International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228329

Network Intrusion Detection System Using Machine Learning

Shailaja Jadhav, Varsha Yadav, Vinaya Bhalerao, Bhavana Shinde, Snehal Kambale
Department of Computer Engineering, Marathwada Mitra Mandal's College of Engineering, KarveNagar, Pune, Maharashtra, India

ABSTRACT

Article Info
Volume 8, Issue 3
Page Number : 208-216

Publication Issue :
May-June-2022

Article History
Accepted: 16 May 2022
Published: 30 May 2022

The latest advances in the internet and communication areas have resulted in a massive expansion of network size and data. As a result, plenty of new dangers have arisen, making it difficult for network security to identify attacks effectively Furthermore, intruders with the intent of executing innumerable assaults within the network cannot be overlooked. An intrusion detection system (IDS) is a tool that inspects network traffic to verify confidentiality, integrity, and availability. Despite the researchers' best efforts, IDS continues to encounter difficulties in boosting detection accuracy while lowering false alarm rates and detecting fresh intrusions. Machine learning (ML)-based IDS systems have recently been deployed as promising solutions for quickly detecting intrusions across the network. This article defines IDS and then presents a taxonomy based on prominent machine learning techniques used in the construction of network-based IDS (NIDS) systems. The benefits and drawbacks of the proposed solutions are discussed in depth in this detailed evaluation of current NIDS-based studies. The proposed technique, evaluation criteria, and dataset selection are then discussed, as well as recent trends and breakthroughs in ML-based NIDS. We highlighted many research obstacles and recommended future research scope for improving ML-based NIDS using the weaknesses of the proposed approaches.

Keywords: nids, dtc, bnb, knn, Dynamic Complex types of security

I. INTRODUCTION

Network security has arisen as a critical research subject as a result of the current interest and advancement in the development of internet and communication technologies over the previous decade. To protect the security of the network and all its related assets within cyberspace, it uses tools such as firewalls, antivirus software, and intrusion detection systems (IDS). 1 The network-based intrusion detection system (NIDS) is an attack detection method that delivers the requisite security by continuously monitoring network traffic for harmful and suspicious activity.

Jim Anderson first presented the idea of IDS in 1980. Many IDS products have been developed and refined since then to meet the needs of network security.

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

However, during the previous decade, technical improvements have resulted in a huge development in network capacity and the number of applications handled by network servers. As a result, a massive amount of critical data is generated and shared among network nodes. The production of a significant number of new assaults, either through mutation of an old attack or a novel attack, has made the safekeeping of these data and network nodes a problematic issue. Security apprehensions can affect almost every node in a network. For example, the data node may be extremely significant to a company. Any compromise of the node's information could have a significant negative impact on the organization's market reputation and financial losses. Existing IDSs have demonstrated inefficiency in detecting a variety of assaults, including zero-day attacks, and in lowering false alarm rates (FAR). This eventually leads to a demand for a network intrusion detection system (NIDS) that is efficient, accurate, and cost-effective.

The researchers investigated the use of machine learning (ML) and deep learning (DL) approaches to meet the requirements of a successful IDS. Both ML and DL fall underneath artificial intelligence (AI) and strive to extract meaningful information from large amounts of data. Due to the invention of extremely powerful graphics processor units, these techniques have gained significant prominence in the world of network security during the last decade (GPUs). Both ML and DL are effective methods for extracting relevant information from data. Estimating regular and abnormal activities based on learned patterns in network traffic to learn relevant information from network traffic, the ML-based IDS mainly relies on feature engineering. DL-based IDS, on the other hand, do not rely on feature engineering and are capable of learning complicated features from raw data because of their deep structure.

II. Background

Intrusion detection is a well-known problem in cybersecurity research and practice. Numerous approaches have been proposed to develop better intrusion detection systems (IDS). Figure 1 presents a general taxonomy of IDS mechanisms based on data collection methods and attack detecting techniques.

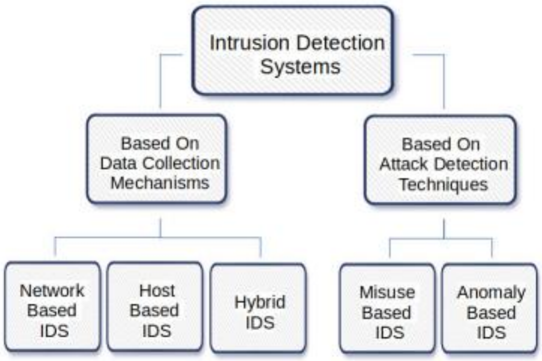

Intrusion Detection Systems
Based On Data Collection Mechanisms
Based On Attack Detection Techniques
Network Based IDS
Host Based IDS
Hybrid IDS
Misuse Based IDS
Anomaly Based IDS

Figure 1: Taxonomy of Intrusion Detection Systems based on data gathering and threat detection strategies

The host-based IDS (HIDS) mechanisms investigate behaviour at the endpoints, particularly the hosts, among the IDS mechanisms based on data collection. The HIDS mechanism works by tracking processes and monitoring system resources including memory usage, CPU use, and disc I/O to detect any aberrant activity. As a result, a HIDS's obligation is limited to the host on which it runs [6]. A network-based IDS (NIDS), in contrast to a HIDS, captures data transferred through the network using sensors and then analyses it to detect suspicious incidents. A NIDS can be deployed in another network segment or work alone at the protected network's entry point to capture and inspect each incoming/outgoing packet. By deploying tap devices at crucial nodes, traffic can be forwarded to an NIDS for inspection. A hybrid IDS, as its name implies, tries to combine the strengths of both HIDS and NIDS. This strategy promotes IDS deployment on both hosts and networks to examine local occurrences and evaluate traffic flow. Although the concept is appealing, the fundamental difficulty for hybrid IDS methods is the enormous administrative burden placed on information security experts. Misuse-based IDS techniques detect the precise behaviour displayed by an attack or its signature. Misuse-based systems are a strong defence against well-documented and acknowledged intrusion attempts. Words, its signature. Misuse-based systems constitute an effective solution against intrusion attempts that are well recognized and documented.

Misuse-based IDS techniques, on the other hand, have drawbacks. Unknown or zero-day assaults may go undetected. In addition, each attack type must be transformed into a rule in order to be detected, necessitating frequent and rapid changes to maintain appropriate protection. An anomaly-based IDS, in contrast to misuse-based IDS, is based on the notion of detecting a deviation from expected system behaviour. Common anomaly-based network IDS systems, for example, typically detect irregular congestion or unusual activity, such as excessive packet re-transmissions. As a result, the primary purpose of anomaly-based IDS is to analyse and categorise network flow in order to distinguish hostile attempts from normal occurrences. Unlike the misuse-based approach, there are no explicit guidelines for categorising what constitutes an infiltration attempt.

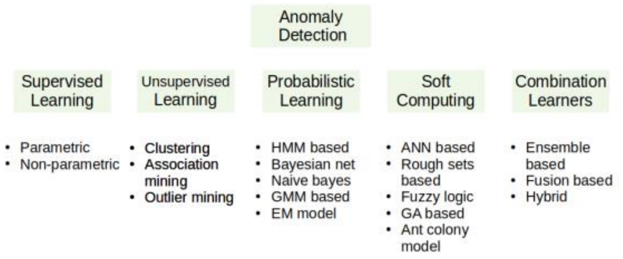


Figure 2 : Anomaly-based intrusion detection via machine learning approaches.

III. Machine Learning

Machine Learning is an application of Artificial Intelligence (AI) that delivers systems the capability to automatically learn and improve from experience without being explicitly programmed. It means with understanding that we have to contribute the learning algorithm examples or instances and tell it to make conclusions from that example to design a hypothesis and uses it to do the necessary task. The task differ from application to another for instances Image Classification, House Price Prediction, Speech Recognition, Email Classification (Spam or not), Packet Classification (Normal or Malicious), . . . etc.

In traditional programming, the programmer provides the inputs and tells to the Algorithm what to do with these inputs (i.e. the algorithm is designed openly to solve a such problem), whereas with ML approach, the programmer must.

Give the input and the output (results) and the ML algorithm will try to find a relationship between them (programe) as shown in the Figure 3.

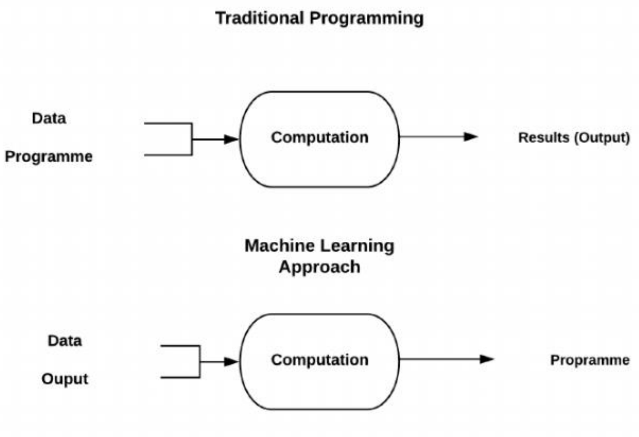


Figure 3: Traditional Programming vs Machine learning approach

Machine learning algorithms are grouped into three categories as follow: (a) supervised learning, (b) unsupervised learning, and (c) reinforcement learning.

A. Supervised Machine Learning Algorithm: we have trained data with known labels. Machine algorithm trained on labelled data. During this type, the data must be labelled accurately to figure. The foremost-supervised learning technique is Classification.

B. Unsupervised Machine Learning Algorithm: Throughout this learning experience, trained data with unidentified labels is available. In addition, it notices straightaway from the data based supported cosine similarity. Clustering is the most extensively Used unsupervised learning technique

C. Reinforcement Learning: Throughout that learning experience, a computer was made available for achieving a particular goal. It features self-improving algorithm that learns from new situations through trial and Error. Machine learning for IDS can solve a variety of issues, including speed and computational time, while also allowing for the development of accurate IDS.

IV. LITERATURE SURVEY

Reference	Dataset	Method	Accuracy
I. Sumaiya Thaseen and C. Aswani Kumar, “Intrusion Detection model using fusion of chi square feature selection and multi class SVM,” J. King Saud Univ. - Computer. Inf. Sci.,	NSL-KDD	SVM Radial Basis Function (RBF)	98.10
W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, “Multi-level hybrid support vector machine and extreme learning machine based on Modified K-means for intrusion detection system,” Expert Syst. Appl., vol. 67, pp. 296– 303, 2017.	KDDCup99	Multi level hybrid Support Vector Machine (SVM) and ELM	95.80
A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, “Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic,” Expert Syst. Appl., vol. 92, pp. 390– 402, 2018.	Real network traffic	Fuzzy Logic	96.50
T. Hamed, R. Dara, and S. C. Kremer, “Network intrusion detection system based on recursive feature addition And bigram technique,” Computer Secure, vol. 73, pp. 137– 155,	ISCX 2012	Recursive Feature Addition (RFA) with SVM	91.90

2018.			
E. K.Viegas and L. S. Oliveira, “Towards reliable anomaly-based intrusion detection in real- world environments,” Computer Networks, vol. 127, pp. 200–216, 2017.	TRAbID (Probe, DoS)	Decision Tree (DT) and Naïve Bayes (NB)	Probe; DT (98.42) , NB (97.29) DoS; DT (99.90), NB (99.66)

V. Dataset

KDD Cup 99:

Since its publication in 1999, KDD CUP 99 has been one of the most popular datasets. The goal of the MIT Lincoln Labs project was to establish a standard collection of data with a wide variety of attacks in order to review and evaluate intrusion detection research. The collection includes nine weeks of raw TCP dump data from a simulated United States Air Force network, as well as multiple attacks. Connection records are created by combining packets from the same connection.

Table 1: Examples of KDD CUP 99 features

Feature name	Type	Description
Description	continuous	length (number of seconds) of the connection
protocol_type	discrete	type of the protocol, e.g. tcp, udp, etc.
service	discrete	network service on the destination, e.g., http, telnet, etc.
src_bytes	continuous	number of data bytes from source to destination
dst_bytes	continuous	number of data bytes from destination to source
flag	discrete	normal or error status of the connection
wrong_fragment	continuous	number of “wrong” fragments

Each connection is classified as normal or as one of four types of attacks mentioned below:

DoS, network probe, Remote to Local (R2L), or User to Root (U2R). A DoS attack is an attempt to prevent users from accessing a system or service. A probe attack is a malicious network activity, such as port scanning that aims to learn about the network's architecture. When an attacker acquires local access to a system over the network, this is known as an R2L assault. A U2R attack takes use of system flaws to gain superuser capabilities.

VI. Design Work

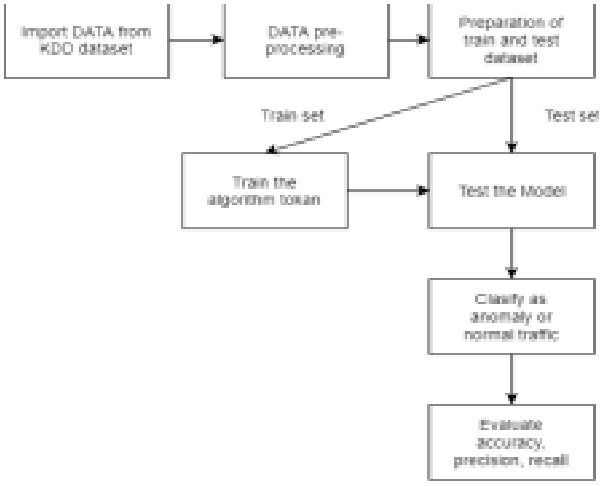

Import DATA from KDD dataset
DATA pre-processing
Preparation of train and test dataset
Train set
Test set
Train the algorithm token
Test the Model
Classify as anomaly or normal traffic
Evaluate accuracy, precision, recall

Figure 4: System Design Architecture

Upload Dataset:

Collect suspicious traffic and ancillary data that defines or characterises it, identifying different network connections or relationships (connectionless traffic), and providing enough detail to aid criminal investigations and prosecutions. Detect incursions that are particular to a protected area. Service overloads, broadcast storms, and message floods are all examples of denial of service attacks.

Pre-processing:

To reduce data burden, perform data reduction, ideally at the source. To eliminate redundancy and false alerts, improve raw data. To create reports for following up on suspicious events or found vulnerabilities that are not getting prompt notice and taking corrective action. Open, track, and document the resolution of a detected intrusion event or vulnerability. An operator to keep track of site-specific activity can use a site profile database.

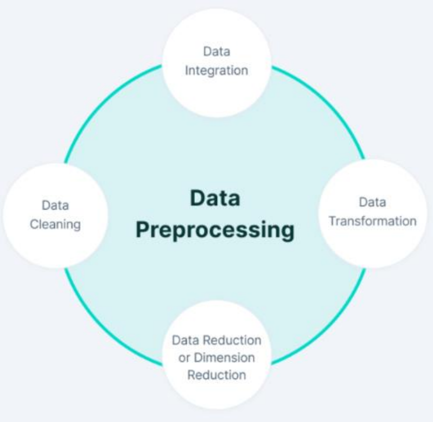


Figure 5: Data-Pre-processing

Feature Selection/Extraction:

A classification task normally requires training and testing data that contains a variety of data examples. The training set encompasses one "target value" (class labels) and multiple “attributes" for each instance (features). SVM's purpose is to create a model that predicts the target value of data instances in the testing set given just the characteristics. The following data is tested and shown on the output: The Attacker Profile output gives information on an attacker, whether an invader or a prober, an outsider or an insider. The Security Profile output gives you extensive security information about a network domain you have chosen. The System Profile shows which areas—addresses, components, and systems—are affected.

Table 2: Features in Pre-processing data

No	Feature Description	Data Type
1	TCP Packets	Integer
2	TCP Source Packets	Integer
3	TCP Fin Flag	Integer
4	TCP Destination Packets	Integer
5	TCP syn flag	Integer
6	TCP urgent flag	Integer
7	UDP Packets	Integer
8	UDP Source Packets	Integer
9	UDP Destination port	Integer
10	ICMP Packets	Integer

VII. Algorithm

A. Naive Bayes:

Naive Bayes is a classification algorithm of Machine Learning based on Bayes theorem, which gives the likelihood of occurrence of the event. Naive Bayes classifier is a probabilistic classifier, which means that given an input, it predicts the probability of the input being classified for all the classes. It is also called conditional probability.

Bayes theorem is as follows:

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)\,P(H)}{P(\mathbf{X})}$$

Bernoulli Naive Bayes:

It works on the Bernoulli distribution and is used for discrete data. Bernoulli Naive Bayes' key feature is that it only accepts binary values for features such as true or false, yes or no, success or failure, 0 or 1, and so on. We know we have to perform the Bernoulli Naive Bayes classifier when the feature values are binary.

Bernoulli Naive Bayes Classifier is based on the following rule:

$$P(xi \mid y) = P(i \mid y)\, xi + (1 - P(i \mid y))(1 - xi)$$

B. Decision tree classifier

Decision Tree is a supervised learning approach that can be applied to classification and regression problems, however it is most commonly employed to solve classification problems. Internal nodes represent dataset attributes, branches represent decision rules, and each leaf node represents the conclusion in this tree-structured classifier.

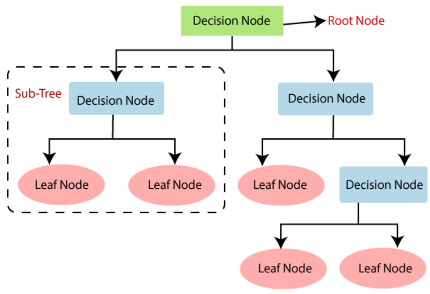


Figure 6: Decision Tree Classifier

Algorithm:

Step 1: Start with the root node, which holds the entire dataset, says S.

Step 2: Using the Attribute Selection Measure, find the best attribute in the dataset (ASM).

Step 3: Subdivide the S into subsets that contain the best attribute's possible values.

Step 4: Create the node of the decision tree that has the best attribute.

Step 5: Create additional decision trees in a recursive manner using the subsets of the dataset obtained in step 3. Continue this process until the nodes can no longer be classified and the final node is designated as a leaf node.

C. K-Nearest Neighbour (KNN)

The K-NN algorithm assumes that the new case/data and existing cases are similar and places the new case in the category that is most similar to the existing categories.

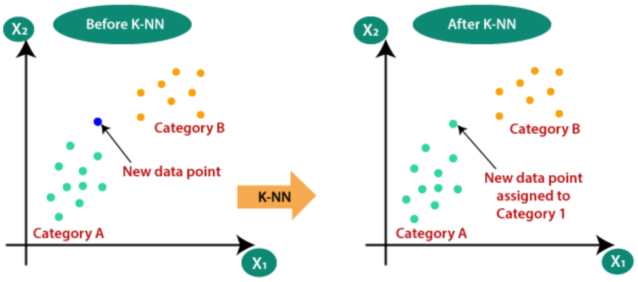


Figure 7: K-nearest neighbor algorithm

Algorithm:

Step 1: Determine the number of neighbours (K).

Step 2: Determine the Euclidean distance between K neighbours.

Step 3: Using the obtained Euclidean distance, find the K closest neighbours.

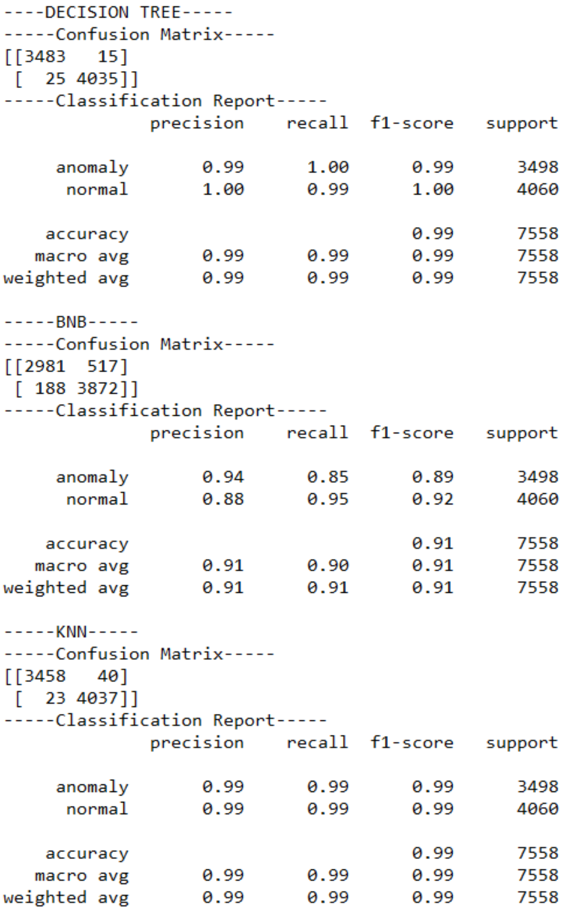
Step 4: Count the number of data points in each category among these k neighbours.

Step 5: Assign the new data points to the category with the greatest number of neighbours.

Step 6: Our model is now complete.

VIII. Result and discussion

This study effort uses the Nave Bayes algorithm, decision tree classifier, and K-nearest neighbour algorithms for training and testing in order to analyses the above literature work. It works with the normal KDD'99 Cup data set. The data set includes 42 features and 494021 cases with 25 predictors that were assigned to five different classes: DoS, probes, user to remote attack (U2R), remote to local (R2L), and normal. Data pre-processing, classifications, and evaluation are three steps in the work.



```
----DECISION TREE-----
-----Confusion Matrix-----
[[3483   15]
 [  25 4035]]
-----Classification Report-----
              precision    recall  f1-score   support

     anomaly       0.99      1.00      0.99      3498
      normal       1.00      0.99      1.00      4060

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558

-----BNB-----
-----Confusion Matrix-----
[[2981  517]
 [ 188 3872]]
-----Classification Report-----
              precision    recall  f1-score   support

     anomaly       0.94      0.85      0.89      3498
      normal       0.88      0.95      0.92      4060

    accuracy                           0.91      7558
   macro avg       0.91      0.90      0.91      7558
weighted avg       0.91      0.91      0.91      7558

-----KNN-----
-----Confusion Matrix-----
[[3458   40]
 [  23 4037]]
-----Classification Report-----
              precision    recall  f1-score   support

     anomaly       0.99      0.99      0.99      3498
      normal       0.99      0.99      0.99      4060

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558
```

Figure 8 : Confusion Matrix & Classification Report of Algorithm

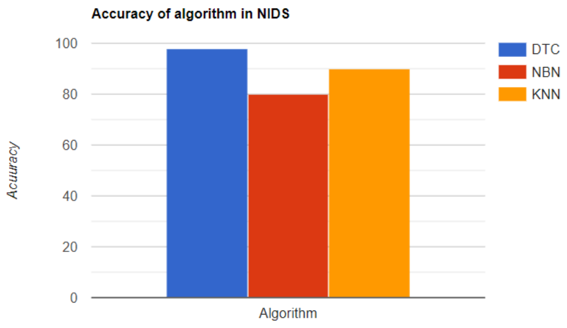


Figure 9: Accuracy of algorithm

IX.CONCLUSION

In this research, we focus on the use of machine learning methods and their applications to identify intrusion detection systems. The following are the four goals of this study review paper: I make recommendations for researchers who are new to the machine learning field and want to contribute to it; ii) present a state-of-the-art overview of machine learning; iii) provides further research directions required into intrusion detection system using machine learning

X. Future Work

Future work will cope with vast volumes of data, and a hybrid multilevel model will be built to improve accuracy. It is concerned with developing a simplified model backed by well-organized classifiers capable of categorizing new attacks with improved performance.

XI. REFERENCES

[1]. Aburomman, A. A., &Reaz, M. B. I. (2016) "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection."Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC): 636-640.

[2]. Al-Jarrah, O. Y., Al-Hammdi, Y., Yoo, P. D., Muhaidat, S., & Al-Qutayri, M. (2018) "Semi-supervised multi-layered clustering model for intrusion detection." Digital Communications and Networks 4(4): 277-286.

[3]. Al-Yaseen, W. L., Othman, Z. A., &Nazri, M. Z. A. (2017) "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system." Expert Systems with Applications 67(1): 296-303.

[4]. An, X., Su, J., Lü, X., & Lin, F. (2018) "Hypergraph clustering model-based association

analysis of DDOS attacks in fog computing intrusion detection system.” EURASIP Journal on Wireless Communications and Networking 249 (1): 1-9.

[5]. Belavagi, M. C., &Muniyal, B. (2016) “Performance evaluation of supervised machine learning algorithms for intrusion detection.” Procedia Computer Science 89(1): 117-123.

[6]. Elrawy, M. F., Awad, A. I., &Hamed, H. F. (2018) “Intrusion detection systems for IoT-based smart environments: a survey.” Journal of Cloud Computing 7 (1): 21

[7]. Elsaeidy, A., Munasinghe, K. S., Sharma, D., &Jamalipour, A. (2019) “Intrusion detection in smart cities using Restricted Boltzmann Machines.” Journal of Network and Computer Applications 135 (1): 76-83.

[8]. Shetty Akshada, Jadhav Shailaja et. al., “Detection of fake accounts in online social networks (OSN)” International Journal of Modern Trends in Engineering and Science IJMTES 2017, Volume 4 -Issue 5 Pages 1-3.

Cite this article as :

Shailaja Jadhav, Varsha Yadav, Vinaya Bhalerao, Bhavana Shinde, Snehal Kambale, "Network Intrusion Detection System Using Machine Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 3, pp. 208-216, May-June 2022. Available at doi : https://doi.org/10.32628/CSEIT228329
Journal URL : https://ijsrcseit.com/CSEIT228329