCA, new attributes are calculated from the old ones. New attributes are linear combinations of the older attributes and so they are linearly independent. Thus any dependency among the attributes in the actual data is omitted. Key terms used in the PCA module are:

- Covariance: This term is the indicator to measure the dependency of attributes. If a and b are two variables and CoV(a,b) represents the covariance between them then it can have the following possible values:
  - $+ve$ meaning a and b changes in the same manner.
  - 0 meaning a and b are not co-related.
  - $-ve$ meaning a and b changes in the reverse direction.
- Eigenvector and Eigenvalue: Vector that does not change direction after applying any linear transformation to the corresponding vector space is called eigenvector.
  The factor by which eigenvector is scaled in the vector space is called eigenvalue.

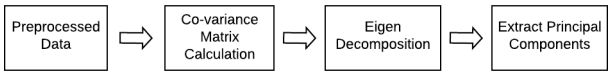

Fig. 2. Flow diagram of PCA

Fig. 2 shows the steps used in PCA which are described as follows:

- Calculate $(n * n)$ covariance matrix by taking the covariance in between each pair of attributes in n-dimensional preprocessed data.
- Calculate the eigenvectors and corresponding eigenvalues from the covariance matrix and then sort the eigenvectors in descending order according to their eigenvalues. These eigenvectors are linearly independent of each other.
- Take the first few eigenvectors $(>= 4)$ as principal components which cover at least 95% of the variance in the preprocessed data.

C. Density-based Clustering (DBC)

DBC enables the system to detect outliers. The main parameters of this algorithm are: The distance ϵ and the number of minimum points (minPts) to determine a cluster. There are three types of points identified by the algorithm:

- Core Point: These are points which have data points greater or equal to minPts inside their ϵ neighbourhood.
- Border Point: These points are reachable from the core points but these points have data points less than minPts value located inside their ϵ neighbourhood.
- Outliers: These points do not lie in the core and border point classes and thus are not reachable from any data point.

Distance between normal network sessions in a particular service is not much (in fact very near to 0). Only an anomalous session will have some values of the recorded features which vary a lot with normal values. Thus the ϵ value can be kept very low to detect malicious sessions. In our proposed method, the distance ϵ value is kept inside $(0.30 - 0.40)$ range and the minPts value is kept inside $(5 - 15)$ range.

For each unlabelled point, the algorithm works as follows:

- If the data point has greater or equal to minPts number of data points in the ϵ neighbourhood then it is marked as a core point.
- If not a core point then it is checked whether the point is reachable from any already detected core points. If yes then it is marked as a border point.
- If not a border point also then the point is marked as an outlier and (-1) label is given to the point.

D. Accumulate Outlier Count

If data points can be clustered in larger dimensions based on density, then that property will also hold in smaller dimensions. In fact, in lower dimensional sub-spaces, distance can be calculated more efficiently. Thus DBC is applied for all of the 3-dimensional sub-spaces which are created from the chosen principal components. Outlier count for each of the data points is recorded. Finally, data points having outlier count in (90-99)% range are marked as anomalies.

E. Attacks detected

The attacks that are detected fall under the following groups;

- Denial of Service (DOS): It is an attack-type where the attacker makes a particular server to go out of service.
- Remote to Local (R2L): In this attack, an attacker tries to gain local or root access of a remote host.
- Probe: In this attack, an attacker searches for entry points which can be exploited to launch other attacks.

F. Strength of our approach

Since our proposed approach uses a combination of PCA and DBC on all possible features, it leads to a strong anomaly detection process due to the following reasons:

- Our proposed approach has the ability to detect an extensive range of anomalies due to the selection of all possible network traffic features
- Our proposed approach performs efficiently due to the reduction of data dimensions using PCA
- It can detect a cluster of any arbitrary shape because of using DBC
- It can accurately detect anomalies as a result of outlier count accumulation from all possible 3-dimensional sub-spaces

IV. Results

The result that we achieved from our proposed system is discussed in detail in this section.

A. Dataset

Our proposed method takes the network traffic as input from the 1999 DARPA Intrusion Detection Evaluation Data Set [14]. Mainly the fourth and fifth week of testing data has been used to detect the anomalous traffic. Data from these two weeks is used because the attack truth table containing the date and time interval information of the attack sessions is available only for the fourth and fifth week of testing data. This truth table also contains the IP address of the victim host on which the attack was executed. Also, network traffic from the first and third week of the data set does not contain any instance of the attacks. These data set contains packet capture files (PCAP) for each day of the fourth and fifth week. All of the attacks are included inside the network traffic of these PCAP files. There is a lot of normal network traffic present in these PCAP files besides the attack traffics. The attack truth table from the data set has been used to detect if there are any false negative and false positive alarms.

B. Performance Metrics

Intrusive samples identified as intrusive and normal samples identified as normal are called true positive (TP) and true negative (TN) respectively. Normal samples detected as intrusive and intrusive samples detected as normal are called false positive (FP) and false negative (FN) respectively. On basis of these four parameters, the following metrics are calculated to determine the overall performance of our proposed method:

- Accuracy (α): Accuracy is the ratio of the number of correct predictions to the total number of input samples. In binary classification, accuracy is calculated as follows:

$$\alpha = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

- True Positive Rate (TPR): TPR or sensitivity is the fraction of attacks detected among the actual attack instances. It is calculated as:

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

- False Positive Rate (FPR): FPR or false alarm probability is the fraction of traffics detected as anomalous among the normal network traffic instances. It is calculated as:

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

C. DOS Attacks

Fig. 3 shows the HTTP sessions as data points for principal components 1, 3 and 5. Apache2 attack is detected using these sessions. Red marked points are outliers. Each combination of three principal components generated identical graphs. Blue data points represent normal session clusters. Similar figures are achieved for other DOS attacks also. Performance metrics value in detecting these DOS attacks is given in Table I.

TPR column of Table I shows that all instances of DOS attacks are detected. FPR column shows that the number of sessions detected as false positives in POD, land and SYN

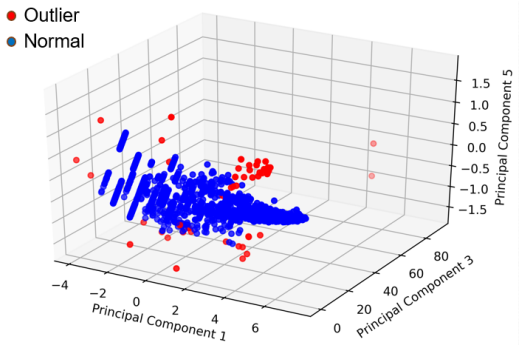

Fig. 3. HTTP sessions as data points concerning principal components 1, 3 and 5

TABLE I
OUTCOMES FROM DOS ATTACK DETECTION

Attack	Accuracy	TPR (%)	FPR (%)
Apache2	99.98	100	0.0116
Back	99.99	100	0.0081
POD	90.90	100	10
Land	99.16	100	0.8379
SYN Flood	98.86	100	1.1334

flood attack is higher in comparison with the detection of apache2 and back attacks. Graphical representation of the anomalous and normal sessions in detecting these DOS attacks is given in Fig. 3. The detailed result achieved from DOS attacks detection is illustrated below:

1) Apache2: Table I represents the performance metrics received in detecting apache2 attack. Total 17113 HTTP sessions are drawn out from the first day of the fifth week of the data set. Four sessions are detected as outliers. Two of them are actual instances of apache2 attack. In these sessions, a high amount of data is transmitted in the form of HTTP user agent through TCP bookkeeping packets. Other two sessions have high error response percentage from the HTTP server.

2) Back: Table I, represents the performance metrics values in back attack detection. Total 24426 TCP sessions are taken from the third day of the fifth week of the data set. Three of them are detected as outliers. Among them, one session is an instance of back attack as it transmits a high amount of data through TCP bookkeeping packets. This session also has a high percentage of RST flag from the server to the client. Other two outliers have high RST and FIN flag percentage.

3) Ping of Death (POD): Table I, shows the performance metrics values in POD attack detection. Thirty-three ICMP sessions are extracted from the first day of the fifth week of the data set. Six of them are detected as outliers. Three of them are actual instances of POD attack as they have many ICMP request packets with high data length (65536 bytes). Other three sessions are false positives having a high number of checksum errors in transmission and replies with no responses from the server.

4) Land: Table I, represents the performance metrics received in detecting land attack. Total 1791 TCP sessions are taken from the second day of the fourth week of the data

set. Among them, 16 are marked as outliers. One of these outliers is an actual instance of land attack. This session has an identical source and destination IP address. Other 15 sessions are false positives. One session has no anomalous factor. Three sessions transmit a high amount of data through the TCP bookkeeping packets. One session has different SYN and SYN-ACK flag percentage. Two sessions have both of these anomalies. One session does not have any packet with RST and FIN flags. Two sessions have high RST and three sessions have high FIN flag percentage. Two sessions have all of these factors.

5) SYN Flood: Table I, represents the performance metrics values in SYN flood attack detection. Total 1942 TCP sessions are taken from the first day of the fifth week of the data set. Twenty-three of them are marked as outliers. One of them is an instance of SYN flood attack having 100% SYN flagged packets with a high average time to live (TTL) value (254) for SYN packets. Other 22 sessions are false positives. Seven of these sessions have no anomalous factor. Four sessions have high FIN flag and four sessions have high RST flag percentage. Two sessions transmit a high amount of data through TCP bookkeeping packets and the percentage of SYN and SYN-ACK flagged packets of these sessions are not identical. Two sessions have unequal SYN and SYN-ACK flagged packet percentage. One session has all of these factors. Other three outliers are instances of apache2 and dosnuke attack.

D. R2L Attacks

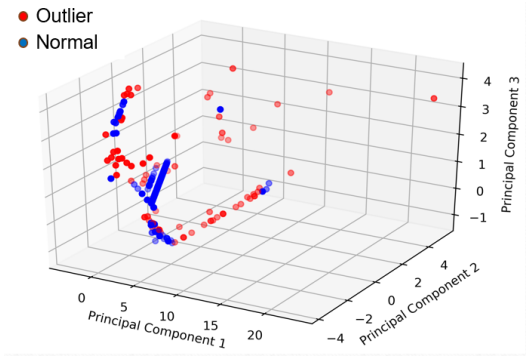

Fig. 4. SMTP sessions as data points concerning principal components 1, 2 and 3

Fig. 4 shows the SMTP sessions as data points concerning principal components 1, 2 and 3. These sessions are used in pp-macro attack detection. Red marked points are outliers. It has many outliers in comparison with Fig. 3. Thus, FPR is also high in detecting this attack in comparison with DOS attacks. Each combination of three principal components generated similar graphs. There are several normal session clusters marked with a bunch of blue data points. Similar graphs are achieved for other R2L attacks also. Performance metrics value in detecting R2L attacks is presented in Table II.

TPR column of Table II shows that all instances of the R2L attacks are detected. FPR column shows that more normal sessions are detected as outliers for pp-macro attack in comparison with FTP-Write and sendmail attack detection. Graphical explanation of the anomalous and normal sessions in R2L attack detection is given in Fig. 4. The detailed result in R2L attack detection is explained below:

TABLE II
OUTCOMES FROM R2L ATTACK DETECTION

Attack	Accuracy	TPR (%)	FPR (%)
PP-Macro	98.1799	100	1.8237
FTP-Write	99.9462	100	0.0537
Sendmail	98.3361	100	1.6652

1) PP-Macro: Table II shows performance metrics values received in detecting pp-macro attack. Total 989 SMTP sessions are taken from the third day of the fourth week of the data set. Among them, 20 are marked as outliers. Two of them are the actual instance of pp-macro attack. In these two sessions, most of the data transmitted are full to TCP segment capacity. Other 18 sessions are false positive. Eight of them contain an unusual combination of DF flag, push flag, SMTP request and response packet percentage and three of them also have an unusual response error rate. Two sessions have unusual request and response rate combination and two sessions have unusual DF and push flag combination. Four sessions have zero error response percentage while other sessions have few error responses. Two sessions have no data transmission.

2) FTP-Write: Table II shows the performance metrics values in FTP-Write attack detection. Total 1860 FTP packets are taken from a subset of the first day of the fourth week of the data set. Among them, two are marked as outliers. One of them is an instance of FTP-Write attack. It tries to append a file into the root directory of FTP server. Another one has a higher number of failure responses from the FTP server.

3) Sendmail: Performance metrics values in detecting sendmail attack are given in Table II. Total 1202 SMTP sessions are extracted from a subset of the first day of the fourth week of the data set. Among them, 21 outliers are detected. One of them is an instance of sendmail attack. This session has no valid starting or closing command for SMTP session yet it tries to transmit data through the SMTP server. Other twenty sessions are false positives. Nine of them have no anomalous factor. Six of them have zero error response percentage where other sessions contain some error responses. Three sessions have high data transmission with no session closing confirmation from the server. Two sessions transmit no data via SMTP.

E. Probe Attacks

Fig. 5 shows the TCP sessions as data points for principal components 1, 2 and 8. These sessions are used to detect portsweep attack. Red marked points are outliers. Again there are many outliers in comparison with Fig. 3 and Fig. 4. Thus, FPR is also high in detecting this attack in comparison with DOS and R2L attacks. Each combination of three principal components generated analogous graphs. Blue data points represent normal session clusters. Performance metrics value in portsweep attack detection is presented in Table III.

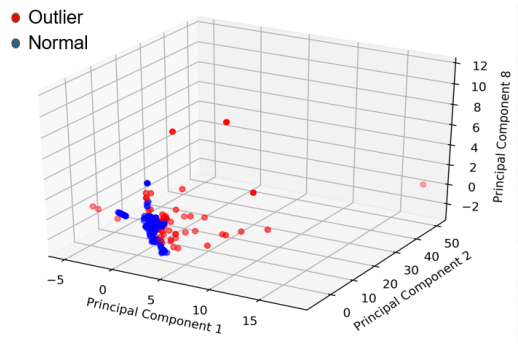

Fig. 5. TCP sessions as data points with principal components 1,2 and 8

TABLE III
OUTCOMES FROM PROBE ATTACK DETECTION

Attack	Accuracy	TPR (%)	FPR (%)
Portsweep	97.92	100	2.077

Again in Table III, the TPR column shows that all portsweep attack instances are detected. FPR value shows that there are many sessions identified as false positives in comparison with DOS and R2L attacks in Table I and Table II. Graphical description of the anomalous and normal sessions in portsweep attack detection is given in Fig. 5. The detailed result achieved in portsweep attack detection is explained below:

1) Portsweep: Table III shows the performance metrics values in portsweep attack detection. Total 1445 TCP sessions are recorded from the first day of the fourth week of the data set. Among them, 31 are detected as outliers. One session among them is an instance of portsweep attack. This session has a high FIN flag percentage where SYN and SYN-ACK percentage is zero. Other 30 sessions are false positives. Twelve of them have no anomalous factor. Four sessions have a high amount of data transmitted through TCP bookkeeping packets. Three sessions have unequal SYN and SYN-ACK percentage. Three sessions have high FIN flag percentage. Other sessions have a combination of anomalous factors from the factors described above.

F. Summary of results

Following is a summary of our findings:

- Our proposed method detected those attacks with no labels or prior knowledge about the attacks.
- We achieved a zero false negative rate (FNR) in detecting the listed attacks.
- In our experiment, precision is quite low. A low ϵ-neighbourhood value helps to detect all instances of the attack leading to zero FNR having the side effect of low precision value.

V. CONCLUSION

In this paper, we have proposed a simple and efficient way of detecting anomalies in network traffic. Our proposed approach has extracted attributes from network packets extensively, applied PCA on them and finally ranked anomalies based on outlier count obtained from DBC in 3-dimensional sub-spaces. Results show that dimension reduction helps significantly in detecting anomalies from a space of a larger and unknown number of dimensions. Again, clustering the sessions in all possible three dimensions has been found useful for anomaly detection rather than clustering in a space of bigger dimensions. Our proposed system has obtained perfect detection and accuracy for the listed attacks. In future, this proposed method can be extended to perform intrusion detection in real-time and for other protocols in the application and transport layer.

REFERENCES

[1] M. A. Jonas, R. Islam, M. S. Hossain, H. S. Narman, and M. Atiquzzaman, "An intelligent system for preventing ssl stripping-based session hijacking attacks," in *IEEE Military Communications (MILCOM)*. Norfolk, VA, USA: IEEE, 12-14 Nov., 2019.

[2] M. I. Ashiq, P. Bhowmick, M. S. Hossain, and H. S. Narman, "Domain flux based dga botnet detection using feedforward neural network," in *IEEE Military Communications (MILCOM)*. Norfolk, VA, USA: IEEE, 12-14 Nov., 2019.

[3] P. Casas, J. Mazel, and P. Owezarski, "Unada: Unsupervised network anomaly detection using sub-space outliers ranking," in *International Conference on Research in Networking*. Berlin, Germany: Springer, May 9., 2011, pp. 40–51.

[4] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 09 November., 2016.

[5] I. Savvas, A. Chernov, M. Butakova, and C. Chaikalis, "Increasing the quality and performance of n-dimensional point anomaly detection in traffic using pca and dbscan," in *26th Telecommunications Forum (TELFOR)*. Belgrade, Serbia: IEEE, 20-21 Nov., 2018, pp. 1–4.

[6] X. Zhao, G. Wang, and Z. Li, "Unsupervised network anomaly detection based on abnormality weights and subspace clustering," in *Sixth International Conference on Information Science and Technology (ICIST)*. Dalian, China: IEEE, 6-8 May., 2016, pp. 482–486.

[7] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," in *National Aerospace and Electronics Conference (NAECON)*. Dayton, OH, USA: IEEE, 27-30 June., 2017, pp. 63–69.

[8] D. S. Terzi, R. Terzi, and S. Sagiroglu, "Big data analytics for network anomaly detection from netflow data," in *International Conference on Computer Science and Engineering (UBMK)*. Antalya, Turkey: IEEE, 5-8 Oct., 2017, pp. 592–597.

[9] Z. Chen and Y. F. Li, "Anomaly detection based on enhanced dbscan algorithm," *Procedia Engineering*, vol. 15, pp. 178–182, 1 Jan., 2011.

[10] Z. Chen, C. K. Yeo, B. S. L. Francis, and C. T. Lau, "Combining mic feature selection and feature-based mspca for network traffic anomaly detection," in *Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*. Moscow, Russia: IEEE, 6-8 July., 2016, pp. 176–181.

[11] W. Chen, F. Kong, F. Mei, G. Yuan, and B. Li, "A novel unsupervised anomaly detection approach for intrusion detection system," in *IEEE 3rd international conference on big data security on cloud (BigDataSecurity)*. Beijing, China: IEEE, 26-28 May., 2017, pp. 69–73.

[12] M. Odiathevar, W. K. Seah, and M. Frean, "A hybrid online offline system for network anomaly detection," in *28th International Conference on Computer Communication and Networks (ICCCN)*. Valencia, Spain: IEEE, 29 July-1 Aug., 2019, pp. 1–9.

[13] M. T. Hossain, M. S. Hossain, and H. S. Narman, "Detection of undesired events on real-world scada power system through process monitoring," in *11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. New York, NY, USA: IEEE, 28-31 Oct., 2020.

[14] "DARPA intrusion detection evaluation dataset," 1999, https://archive.ll.mit.edu/ideval/data/1999data.html.