Evaluation of CICIDS2017 with Qualitative Comparison of Machine Learning Algorithm

Toka Elmasri
German University in Cairo
Cairo, Egypt
toka.elmasri@student.guc.edu.eg

Nour Samir
German University in Cairo
Cairo, Egypt
noureldien.amin@student.guc.edu.eg

Maggie Mashaly
German University in Cairo
Cairo, Egypt
maggie.ezzat@guc.edu.eg

Youmna Atef
German University in Cairo
Cairo, Egypt
youmna.atef@guc.edu.eg

Abstract—**Anomaly Intrusion Detection Systems (AIDS) are crucial for the network security of any organization due to the evolution of novel malware attacks that are capable of deceiving the traditional detection methods. In this paper, we develop three AIDS models using machine learning K Nearest Neighbors (KNN), enhanced KNN and Local Outlier Factor (LOF) techniques. The three approaches were applied on the CICIDS2017 dataset for training, testing and evaluation. A comparison between the three approaches was provided and our model produced promising results with average accuracy of 90.5% for the LOF approach. Contrary to the previous work, our models were tested with no prior training on abnormal samples demonstrating an encouraging average detection rate of 92.74% for zero day attacks.**

Keywords—Cybersecurity, Anomaly Detection, Intrusion Detection, CICIDS2017, Machine Learning

I. Introduction

In today's information age, the surge to have a secure and resilient system against novel attacks and malware is evolving. According to [1] security breaches have escalated 11% since 2018 and 67% since 2014 with hackers attack rate of 39 attacks per second. This resulted in a forecasted worldwide expenditure of $ 133.7 billion in 2020 on cybersecurity and motivating further research for highly efficient Intrusion Detection System (IDS) to overcome these challenging threats. Researchers have categorized the IDSs into two main categories; Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS) [11] while the techniques used are either Signature-Based Intrusion Detection System (SIDS), Anomaly-based Intrusion Detection System (AIDS) or a hybrid combination of both techniques.

The Signature-Based Intrusion Detection Systems also known as Misuse IDSs depend on extracting a sequence or pattern created by the malware known as malware signature, this pattern is then compared to a database of known malwares and if a match is found then an alarm will be raised. This approach is highly efficient in detecting previously known malwares, however it fails to identify any novel attack. Moreover, nowadays advanced malwares can alter their signature frequently evading detection by the traditional SIDS approach.[9]

In the Anomaly-Based Intrusion Detection System, the normal behavior of the system is learned and modeled. The behavior of the system is monitored and any deviation from the normal defined behavior is reported as a malicious behavior. This system overpowers the traditional SIDS as it has the potential to detect zero day attacks, however these systems suffer from higher false alarm rates.

Finally, the hybrid intrusion detection systems which aim to take advantages of both existing approaches enabling the detection of new novel attacks while minimizing the false positive rate and using the database of already known malwares. [10]

A key factor in developing, testing and evaluating an efficient AIDS is acquiring a balanced and well-formed benchmarked dataset. There exists a number of such datasets such as the DARPA98, KDD99, NSL-KDD, ISC2012 and the CICIDS2017 dataset. In this paper, we focus on studying and optimizing the CICIDS2017 under different machine learning algorithms.

The work presented in this paper is the first to test the LOF and enhanced KNN approaches over the CICIDS2017 dataset. Moreover, it is also the first work over the CICISDS2017 dataset that trained only on normal data samples and full sample size. Thus, having the potential of detecting zero day attacks.

The rest of this paper is as follows: section two will cover the literature review. A detailed explanation of the algorithms used and the preprocessing done on the dataset is illustrated in section three. Section four will include the results obtained by testing the dataset, followed by a discussion in section five. Finally, the paper is concluded and future work is proposed in section six.

II. Literature Review

A. CICIDS 2017 Dataset

In [2] a detailed study on the CICIDS2017 files was provided, the paper proposed some modification such as introducing new classes to the dataset in order to resolve the

978-1-7281-8266-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IEEECloudSummit48914.2020.00013

class imbalance issue. The work done in [3] highly resembles our approach however in this paper we study all the provided attack files of the CICIDS2017 dataset applying KNN and LOF machine learning algorithms. Although we use the same approach in applying the PCA analysis, we provide a detailed analysis for the effect of PCA through a full comparison with respect to time complexity and accuracy.

In [4] Jiang et al, the paper focused on the detection of different types of DDOS attacks only, introducing a new hybrid detection model that combines traffic and user behavior features. They applied three layers Back Propagation Neural Network (BPNN) for the classification scoring 99% accuracy. Similarly, Aksu et al.,[5] focused on the classification between the benign and DDOS attacks in the CICIDS2017 dataset. The data were preprocessed using the Fisher Score algorithm reducing the number of features to 30 features. Three machine learning algorithms were applied (KNN, SVM, DT) with reported highest accuracy of 0.9997 with KNN, 0.99 with DT and 0.5776 with SVM. The results of [4][5] were promising, however their approaches focused on detecting certain types of attacks with no proof that these approaches can generalize well with other types of attacks.

The flow based technique was used in [6], were Ullah et Mahmoud proposed a two layer detection model. The first level of the model is responsible for classifying the network traffic flow as benign or anomalous using only the flow-based features. The first layer used decision tree algorithm, while the second layer is used to classify the anomalous flow using Random Forest algorithm. In addition to that, Recursive Feature Elimination (RFE) is used to select significant features and Synthetic Minority Over-Sampling Technique (SMOTE) is used for oversampling, while Edited Nearest Neighbors (ENN) is for cleaning the CICIDS2017 dataset. In this paper, a detailed result is applied for each type of attack presented at the dataset scoring an average specificity and F score of 100%. In [12] a flow based intrusion detection system for multi-class classification was developed using deep neural networks. The model used the Pearson correlation test in order to reduce the number of features. The model showed promising results per some types of attacks such as the DDOS, where a 99.8 precision value and 99.8 recall were recoded, however no overall average accuracy of the model performance was reported.

In [14] Rosay et al, a multi-layer fully connected feed forward neural network classifier was proposed. The paper used a train, cross-validation and test data split and the authors clarified they had two evaluated variants of the CICIDS2017 dataset. The first contained 73 features while the seconds had 70 features. The paper did not implement a feature reduction technique but relayed on statistical analysis which proved not as highly efficient in detecting and reducing the correlated features and the overall running time of the algorithm. The model was able to score approximately 99% average accuracy while keeping the false positive rate below 7%.

B. Machine Learning Algorithms

The work done in [7] by A. Lazarevic et al, evaluated different supervised and unsupervised machine learning algorithms against the DARPA 1998 dataset and real network data. The results indicated that the LOF algorithm had the highest detection rate for the DARPA dataset and was able to detect multiple novel attacks highlighting the strength of this machine learning approach. The paper also suggested that unsupervised support vector machine and KNN are very promising in detecting novel attacks.

To the best of our knowledge, this is the first paper to implement the LOF approach on the CICIDS2017 dataset. Moreover, we implement an enhanced KNN algorithm using the similarity approach. In contrast to most of the previous work [3][4][5], this paper studies all the attacks provided in the dataset not focusing on a specific type. Moreover, unlike the rest of the previous work which either trained on the anomalous data or were ambiguous about their training methods failing to mention if it contained any malicious samples, all of our models trained only on normal data samples and full sample size. Thus, having the potential of detecting zero day attacks.

III. Methodology

In this paper, we use a semi supervised anomaly detection approach. In this approach the model is trained using only normal data, thus in the cross validation and testing phase, the model can classify the anomalous activities based on the deviation from a certain threshold. We train on 60% of the normal data in the CICIDS2017 dataset. The rest is divided equally between the cross validation set and the testing set.

A. KNN

The K-Nearest Neighbor (KNN) is a straightforward, machine learning algorithm. It is a non-parametric algorithm that does not make any underlying assumptions about the data distribution. KNN is known to be a lazy algorithm as it does not learn a particular function from the training model, instead it 'memorizes' the dataset. The parameter K represents the number of neighbors that are nearest to the data point being tested or evaluated. Unsupervised or semi-supervised algorithms do not contain labels; therefore, the algorithm was implemented without the use of labels. In addition to that, the training model consists of normal data only, while the cross validation and test sets contain both normal and abnormal data.[15]

The KNN algorithm follows a sequence of steps to be able to identify the anomalous events whenever they are injected into the system. These steps are as follows; first, the algorithm models a set of normal events provided by the system. Next, whenever a new event occurs, the algorithm calculates the distance of the new event with the K nearest normal events in the model created. Depending on this distance and a threshold previously decided by the algorithm, the event is flagged as anomaly if its distance is too large compared to the threshold or identified as normal if the distance is small. The distance could be measured by different mathematical equations such as Euclidean, Manhattan, Minkowski, or Hamming distance. The Euclidean distance is implemented in this algorithm. Moreover, the threshold and k parameters are decided by running the algorithm multiple times and measuring its performance using the cross validation set to decide which threshold and k give the optimal results.

B. *KNN with similarity*

Similarity KNN [16] is a type of algorithm similar to the normal KNN with one additional parameter which is the similarity between neighbors. This attribute measures the degree of how similar an event is to its K neighbors. This parameter ranges from 0 to 1. The closer the number is to 1 the more similar it is to its neighbors and hence it is defined as a normal event. This similarity measure is calculated using the cosine similarity equation shown in (1). This measure provides one more parameter that helps distinguish between normal and malicious events more accurately.

$$\frac{A.B}{||A||\times||B||} = \frac{\sum_i^n A_i \times B_i}{\sqrt{\sum_i^n A_i^2} \times \sqrt{\sum_i^n B_i^2}} \tag{1}$$

C. *LOF*

The local outlier factor (LOF) algorithm is an anomaly detection algorithm which calculates the local density deviation of a given data point with respect to its neighbors. A data point is considered an outlier when it has a lower density than its neighbors. The local density is estimated by the typical distance at which a point can be "reached" from its neighbors. The definition of "reachability distance" used in LOF is an additional measure to produce more stable results within clusters. LOF is a score that tells how likely a certain data point is an outlier/anomaly[13].

This section presents basic definitions and steps for the calculation of LOF score which is based on the concept of local density. The LOF is calculated in 5 steps, each step is essential for obtaining the final value of each data object. First, the algorithm calculates the k-nearest distances between a certain event and the other k-nearest events. Parameter k being the number of neighbors needed to calculate the distance with the original event. After locating the k-nearest events and their distances with the original event, these distances are used to calculate the reachability distance, which is the second step in this algorithm. The reachability distance is the sum of the maximum k distance of its neighbor and the distance between the original event and this neighbor using the equation (2):

$$Reachability\ Distance\ (A,B)_k = \sum max\Big(\big(k - distance(B)\big), distance(A,B)\Big) \tag{2}$$

Where B represents the neighbor while A represents the original event. Using the reachability distance we calculate the local reachability density which is the reciprocal of the average of the calculated reachability distance. It is calculated using equation (3) where N(A,k) is the set of k nearest neighbors of A:

$$LRD(A)_k = \frac{1}{k}\sum_{o\in N(A,k)}(reachability\ distance - distance(A,B))^{-1} \tag{3}$$

Lastly, the LOF score is calculated in this step to decide whether this event is an outlier or not. The LOF score requires the LRD of its original event and the k-neighbors event to be calculated to satisfy equation (5):

$$LOF(A)_k = \frac{1}{k}\sum_{o\in N(A,k)}\frac{LRD(O)_k}{LRD(A)_k} \tag{5}$$

The LOF score is compared with a threshold parameter, if this score is bigger than the threshold, then the event is considered an outlier. If the calculated LOF value for a certain event is smaller than a predefined threshold, this indicates that the event has higher density than neighbors thus, the event is considered an inlier. On the other hand, if the LOF value is greater than the predefined threshold indicating that it has lower density than neighbors the event will be considered an outlier. After knowing whether each of the data is an outlier or inlier, they are used to calculate the confusion matrix. This confusion matrix is a table used to describe the performance of a classification model, where the matrix is a summary of prediction results.

D. *Preprocssing of Datasets*

Preparing the dataset before applying any algorithm to it is very important as there could be some redundant records or correlated features that could affect the accuracy and function of the algorithm. Therefore, preprocessing of datasets is crucial to ensure a well build dataset and accurate results. Firstly, we remove any redundant features. Secondly, nominal features that have no effect on the output are removed as well. In addition, duplicate records/events are erased. Moreover, features/columns with variance equal zero are removed, and records that include infinity or NaN are replaced by the mean of the feature and zero respectively. All these steps ensure that the dataset is robust and will provide precise and true results without any bias factor affecting the results. Normalization is another factor that assists in scaling the dataset. This is important when evaluating any dataset in order to have all attributes within the same scale and not have extreme variations in these attributes which might not provide decisive and correct results. Furthermore, principal component analysis, PCA, is also applied to the dataset to remove any correlated features and provide only the independent features that will help improve the results. Both normalization and PCA are described in the next section.

E. *Normalization and PCA*

Both normalization and PCA are applied using the Sklearn python library. The normalization pre-defined function performs L2 norm on the data. The L2 norm calculates the distance of the vector coordinate from the origin of the vector space. As such, it is also known as the Euclidean norm as it is calculated as the Euclidean distance from the origin. The result is a positive distance value. The L2 norm is calculated as follow:

- $v= [1,2,3]$, norm of v: $||v||= \sqrt{(1^2+2^2+3^2)} = \sqrt{14}=3.741657387$
- Then normalize the vector using $v/||v||$
- $1/\sqrt{14}=0.267$
- $2/\sqrt{14}=0.534$
- $3/\sqrt{14}=0.801$
- Normalized $v= [0.267,0.534,0.801]$

As for the PCA, it is a technique that reduces dimensions and helps identify the correlation between features and transform them to a lower dimension by clustering correlated features without losing important information. The steps of computing PCA are as follow:

- Standardization of the data
- Computing the covariance matrix
- Calculating the eigenvectors and eigenvalues
- Computing the Principal Components
- Reducing the dimensions of the data set

IV. Results

The results presented in this section took in consideration the effect of applying PCA on the accuracy, detection rate and time complexity of the model. The accuracies across all three models were recorded for all different attacks and are provided in tables [1], [2] and [3] respectively. Furthermore, tables [4],[5],[6] demonstrate the effect of applying PCA on reducing the number of features and the time taken to train and test the models. Table [7] shows the highest detection rates achieved in our dataset evaluation.

TABLE I

KNN ACCUARCY RESULTS

File Name	*Attack*	*KNN Accuracy Without PCA*	*KNN Accuracy With PCA*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	87.992	87.992
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	84.899	84.880
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	87.539	87.539
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	88.992	90.023
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	88.154	88.180
Tuesday-WorkingHours.pcap_ISCX	Patator	77.626	77.728
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	75.076	78.250

TABLE II

SIMILARITY KNN ACCUARCY RESULTS

File Name	*Attack*	*KNN Accuracy Without PCA*	*KNN Accuracy With PCA*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	87.059	86.477
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	89.532	98.284
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	89.193	88.723
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	94.000	92.394
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	92.917	96.719
Tuesday-WorkingHours.pcap_ISCX	Patator	84.396	88.182
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	65.470	67.307

TABLE III

LOF ACCUARCY RESULTS

File Name	*Attack*	*LOF Accuracy Without PCA*	*LOF Accuracy With PCA*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	91.23	92.33
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	95.81	95.84
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	93.46	93.58
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	94.85	95.13
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	93.70	93.95
Tuesday-WorkingHours.pcap_ISCX	Patator	85.95	89.52
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	80.79	81.04

Regarding table [1], the accuracies remained the same in the DDOS and Botnet attack files before and after PCA. There was a slight elevation noticed in the Webattack and Patator attacks with 0.026% and 0.102% respectively. In addition, the Portscan attack was the only file with a slight decrease of 0.019%. Lastly, the Infiltration and Heartbleed & DOS attacks have a noticeable increase of 1.031% and 3.174% respectively.

The results in table [2] highlights the effect of PCA on the accuracy. The DDOS, Botnet and Infiltration attacks decreased slightly by 0.582%, 0.47% and 1.606% respectively. There was a significant increase in the accuracies for the Webattack, Patator and Heartbleed & DOS files, where the accuracies were higher by 3.802%, 3.786% and 1.837% consecutively. The most significant variation was in the Portscan file, where the accuracy was improved by 8.752%. This indicates that the PCA had a positive effect on the results.

The results in table [3], show that the PCA had a slight effect on the LOF accuracy, increasing the accuracy of all files by an insignificant percentage except in the Patator class were the accuracy had a jump of 3.57% after applying the PCA.

TABLE Iv

TIME REDUCTION AFTER PCA FOR KNN

File Name	*Attack*	*NO. of components after PCA*	*Percentage decrease in time*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	40	38.64%
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	23	59.31%
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	21	64.82%
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	6	84.57%
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	30	51.25%
Tuesday-WorkingHours.pcap_ISCX	Patator	30	54.46%
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	9	80.28%

TABLE V

TIME REDUCTION AFTER PCA FOR SIMILARITY KNN

File Name	*Attack*	*NO. of components after PCA*	*Percentage decrease in time*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	13	69.15%
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	19	66.88%
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	40	37.84%
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	25	58.77%
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	2	88.95%
Tuesday-WorkingHours.pcap_ISCX	Patator	10	71.53%
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	15	65.56%

TABLE vI

TIME REDUCTION AFTER PCA LOF

File Name	*Attack*	*NO. of components after PCA*	*Percentage decrease in time*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	10	55.39%
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	20	44.24%
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	20	35.87%
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	10	27.17%
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	10	38.1%
Tuesday-WorkingHours.pcap_ISCX	Patator	10	27.91%
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	50	69.02%

The results shown in tables [4], [5] & [6] emphasize the effect of PCA, not only improving the accuracy in most of the files but also in reducing the time complexity significantly in all the models. As the percentage decrease in time taken with PCA is almost more than 27 percent for all the files. This feature is highly valuable to algorithms when applied to large datasets.

TABLE vII

DETECTION RATE COMPARISON

File Name	*Attack*	*Model with highest TP*	*Detection Rate*
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	DDOS	LOF without PCA	99.96%
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Portscan	LOF without PCA	99.99%
Friday-WorkingHours-Morning.pcap_ISCX	Botnet	LOF with PCA	95.92%
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Infiltration	Similarity KNN without PCA	72.22%
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Webattack	LOF with PCA	94.12%

Tuesday-WorkingHours.pcap_ISCX	Patator	Similarity KNN without PCA	93.148%
Wednesday-workingHours.pcap_ISCX	Heartbleed& DOS	LOF without PCA	93.82 %

V. Disscussion

The results with respect to accuracy; showed that the LOF had a better performance for the DDOS, Botnet, Infiltration, Parator and HeartBleed attacks. On the other hand, the similarity KNN showed higher accuracies for the Portscan and Webattack attacks. Both algorithms outperformed the simple KNN algorithm for most of the attack types.

The results in table [7] shows that both LOF and similarity KNN demonstrated the highest detection rates, with LOF having a potential to reach much higher detection rates. We have to clarify that the models highest detection rates did not correspond to the highest accuracies achieved and reported in the previous tables.

The approach taken in this paper differs from the rest of the previous works making a straightforward comparison hard. As the authors in [3][6][12] were ambiguous about the usage of abnormal data samples in the training phase. Moreover, in [6] only the cross validation results were shared with no testing results provided for comparison. Moreover, the work in [4-5] focused and trained the model in order to detect a specific type of attack (DDOS). The rest of the literature review trained on anomalous data samples, thus having prior knowledge of all the types of abnormal behavior expected in the dataset. The approach taken by the previous works achieved high accuracies and detection rates, however provides no indication of the performance regarding the novel attacks, giving an edge to our work and making a direct comparison between the accuracies achieved not impartial. Finally with that being illustrated we tried to perform a qualitative comparison with respect to the accuracy and the detection rate matrices where our models were able to reach average accuracies of 88.3% and 90.54% using similarity KNN and LOF respectively. The models have a good detection rate in comparison to the literature review [2][12] for the DDOS, Portscan and Webattack attacks.

VI. Conclusion

Developing sophisticated anomaly intrusion detection systems is crucial as they are the first defense line against novel attacks. In this paper three IDS models were proposed using KNN, enhanced KNN and LOF algorithms. The models were tested and evaluated against the CICIDS2017 dataset achieving promising results, specifically with the LOF and enhanced KNN where the models were able to achieve up to 90.54% and 88.3% accuracies respectively without any prior training on the abnormal data. Both models proved to have a high potential of detecting novel attacks due to the training approach implemented. For future work, we target to combine both models with using a probabilistic approach. We also aim to try other feature reduction techniques and implement other machine learning algorithms such as support vector machine.

References

[1] https://www.varonis.com/blog/cybersecurity-statistics/

[2] Panigrahi, Ranjit & Borah, Samarjeet. (2018). “A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems,” 7. 479-482.

[3] Boukhamla, Akram & Coronel, Javier. (2019). “CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed”.

[4] J. Jiang *et al*., "ALDD: A Hybrid Traffic-User Behavior Detection Method for Application Layer DDoS," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, NY, 2018, pp. 1565-1569, doi: 10.1109/TrustCom/BigDataSE.2018.00225.

[5] D. Aksu, S. Ustebay, M. Aydin, and T. Atmaca, “Intrusion Detection with Comparative Analysis of Supervised Learning Techniques and Fisher Score Feature Selection Algorithm,” 09 2018, pp. 141–149.

[6] I. Ullah and Q. H. Mahmoud, "A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks," *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2019, pp. 1-6, doi: 10.1109/CCNC.2019.8651782.

[7] A. Lazarevic, L. Ertz, V. Kumar, A. Ozgur, and J. Srivastava. “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” Proc. SIAM Conf. Data Mining, *2003*..

[8] “Evaluation of Network Intrusion Detection with Features Selection and Machine Learning Algorithms on CICIDS-2017 Dataset,” *International Conference on Advances in Engineering Science Management & Technology (ICAESMT) - 2019, Uttaranchal University, Dehradun, India*

[9] B. Wanswett and H. K. Kalita, "The Threat of Obfuscated Zero Day Polymorphic Malwares: An Analysis," *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, Jabalpur, 2015, pp. 1188-1193, doi: 10.1109/CICN.2015.230.

[10] M. Ali Aydın, A. Halim Zaim, K. Gökhan Ceylan, “A hybrid intrusion detection system design for computer network security,Computers & Electrical Engineering,” Volume 35, Issue 3,2009, Pages 517-526, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2008.12.005.

[11] D. A. Bhosale and V. M. Mane, "Comparative study and analysis of network intrusion detection tools," *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Davangere, 2015, pp. 312-315, doi: 10.1109/ICATCCT.2015.7456901.

[12] P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou and D. Tzovaras, "An Intrusion Detection System for Multi-class Classification Based on Deep Neural Networks," *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Boca Raton, FL, USA, 2019, pp. 1253-1258, doi: 10.1109/ICMLA.2019.00206.

[13] Alpaydin E (2020) “Introduction to machine learning,” 2nd edn. MIT Press, London

[14] Rosay A., Carlier F., Leroux P. (2020) MLP4NIDS: “An Efficient MLP-Based Network Intrusion Detection for CICIDS2017 Dataset,” In: Boumerdassi S., Renault É., Mühlethaler P. (eds) Machine Learning for Networking. MLN 2019. Lecture Notes in Computer Science, vol 12081. Springer, Cham

[15] Liao, Yihua & Vemuri, Rao. (2002). “Use of K-Nearest Neighbor classifier for intrusion detection,” Computers & Security. 21. 439-448. 10.1016/S0167-4048(02)00514-X.

[16] Li, Yang & Fang, Binxing & Guo, Li & Chen, You. (2007). “Network anomaly detection based on TCM-KNN algorithm,” 13-19. 10.1145/1229285.1229292.