Machine Learning-Based Model for Intrusion Detection System

Olaniyi Ayeni, Dorcas Oluwasanmi
Department of Cyber Security
Federal University of Technology
Akure, Nigeria

Abstract

With the growing utilization of Internet resources, cyber attackers employ various methods to target network services, including unauthorized access, security breaches, and system misuse. Consequently, network security has become an indispensable component of network systems. To identify such attacks efficiently and effectively, an Intrusion Detection System (IDS) is necessary. Gurung et al. [1] attempted to develop a system using deep learning techniques for intrusion detection, which not only learns from known patterns but also adapts to previously undefined patterns. In their research, two machine learning algorithms, Gaussian Naïve Bayes (GNB) and Decision Tree (DT), were employed for data classification. The NSL-KDD dataset served as the training and testing dataset for these machine learning models. Univariate feature selections using ANOVA F-test were applied to eliminate irrelevant and unnecessary features from the dataset. The Second Percentile method from sklearn.feature _selection was utilized to select features based on the highest scores' percentile. Once this subset was identified, Recursive Feature Elimination (RFE) was implemented. Consequently, feature selection reduced the dataset's dimensionality, leading to a decrease in computational complexity. The performance of the proposed model was assessed using two randomly selected feature subsets from the NSL-KDD dataset. The training and testing results for the Decision Tree and Naïve Bayes algorithms were 99.90% and 90.11%, and 98.63% and 85.53%, respectively.

Keywords: Intrusion Detection System, Decision Tree, Naïve Bayes, Machine Learning, Cyber-attacks

1. Introduction

In the rapidly evolving landscape of cybersecurity, safeguarding computer networks from unauthorized access and malicious activities is of paramount importance. Intrusion Detection Systems (IDS) play a pivotal role in fortifying the security posture of these networks. This project endeavors to enhance the efficacy of intrusion detection through the integration of sophisticated machine learning models – Decision Tree and Naive-Bayes.

The proposed Intrusion Detection System employs a multi-faceted approach that leverages the strengths of both Decision Tree and Naive-Bayes algorithms. Decision Tree, known for its intuitive rule-based structure, excels in discerning patterns and classifying network behaviors. On the other hand, Naive-Bayes, based on probabilistic reasoning, provides a robust framework for analyzing the likelihood of different network events.

In this research, we aim to develop an adaptive and proactive defense mechanism against evolving cyber threats. By harnessing the predictive capabilities of machine learning, the Intrusion Detection System aspires to minimize false positives, enhance detection accuracy, and fortify networks against a spectrum of cyber-attacks. This project stands at the intersection of cybersecurity and artificial intelligence, contributing to the ongoing efforts to create resilient and intelligent defense mechanisms in the digital realm.

2. Review of Related Works

The contributions made by various researchers are discussed in this section. Palmer [2] has put forward the significance of intrusion detection in network infrastructures. However, rule-based intrusion detection systems (IDSs) can sometimes pose challenges and require significant time investment for maintenance. To address this issue, numerous machine learning algorithms have been explored in recent years to assist with intrusion detection. These techniques rely on training datasets, with many researchers utilizing the KDD '99 dataset. For these techniques to perform effectively in real-world scenarios, it is crucial to train them using realistic datasets. This paper aimed to determine the suitability of the KDD '99 dataset for such applications. Although the direct results of the experiment testing the validity of the KDD '99 dataset were inconclusive, they prompted further investigation of the data. Upon examining the dataset itself and comparing its structure to that of simulated attacks, it became apparent that the KDD '99 dataset may not be the optimal choice for training machine learning algorithms intended for real-world applications. Most

 | DOI: 10.20533/jitst.2046.3723.2023.0099

works in the field of intrusion detection predictive modeling employ similar types of datasets for training and testing. However, it is challenging to generalize real-time events based on these datasets. Consequently, the performance of the majority of predictive models tends to decrease when applied to actual network traffic.

Several approaches have been proposed to classify normal connections with anomalies for detecting network intrusions. In a detailed analysis conducted by Revathi et al. [3] using the NSL-KDD dataset, only relevant features were considered, both with and without feature reduction. Various classification algorithms, including J48 decision tree, Random Forest, Support Vector Machine, and Naive Bayes, were employed. The Random Forest algorithm achieved the highest test accuracy in both cases.

In their paper titled "A hierarchical network intrusion detection system using statistical pre-processing and neural network classification," Zhang et al. [4] developed a statistical neural network classifier capable of identifying UDP flood attacks. Among various neural network classifiers compared, the back propagation neural network (BPN) demonstrated greater efficiency for IDS development. Another study by Roy-I Chang et al. titled "Intrusion detection by back propagation network with sample query and attribute query" utilized the back propagation method with Sample Query and Attribute Query for Intrusion Detection. This approach analyzed and identified the most crucial components of the training data, resulting in reduced processing time and storage requirements.

Regarding machine learning-based IDSs, Panda et al. [5] proposed a new hybrid intelligent system that combined naive Bayesian with decision tree and rule-based classifiers. This system employed non-nested generalized samples and extended repeated incremental pruning. In another study by Sainis et al. [6] titled "Feature Classification and Outlier Detection to Increased Accuracy in Intrusion Detection System," the authors compared the effectiveness of newer datasets such as NSL-KDD and GurKDDcup with the older and frequently used DARPA KDD Cup 1999 dataset. They examined the number of available features in each dataset and assessed their relevance in detecting specific types of attacks. Additionally, five machine learning classification algorithms (SVM, Naive Bayes, KNN, Decision Tree-based C4.5 or J48, and Random Forest Algorithm) were used to address the attack classification problem and simplify the complexity of the classification models.

In a paper by Neyole et al. [7] titled "An Intrusion Detection System Based on a Simplified Residual Network," a novel IDS was proposed. The system consisted of a data preprocessing module, a random oversampling module, an S-Resnet layer, a fully connected layer, and a Softmax layer. This architecture aimed to prevent overfitting, improve the model's generalization ability, and reduce the parameters and computational requirements. The proposed IDS exhibited better performance in terms of accuracy and recall, particularly for R2L and U2R attacks, compared to existing IDSs.

Hang Xu et al. [8] conducted a study titled "Machine Learning Enhanced Real-Time Intrusion Detection Using Timing Information." Their research involved two analyses: Timing Analysis and Third Party Model Verification. These analyses contributed to a security monitoring system that validated worst-case timing bounds of the target controller and compared its control outputs against model-based predictions derived from machine learning techniques

Debicha et al. [9] presented a research titled "Efficient Intrusion Detection Using Evidence Theory" where they utilized Dempster-Shafer theory, a framework for reasoning under uncertainty, to construct an evidential classifier. They evaluated the performance of their approach using the NSL-KDD dataset, an improved version of the KDDCUP'99 dataset. Their approach outperformed some state-of-the-art methods on the more challenging KDDTest-21 dataset, while producing comparable results on the KDDTest+ dataset.

Gurung et al. [1] proposed a deep learning approach for network intrusion detection using the NSL-KDD dataset. Their system utilized a deep network to train on anomaly patterns and classify network traffic as normal connections or intrusions. Unsupervised feature learning was performed using a sparse auto-encoder, followed by classification using a logistic classifier. The system showed promising results in terms of accuracy, precision, and recall, indicating potential for future use and modifications.

Almseidin et al. [10] conducted an evaluation of machine learning algorithms for intrusion detection systems. They implemented various algorithms, including J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network, to assess the model's accuracy based on the KDD dataset. Different types of attacks (DOS, R2L, U2R, and PROBE) were considered, and the classifiers' efficiency and performance were tested. The dataset contained approximately 79% DOS attacks, 19% normal packets, and 2% other types of attacks. The experiments utilized 148,753 instances of records as training data for the selected machine learning classifiers.

Mrutyunjaya Panda et al. [11] proposed a network intrusion detection system based on the Naïve Bayes algorithm. Their framework constructed network service patterns using labeled datasets and detected attacks using the Naïve Bayes Classifier algorithm. Compared to a neural network-based approach, their system achieved higher detection rates, lower processing time, and lower costs. However, it generated slightly more false positives. To address

 | DOI: 10.20533/jitst.2046.3723.2023.0099

this limitation, the researchers suggested considering an active platform or event-based classification using Bayesian networks to reduce false positives and improve efficiency.

Kalekar et al. [12] presented a learning algorithm for network intrusion detection using the Naive Bayesian approach. Their algorithm achieved balanced detection and maintained acceptable levels of false positives for different types of network attacks. It also addressed challenges in data mining such as handling continuous attributes, missing attribute values, and reducing noise in training data. The proposed system utilized the KDD'99 dataset and employed the Naive Bayes algorithm for real-time detection.

Raeeyat et al. [13] proposed a system with four modules: Data pre-processing, Misuse detection, Anomaly detection, and Evaluation and comparison. The data underwent pre-processing before being analyzed by the Misuse detection module, where important features were extracted using PCA. The data were then examined using the Adaboost algorithm based on the C4.5 decision tree to determine if it was a normal packet or an intrusion. The Anomaly detection module analyzed data simultaneously, while the Evaluation and comparison module determined if an instance was an intrusion by considering the outputs from both the Misuse and Anomaly detection modules.

3. Motivation/ Research Gap

The increasing and evolving targeted network attacks pose a threat to businesses, leading to the need for enhanced network security systems to prevent potential data and capital losses. Intrusion Detection Systems (IDS) play a crucial role in any security system by actively detecting unauthorized access attempts that can compromise the basic TRIAD, that is; confidentiality, availability, and integrity of computer networks. In their research titled "Deep learning approach on network intrusion detection system using nsl-kdd dataset," Gurung et al. [1] aimed to develop an IDS using deep learning techniques that can learn and adapt to previously undefined patterns. The objective of this work was to design an Intrusion Detection System using a machine learning approach that can identify all potential threats and signs of intrusions while minimizing false alarms. The approach demonstrates flexibility in adapting to new intrusion patterns and changes in user behavior over time.

4. System Analysis and Design

In the previous section of this research, an extensive literature survey was conducted on different methods and techniques used in intrusion detection. This chapter focuses on essential concepts related to intrusion detection, including data sets, training and testing data sets, sampling, modeling, knowledge discovery, validation, predictive models, accuracy, and visualization. The implementation methods of some intrusion detection techniques will also be discussed. Furthermore, a comparative analysis of these techniques will be conducted, considering their advantages, disadvantages, speed of detection, scalability, cost, and accuracy

4.1. Machine Learning

Machine learning is a branch of artificial intelligence that makes use of probabilistic, statistical, and optimization techniques to train computer systems in exploring and learning complex models from large, noisy data. It enables systems to improve their performance based on past experiences and make intelligent decisions or accurate predictions, such as identifying harassment based on observations. The main goal is to develop learning algorithms that can automatically perceive, learn and act without human intervention. Machine learning is particularly useful when humans are prone to analysis errors or when establishing connections among multiple functions. It offers an alternative approach to intrusion detection by utilizing past real-world data to predict intrusions and enhance system and engine designs.

4.2. Supervised Learning

Supervised learning is a method of machine learning that involves using labeled instances to train a model. It utilizes (x1) as an input variable and (y1) as its output variable, employing classification algorithms to learn the mapping function between the input and output. Equation 1 represents this mapping as follows:

$$y1 = f(x1) \quad (1)$$

Supervised learning problems can be further categorized into classification and regression problems.

4.3. Naïve Bayes

Naïve Bayes algorithm is a type of supervised machine learning classifier. It is a widely used and simple algorithm based on Bayes' theorem. The Naïve Bayes classifier assumes that the occurrence of a specific feature in a class is completely independent of the presence of any other feature. It is straightforward to build and particularly effective for larger datasets. The name "naïve" is derived from the simplified assumptions it makes about the attributes. The Naïve Bayes algorithm is a probabilistic algorithm represented by Equation 2:

 | DOI: 10.20533/jitst.2046.3723.2023.0099

$$p\left(\frac{c_j}{x_j}\right) = \frac{p(x_j/c_j)p(c_j)}{p(x_j)} \quad (2)$$

when given:

$p\left(\frac{c_j}{x_j}\right)$ denotes the posterior probability, which represents the probability of c_j occurring when given the evidence that x_j has already occurred.

$p(c_j)$ represents the prior probability of the class, which indicates the probability of c_j occurring.

$p(x_j/c_j)$ signifies the prediction probability of a class, denoting the probability of x_j occurring given the evidence that c_j has already occurred.

$p(x_j)$ corresponds to the prior prediction probability of a class, indicating the probability of x_j occurring. The Naïve Bayes classifier exhibits a relatively low error rate compared to more complex models.

The error rate in the Bayes classifier is relatively low when compared to other complicated models.

4.4. Decision Tree

The Decision Tree Algorithm is a supervised machine learning algorithm, which goal is to create a model that predict values of a variable, where data is repeatedly divided at each row based on specific decision rules until the final outcome is achieved.

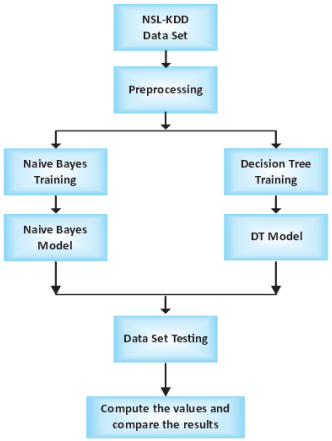

Figure 1. Sequence Diagram of the Model

It continuously splits the data based on predetermined criteria, creating a tree-like structure.

$$Gini(P) = \sum_{i=1}^{n} p_i(1-p_i) = 1 - \sum_{i=1}^{n} (p_i)^2$$
$$where\ P = (p_1, \ldots, p_n). \quad (3)$$

The ultimate outcome is determined by traversing the tree using the learned rules from the training data.

4.5. NSL-KDD Dataset

The NSL-KDD data set is an improved version of the well-known KDD-cup99 dataset, where redundant entries that were causing biased classification results have been removed. This dataset is publicly available, and during the preprocessing phase, category encoding is applied to convert the dataset's nominal features into numerical values. This is necessary because some models cannot directly handle nominal values. The main objective of selecting and testing this dataset is to compare different techniques and develop an efficient, and proactive intrusion detection model.

The NSL-KDD dataset groups intrusion or attacks into several categories, which are discussed below:

A. Probing or Probe: This is considered a significant threat, and the NSL-KDD dataset classifies it as an attack. Probing attacks can be identified based on their features, such as source-bytes and connection duration.

B. User to Root (U2R): The U2R attack involves an attacker attempting to gain administrative control and privileges by exploiting vulnerabilities in a system or network. Unauthorized access to super local privileges is considered an attack, and any detection of such attacks should trigger an alarm.

C. Denial of Service (DoS) Attack: Denial of Service is a severe threat that aims to exhaust the resources of a network or system. In the NSL-KDD dataset, DoS attacks are categorized as a type of attack or anomaly.

D. Root to Local (R2L) Attack: The R2L attacks, infiltration of a system or network is carried out with the use of a remote machine. The NSL-KDD dataset enables the identification of such attacks based on the source of origin, connection duration, and requested service.

For experimentation purposes, the NSL-KDD dataset is divided into two datasets, namely, training and testing datasets. The training data set, KDDtrainPlus, consists of 125,973 records, while the testing data set, KDDtestPlus, contains 22,543 tuples (see Table 1). The Algorithm used to Build the Model consists of:

i. Pre-process the dataset.

ii. Divide the data set into training data and testing data

iii. Build the classifier model on training dataset for:

- Decision Tree

 | DOI: 10.20533/jitst.2046.3723.2023.0099

- Gaussian Naive Bayes

iv. Read the test data

v. Test the classifier models on training data

vi. Compute and compare Accuracy, Precision, Recall, and F1-Score for all

Table 1. Confusion Matrix of Actual vs Predicted Class

Confusion Matrix		Predicted Class	
		Attack	Normal
Actual Class	Attack	TP	FN
	Normal	FP	TN

5. Results

Data preprocessing is the initial step performed before commencing any process. Its purpose is to convert and transform raw, unfiltered data into a more suitable, understandable, and usable format. Real-world data often exhibits incompleteness, inconsistency, inaccuracy, unstructured form, as well as errors and missing values. Data preprocessing addresses these issues by cleaning, formatting, and organizing the raw data, making it ready for use in building machine learning models. Since data preprocessing involves multiple tasks, it cannot be accomplished in a single process. One of the steps involved is the conversion of all features into numerical values using one-hot encoding. Additionally, feature scaling is applied to prevent features with large values from exerting excessive influence on the results. The following are the steps typically undertaken in data preprocessing: importing libraries and performing version checks, loading the dataset, examining the shape of the train and test datasets, inspecting the labels of the training and test datasets, and so on.

```
In [2]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import sys
        import sklearn
        print(pd.__version__)
        print(np.__version__)
        print(sys.version)
        print(sklearn.__version__)

        1.2.4
        1.21.6
        3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)]
        0.24.2
```

```
In [31]: # first five rows
         df_train.head(5)
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_h
0	0	0	tcp	ftp_data	SF	491	0.1	0.2	0.3	0.4	...	25	
1	1	0	udp	other	SF	146	0.0	0.0	0.0	0.0	...	1	
2	2	0	tcp	private	S0	0	0.0	0.0	0.0	0.0	...	26	
3	3	0	tcp	http	SF	232	8153.0	0.0	0.0	0.0	...	255	
4	4	0	tcp	http	SF	199	420.0	0.0	0.0	0.0	...	255	

5 rows × 43 columns

Confusion Matrices: DoS, Probe, R2L and U2R respectively

Predicted attacks	0	1
Actual attacks		
0	9602	109
1	2625	4835

```
Y_R2L_pred2=clf_rfeR2L.predict(X_R2L_test2)
# Create confusion matrix
pd.crosstab(Y_R2L_test, Y_R2L_pred2, rownames=['Actual attacks'
```

Predicted attacks	0	3
Actual attacks		
0	9649	62
3	2560	325

Predicted attacks	0	2
Actual attacks		
0	8709	1002
2	944	1477

```
Y_U2R_pred2=clf_rfeU2R.predict(X_U2R_test2)
# Create confusion matrix
pd.crosstab(Y_U2R_test, Y_U2R_pred2, rownames=['Actual attacks']
```

Predicted attacks	0	4
Actual attacks		
0	9706	5
4	52	15

Table 2. Cross Validation: Accuracy, Precision, Recall and F-Measure (approx. 4 dp)

	Accuracy	Precision	Recall	F-Measure
DoS	0.9964	0.9951	0.9967	0.9959
Probe	0.9957	0.9939	0.9928	0.9933
R2L	0.9795	0.9722	0.9698	0.9709
U2R	0.9966	0.8648	0.9167	0.8863

Recursive Feature Elimination (RFE)

This is a feature selection method that fits into a model and is used to eliminate all the non-essential features that are present or until the specified number of features are reached in the dataset (see Figures 2, 3, 4 and Table 3).

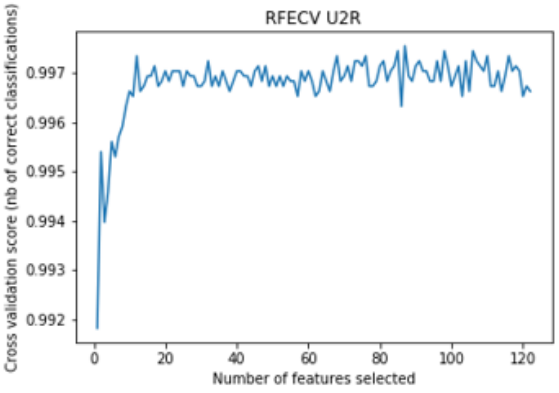

Figure 2. Recursive Feature Elimination curve for U2R

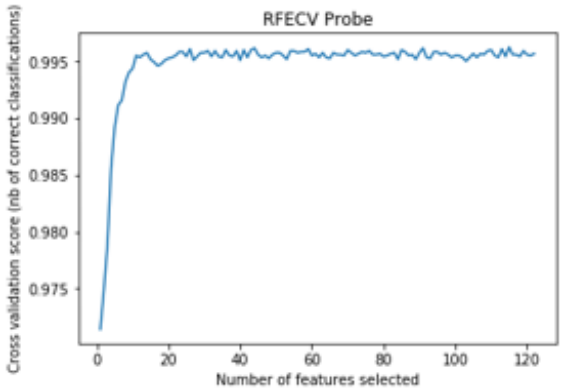

Figure 3. Recursive Feature Elimination curve for Probe

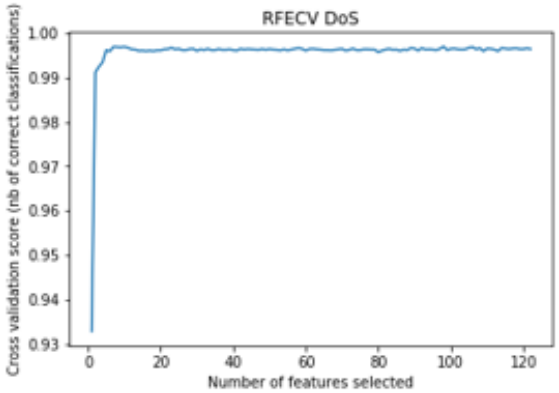

Figure 4. Recursive Feature Elimination curve for DoS

Table 3. Evaluation with Previous Work

Authors	Precision	Recall	Accuracy
This work	96.9%	97.2%	97.6%
Sandeep Gurung et al	84.6%	92.8%	87.2%
Mahmood Mohamed Sakr et al.	99.08%	99.12%	99.10%

6. Conclusion

Learning algorithm was introduced for intrusion detection system that utilizes the Naive Bayesian classifier and Decision Tree model. This algorithm achieves balanced detections and maintains an acceptable level of false positives for different types of network attacks while eliminating redundant attributes. Additionally, the proposed algorithm tackles certain challenges in data mining, including handling continuous attributes, managing missing attribute values, and reducing noise in training data. To train and test the machine learning models (Naïve Bayes and Decision Tree), we employed the NSL-KDD dataset. Feature selection techniques were employed to remove irrelevant, redundant and unwanted features from the data set. This process reduces the dataset's dimensionality, subsequently decreasing computational complexity.

7. References

[1] Gurung, S., Ghose, M. K., and Subedi, A. (2019). Deep learning approach on network intrusion detection system using nsl-kdd dataset. International Journal of Computer Network and Information Security (IJCNIS), vol. 11, no. 3, 2019, pp. 8–14.

[2] Palmer, J. (2011). Naive Bayes Classification for Intrusion Detection Using Live Packet Capture. Data Mining in Bioinformatics, Spring.

[3] Ravathi, S., Malathi, A. (2013). A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. International Journal of Engineering Research and Technology (IJERT), ISSN: 2278-0181. Vol. 2 Issue 12, Dec.

[4] Zheng Zhang., Jun Li., Manikopoulos, C.N. Jorgebson, Jose Ucles, J. (2001). HIDE: A hierarchical network intrusion detection system using statistical pre-processing and neural network classification", IEEE workshop proceedings on Information assurance and security, 2001, pp.85-90.

[5] Panda, M., Abraham, A., Patra, M.R. (2015). Hybrid intelligent systems for detecting network intrusions. Secur. Commun. Netw. 2015, 8, 2741–2749.

[6] Nachiket Sainis et al, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 10 (2018) pp. 7249-7255. Research India Publications. http://www.ripublication.com (Access Date: 21 May 2023).

[7] Neyole et al. (2019). Institute of IOT and IT-based Industrialization, Xi'an University of Post and Telecommunications, Xi'an 710061, China, Shaanxi Provincial Information Engineering Research Institute, Xi'an 710075, China.

[8] Hang Xu H., and Mueller, F., Acharya, M., and Kucheria, A. (2018). Machine Learning Enhanced Real-Time Intrusion Detection Using Timing Information. North Carolina State University, USA.

[9] Debicha, I., Debatty, T., Mees, W., Dricot, J-M. (2021), Efficient Intrusion Detection Using Evidence Theory. arXiv:2103.08585v1 [cs.LG] (Access Date: 15 March 2024).

[10] Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M. (2017). Evaluation of machine learning algorithms for intrusion detection system. IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). Subotica, Serbia. DOI: 10.1109/SISY.2017.8080566.

[11] Panda, M., and Patra, M. R. (2007). Network Intrusion Detection Using Naïve Bayes. IJCSNS International Journal

 | DOI: 10.20533/jitst.2046.3723.2023.0099

of Computer Science and Network Security, VOL.7 No.12, December.

[12] Kalekar, A., Kshatriya, N., Chakranarayan, S., and Wadekar, S. (2014). Real Time Intrusion Detection System using Machine Learning. International Journal of Engineering Research and Technology (IJERT) Vol. 3 Issue 2, February – 2014.

[13] Raeeyat, A., and Sajedi, H. (2012). HIDS: DC-ADT: An Effective Hybrid Intrusion Detection System based on Data Correlation and Adaboost based Decision Tree classifier, Journal of Academic and Applied Studies, vol. 2(12), Dec, pp.19-33.

Further Readings

Wikipedia Encyclopedia. (2016). Anomaly-based intrusion detection system. https://en.wikipedia.org/wiki/ Anomaly based_intrusion_detection_system (Access Date: 20 Decem ber 2023).

Ayeni O. A., Ewa Stanley C., Otasowie, O. (2022). Intrusion Detection System using Deep Learning. International Conference for Internet Technology and Secured Transactions (ICITST), London, UK.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

Sakr, M. M., Tawfeek, M., El-Sisi, A. (2019). Network Intrusion Detection System based PSO-SVM for Cloud Computing. https://www.researchgate.net/publication/33 1867649 I.J. Computer Network and Information Security, 3, 22-29.

Belavagi. C. M., and Balachandra Muniyal. (2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016).

Ayeni, O. A. (2022). A Supervised Machine Learning Algorithm for Detecting Malware. Journal of Internet Technology and Secured Transactions (JITST), Volume 10, Issue 1, 2022.

Chang, R-I et al., (2017). Intrusion detection by back propagation network with sample query and attribute query. International Journal of computational Intelligence Research.

Mei-Ling, S., Chen, S-C., Sarinnapakorn, K., Chang, L-W. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. Standard Form 298 (Rev.8-98).

Jyothsna, V., Rama Prasad, V. V., Munivara Prasad, K. (2011). A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications, pp. 26-36.

 | DOI: 10.20533/jitst.2046.3723.2023.0099