DATA ACQUISITION AND PRE-PROCESSING USING KF MODEL FOR AN INTRUSION DETECTION SYSTEM IN WEB MINING

Sagar Babu Jeldi[1], Dr.Ashok Kumar P.S.[2]

Article History: Received: 26.02.2023 **Revised:** 12.04.2023 **Accepted:** 29.05.2023

ABSTRACT

Intrusion Detection Systems (IDS) plays an important role in web mining to detect and prevent unauthorized access, attacks, and anomalies in web-based systems. Monitoring unauthorized or malicious activity on networks or systems is the purpose of IDS. Building an effective intrusion detection system for web mining is a complex task that requires expertise in web security, data analysis, and machine learning. By implementing data mining in a cyber-attack intrusion detection system, organizations can sense and react to threats more rapidly and effectively, reducing the hazard of harm and loss. Data mining can also provide insights into the nature and characteristics of cyber-attacks, which can inform the development of more effective security measures. In this paper, log files are collected and preprocessed using Kalman filter with the combination of Correlation-based Feature Selection (CFS). The experimental findings demonstrate that such pre-processing and CFS combination is effective are applied with different classification algorithms in machine learning models using KDD'99 dataset.

Keywords: Intrusion Detection Systems, Web Mining, Kalman Filter, Machine Learning, Preprocessing, CFS.

[1]Research Scholar, Don Bosco Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University. Email: bbbsag@gmail.com

[2]Professor & H.O.D, Dept. of CSE, H.K.B.K College of Engineering.

Email: ashokdbit2017@gmail.com

DOI: 10.31838/ecb/2023.12.s3.428

1. INTRODUCTION

Data acquisition systems are becoming more sophisticated as they combine cutting-edge technology that includes the IoT. As a consequence of these developments, the system grew more effective and easier to use, but it also became more vulnerable to cyber-attacks. IoT is developing as novel technologies aimed at the creation of a variety of vital applications. Nevertheless, these kinds of apps are still based on centralised data architecture and face a number of significant concerns such as security, confidentiality, and single point of failure. Because current networks are complex and security attacks are on the rise, machine learning (ML) has become the most popular IDS technique.

Cyber-attacks are growing more complex, making it more difficult to identify breaches. Failure to prevent invasions may erode security services' credibility. One of the most successful models for identifying attack behaviours is intrusion detection systems (IDS) [1]. The most visible challenges in network security is the growth of malicious behaviour in network traffic. This practise reduces the efficiency of numerous organisations and end-users. In addition to Internet use, various connected devices share a great deal of information. In today's network landscape, network safety is a dominant study area because data should be exchanged securely between interacting devices. Due to this, IDS are often integrated with other security systems such as firewalls and access controls.

IDS is a vital module of network security that aids in the detection and prevention of unauthorized activities and attacks on computer systems and networks [2]. With the increasing complexity and frequency of cyber threats, IDS plays a dynamic role in safeguarding penetrating information and confirming the reliability of network infrastructure.

The determination of IDS is to observe and analyze network traffic, system logs, and user activities to identify any signs of malicious behavior or intrusion attempts. It employs various methods, such as signature-based detection, irregularity detection, and behavior exploration, to recognize known attack patterns and deviations from normal system behaviour [3].

IDS can be categorized into two main types: network as well as host-based IDS. NIDS monitors network traffic in real-time, analyzing packets and looking for suspicious patterns or signatures of known attacks [4]. HIDS, on the other hand, is installed on individual hosts and focuses on detecting unusual activities or unauthorized access at the host level, such as file modifications, login attempts, or system calls.

Upon detecting a potential intrusion or security breach, IDS generates alerts or triggers appropriate actions, such as notifying system administrators, blocking network traffic from suspicious sources, or activating additional security measures [5]. IDS is often complemented with Intrusion Prevention Systems (IPS), which not only detect but also actively prevent or mitigate detected threats.

The effectiveness of an IDS depends on factors such as the quality and accuracy of the underlying detection algorithms, the comprehensiveness of the attack signature database, and the capability to adapt to evolving attack techniques. Continuous monitoring, regular updates, and a thorough understanding of the network environment are essential for maintaining a robust and reliable IDS.

RELATED WORKS

Examining and putting into practise various security measures are required to ensure the safety of these vital facilities.

These solutions may be based on hardware, such as IDS, management, software respectively. In this study [6], the authors analysed and presented the most current research on the topic of employing Deep Learning to construct reliable IDSs. An assault on a mining pool revealed key fault points inside a blockchain-enabled internet of things network [7]. In addition, this programme creates an extremely large volume of data. Data analytics can be done autonomously with machine learning as well as with decision-making skills, since ML is an autonomous method of studying large amounts of data.

An ML framework to detect attack variations on the Internet is presented by this study [8,] which combines variational automatic coding and multilayer perceptrons. An array of attack variants would be predicted using the framework and datasets that are imbalanced. An effective range-based sequential searching method is included as part of the detection engine. This allows the engine to tackle the segmentation difficulty that arises during the effective preprocessing of data coming from many sources. An intrusion detection system that is tailored specifically for Apache web servers is suggested in this research paper [9]. Training using the suggested technique is accomplished with the help of the Naive Bayes machine learning algorithm.

A hybrid intrusion detection system is suggested in this work [10]. A combination of packetas well as flow-based IDS, as well as Dempster-Shafer Theory, are used in the execution of the suggested method. To address these issues in IoT-enabled smart towns, this research [11] suggest an IDS that is centred on hybrid optimisation and deep learning. In the beginning, the dataset goes through pre-processing so that efficient and accurate IDS may be obtained. The method of optimising BP neural network using Adaboost algorithm that is presented in this research [12] has a superior average detection rate and detection speed than other algorithms when it comes to each sort of assault. It has been shown that the Adaboost algorithm can successfully handle the issue of intrusion detection.

For the purpose of classifying network assaults, this research [13] offer a unique method of Random Forest Algorithm. The approaches that have integrated feature selection are the primary topic of the study [14], which focuses on those methods. In addition to this, the study provides a full explanation of more current datasets as well as a discussion of the various IDS datasets. The ability of network managers to identify potentially dangerous traffic on the network that is created by malware and other malicious instruments is inhibited. An explainable artificial intelligence solution was created by the authors of this research [15] as part of a revolutionary machine learning architecture in order to assist them in their attempts to keep the network safe.

In this work, new algorithms for intelligent rules-based characteristics collecting and rules-based enhanced vector support computer were provided. Additionally, a study of the currently available intelligent methodologies for intrusion detection systems was presented [16]. To be more specific, preprocessing as well as entropy method are carried out in order to improve data quality and relevant training. Following this, a decision tree classifier is constructed in order to reliably identify intrusions [17].

The use of the internet has rapidly evolved into one of the most significant everyday activities across a variety of domains [18]. There is a significant and growing population of consumers who rely on the internet to carry out their commercial and shopping activities. An effective kind of computer security technology is known as an intrusion detection system [19]. This type of system is able to detect, prevent, and perhaps respond to computer-related hostile activity. Current Intrusion

Detection Systems were designed with obsolete attack datasets in prediction latency [20].

2. PROPOSED METHODOLOGY

In this proposed research work, Kalman filter is used for preprocessing method and CFS technique used for feature selection. An experimental output is tested using different ML algorithms. Preprocessing, Feature Selection, and Classification are the three components that make up the system that was built. The overall system architecture is shown in Figure 1.

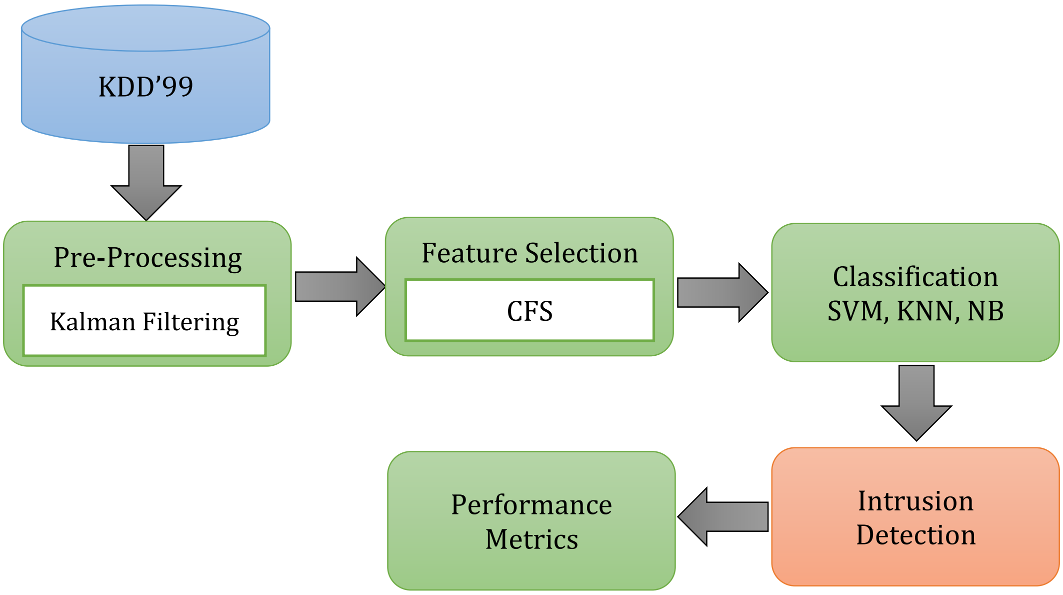

Figure 1: Architecture Diagram

2.1 Dateset Description

In IDS model, KDD'99 dataset are frequently utilised. It was developed in 1999 for the 3rd International Knowledge Discovery and Data Mining Tools Competition. The dataset was developed to provide a benchmark for evaluating IDS and techniques. The KDD'99 dataset is based on the DARPA 1998 Intrusion Detection Evaluation Program, which simulated a military network environment with various types of attacks. The dataset comprises of network traffic data taken from a simulated network environment, including both normal and attack instances. The original KDD'99 dataset contains approximately 4.9 million network connections, which are classified into five main categories:

Normal: Represents normal connections that do not correspond to any known attack types.

DoS (Denial of Service): Denotes attacks that aim to overwhelm a network or system, rendering it unavailable.

Probe: Refers to activities that try to gather information about a target network, such as port scanning or reconnaissance.

U2R (User to Root): Represents attacks where an unauthorized user attempts to gain root-level access to a system.

R2L (Remote to Local): Denotes attacks where an unauthorized remote user tries to gain local access to a system.

The dataset provides a wide range of features or attributes for each network connection, including protocol type, service, source and destination IP addresses, source and destination port numbers, connection duration, and more.

2.2 Preprocessing

In this proposed framework, Kalman filtering is employed for data preparation to reduce noisy data before proceeding with the processing. KDD datasets typically include numeric and categorical values. Some outside parameters may cause the tuple values to be missing in some situations. As a consequence, the suggested Kalman filtering is primarily utilised for managing and processing missing information in order to prevent misleading outcomes. Typically, the missing values in the KDD dataset are discovered by deriving the equation and determining the covariance error. As a result, unambiguity may be minimised significantly and classification accuracy can be achieved. Furthermore, Kalman filtering lays the path for locating and projecting lost data.Furthermore, prediction is performed for the information while modifying the coefficient approximations of the Kalmanfilter to account for data loss. In the meantime, the data's mean value is being updated with the missing data.

The mathematical model for the Kalman filter is based on a set of linear equations that describe the dynamics of a system and the relationship between the system's states and measurements. The model consists of two main components: the state conversion equations and the dimension equations.

State Transition Equations: The state transition equations explain the evolution of the system's state across time. They can be represented as follows:

$$State\ Prediction: V(l) = M * V(l-1) + C * x(l-1) + n(l-1) \qquad (1)$$

Here, $V(l)$ represents the vector of state at time step l, M is the matrix of state transitions that describes the dynamics of the system, C is the control input matrix, x(l-1) is the vector of input control input in step l-1, and n(l-1) is the noise vector generated by the method represents the uncertainty or disturbances in the system.

Measurement Equations: The measurement equations describe how the measurements are related to the system's state. They can be represented as follows:

$$Measurement\ Prediction: z(l) = H * V(l) + n(l) \qquad (2)$$

Here, z(l) represents the measuring vector at time step l, H is the measurement matrix that corresponds to the measurements and maps the system's state., and v(l) is the measurement noise vector, which reflects measurement noise or mistakes.

The Kalman filter uses these equations to estimate the system's true state based on the available measurements. It is divided into two major steps: prediction and updating.

Prediction Step: In the prediction stage, the Kalman filter forecasts the present state depending on the previous state and the system dynamics. It involves the following equations:

$$State\ Prediction: x_{hat(n|n-1)} = M * x_{hat(l-1|l-1)} + C * n(l-1) \quad (3)$$

$$Error\ Covariance\ Prediction: P(l|l-1) = M * P(l-1|l-1) * M^T + Q(l-1) \ (4)$$

Here, x_hat(l|l-1) based on the preceding estimate, reflects the expected state estimate at time step l, P(l|l-1) is the predicted error covariance matrix, which indicates the predictability of the expected state., and Q(l-1) is the noise covariance matrix in the procedure.

Update Step: The Kalman filter utilises the data during the update stage to improve the state prediction. It consists of the following equations:

$$Kalman\ Gain: K(k) = P(l|l-1) * H^T * \left(H * P(l|l-1) * H^T + R(l)\right)^{-1} \quad (5)$$

$$State\ Update: x_{hat(l|l)} = x_{hat(l|l-1)} + L(l) * \left(z(l) - H * x_{hat(l|l-1)}\right) \quad (6)$$

$$Error\ Covariance\ Update: P(l|l) = (I - L(l) * H) * P(l|l-1) \qquad (7)$$

Here, L(l) represents the Kalman gain that determines the weight given to the measurements, R(l) is the matrix of measurement noise covariance, I is the matrix of identity, and x_hat($l|l$) and P($l|l$) are the updated state estimate and error covariance matrix, respectively.

The Kalman filter constantly enhances its state estimation based on fresh observations by repeatedly performing the prediction and update processes, incorporating both the system dynamics and the measurement information to provide an optimal estimation of the true state.

2.3 Feature Selection

In this step, feature selection is carried out by using the data set that was pre-processed in the previous stage. Following that, a CFS model is used to select features.

The correlation coefficient quantifies the strength and direction of a two-variable linear connection. Pearson's correlation coefficient is the most often utilised of the several forms of correlation coefficients. Pearson's correlation coefficient is calculated as follows:

$$r = (\Sigma((ai - \bar{a})bi - \bar{b}))) \,/\, sqrt(\Sigma((ai - \bar{a})^2) * \Sigma((bi - \bar{b})^2)) \qquad (8)$$

Where:

r is the coefficient of correlation

Σ represents the summation symbol

ai and bi are separate variables' data points

$\bar{a}$ and $\bar{b}$ are the difference between the two variables

In other words, the formula computes the total of the variances between every data point and the mean of the corresponding variable, multiplied by the product of the two variables' standard deviations.

Where r is the resultant correlation coefficient, distances from -1 to 1. If the value is one, then the variables are connected in a perfect linear manner, if it is one, then the variables are not connected. Values closer to -1 or 1 indicate a stronger correlation, with values closer to -2 suggesting a weaker correlation.

Calculating the correlation coefficient between two variables requires several calculations involving means, variances, and covariances. Here's a pseudocode representation of how to compute the correlation coefficient:

```
function calculateCorrelationCoefficient(a, b):
    n = length(a)  // number of data points
    // Calculate means of a and b
    mean_a = sum(a) / n
    mean_b = sum(b) / n
    // Calculate variances of a and b
    var_a = 0
    var_b = 0
    for i = 0 to n-1:
        var_a += (a[i] - mean_a) * (a[i] - mean_a)
        var_b += (b[i] - mean_b) * (b[i] - mean_b)
    var_a /= n
    var_b /= n
    // Calculate covariance of a and b
    cov = 0
    for i = 0 to n-1:
        cov += (a[i] - mean_a) * (b[i] - mean_b)
    cov /= n
    // Calculate correlation coefficient
    correlation_coefficient = cov / sqrt(var_a * var_b)
    return correlation_coefficient
```

In the above pseudocode, a and b represent the two variables for which the correlation coefficient is calculated. The pseudocode calculates the means of a and b, then iterates over the data points to calculate the variances and the covariance. In the end, the correlation coefficient is determined by divided the coefficient of covariance by the square root of the variance product.

Note that this pseudocode assumes that the input arrays x and y contain the same number of data points and that the necessary mathematical functions, such as sum() and sqrt(), are available for use.

2.4 Classification

Classification is the process of categorising data based on a label or target class. As a result, methods for solving categorization issues are classified as supervised learning. In this study, we examine the influence of our proposed framework on classification performance using three classification methods. In our study, we employ the k-Nearest Neighbour (k-NN) classification, Support Vector Machine (SVM), and Naive Bayes (NB) as a pilot. The k-NN classifies things using learning data that is closest to the item. The goal of this approach is to categorise new objects using characteristics and training examples. It is fairly basic and straightforward to apply, comparable to the clustering approach, which groups new data based on its distance to some existing data/nearest neighbour. Before calculating the distance between the data and the neighbour, the value of neighbouring k (neighbour) must be determined. The Euclidean formula is then employed to determine the distance between two points, namely the point in training and the point in testing. In this formula, d(a; b) represents the Euclidean distance, a represents the first data of the i feature, b represents the second data of the i feature, and n represents the total number of features.

$$d(a,b) = a_0 + \sum_{i=0}^{n} (a_i + b_i)^2 \qquad (9)$$

The SVM idea is basically a way for determining the optimum hyperplane to serve as a separator between two classes. In SVM, discriminating boundaries or alternative lines may be reduced to the two class members +1 and 1. The closest pattern is used to generate the support vector. As a result, the heart of the SVM algorithm is determining the position of the optimum hyperplane.

Naive Bayes is a supervised machine learning technique that conducts statistical classification by constructing a set of probability from the total number of the analysed dataset's frequency and value combinations. It is predicated on the idea that the significance of an attribute is an assortment of values that are not tied to each other when provided with the value of output. If it is designated, the probability will be watched concurrently in order to calculate the individual probability value.

3. RESULTS AND DISCUSSIONS

We conduct trials using machine learning methods. To improve accuracy, further testing is undertaken on the pre-processing data as well as feature selection. We separate the categories based on the available dataset: KDD Cup 1999.

We utilise the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) scores to evaluate the performance of each categorization. The data are then analysed to determine the accuracy, True Positive Rate (TPR), and

False Negative Rate (FPR). The system classifies TP as a successful attack; FP as a normal activity that is tagged as an attack by the system; TN as a normal flow that can be appropriately identified by the system; and FN as an attack flow that is not recognised by the system. The following describes the experimental outcomes on several datasets.

Table 1 contains the outcomes of the test results coming from this first scenarios of KDD'99 data. It shows that the SVM classification works best, with 99.96 % accuracy and 99 percent TPR.

Table 1: The first and second KDD Cup99 test results

Classifier	Results (%)		
	Accuracy	TPR	FPR
k-NN	98.62	98	2
SVM	99.96	99	1
Naïve Bayes	92.26	92	8

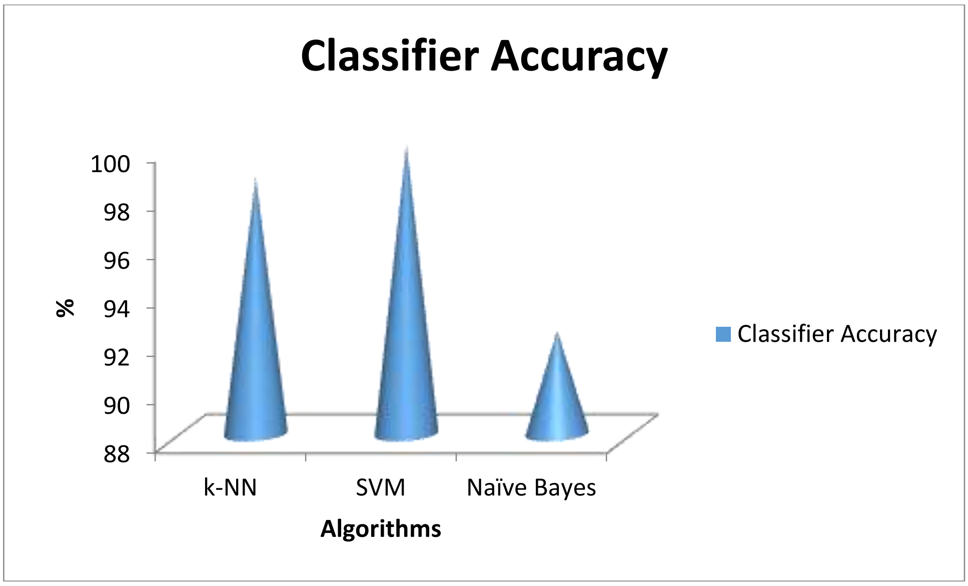

Figure 2: Classifier Accuracy

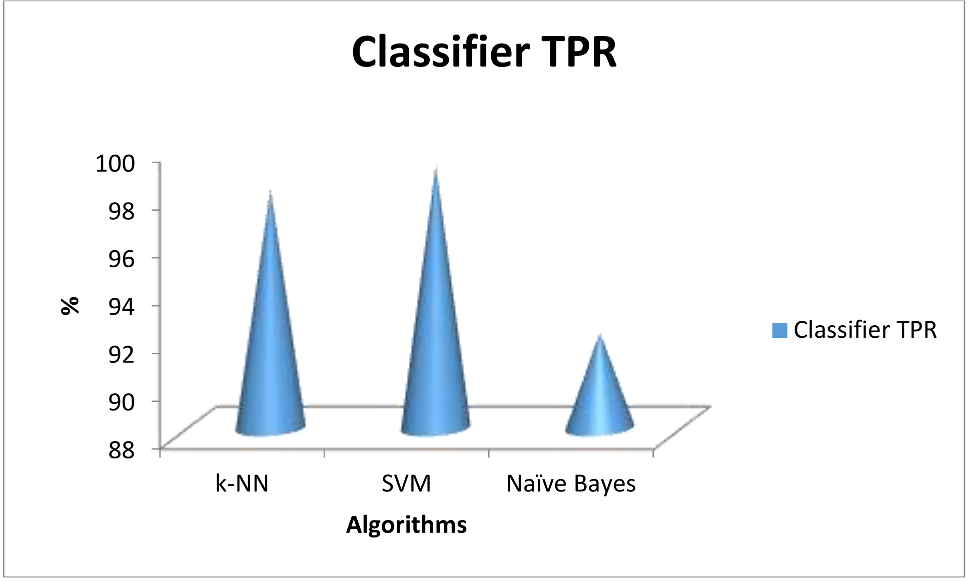

Figure 3: Classifier TPR

Figure 2 and 3 clearly shows that SVM classifier gives better accuracy for Intrusion Detection and provides higher TPR than other classifiers.

4. CONCLUSION

An IDS is a critical security tool that helps protect computer systems and networks from unauthorized access, attacks, and data breaches. By monitoring network and system activities, it plays a vital role in detecting and responding to potential threats, enhancing the overall security posture of organizations and ensuring the confidentiality, integrity, and availability of critical information resources. In this research, we presented a kalman filtering as preporcessing method and CFS for feature selection technique, which were then evaluated on distinct data set of KDD'99 using three different algorithms (k-NN, SVM, and Naive Bayes) for detecting intrusion in a webmining. Overall, it is shown that this procedure may increase efficiency in terms of accuracy, TPR, and FPR. In greater detail, SVM achieves the best result, with accuracy, TPR, and FPR of 99.96%, 99%, and 1%, respectively. Furthermore, the assessment can be carried out by measuring the performance of the suggested approach with that of another existing method, with the new way often outperforming the other. This suggested strategy may be used to additional datasets in the future. This is to assess its capacity to operate with varied data properties. Additionally, additional data reduction may be performed in order to have simpler data. Its purpose is to minimise running time and complexity.

5. REFERENCES

[1] Obaid, A. J., Ibrahim, K. K., Abdulbaqi, A. S., & Nejrs, S. M. (2021). An adaptive approach for internet phishing detection based on log data. *Periodicals of Engineering and Natural Sciences*, *9*(4), 622-631.

[2] Hosseini, S., & Sardo, S. R. (2021). Data mining tools-a case study for network intrusion detection. *Multimedia Tools and Applications*, *80*, 4999-5019.

[3] Kurniawan, Y. I., Razi, F., Nofiyati, N., Wijayanto, B., & Hidayat, M. L. (2021). Naive Bayes modification for intrusion detection system classification with zero probability. *Bulletin of Electrical Engineering and Informatics*, *10*(5), 2751-2758.

[4] Awad, N. A. (2021). Enhancing network intrusion detection model using machine learning algorithms. *CMC-Comput. Mater. Contin*, *67*, 979-990.

[5] Lou, P., Lu, G., Jiang, X., Xiao, Z., Hu, J., & Yan, J. (2021). Cyber intrusion detection through association rule mining on multi-source logs. *Applied Intelligence*, *51*, 4043-4057.

[6] Balla, A., Habaebi, M. H., Islam, M. R., & Mubarak, S. (2022). Applications of deep learning algorithms for Supervisory Control and Data Acquisition intrusion detection system. *Cleaner Engineering and Technology*, 100532.

[7] Kumar, R., Kumar, P., Tripathi, R., Gupta, G. P., Garg, S., & Hassan, M. M. (2022). A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. *Journal of Parallel and Distributed Computing*, *164*, 55-68.

[8] Lin, Y. D., Liu, Z. Q., Hwang, R. H., Nguyen, V. L., Lin, P. C., & Lai, Y. C. (2022). Machine learning with variational AutoEncoder for imbalanced datasets in intrusion detection. *IEEE Access*, *10*, 15247-15260.

[9] Muhammad, M. U. U. A. H., & Saleem, A. M. S. F. M. (2022). Intelligent Intrusion Detection System for Apache

Web Server Empowered with Machine Learning Approaches. *International Journal of Computational and Innovative Sciences*, *1*(1), 1-8.

[10] Qiu, W., Ma, Y., Chen, X., Yu, H., & Chen, L. (2022). Hybrid intrusion detection system based on Dempster-Shafer evidence theory. *Computers & Security*, *117*, 102709.

[11] Gupta, S. K., Tripathi, M., & Grover, J. (2022). Hybrid optimization and deep learning based intrusion detection system. *Computers and Electrical Engineering*, *100*, 107876.

[12] Xia, W., Neware, R., Kumar, S. D., Karras, D. A., & Rizwan, A. (2022). An optimization technique for intrusion detection of industrial control network vulnerabilities based on BP neural network. *International Journal of System Assurance Engineering and Management*, *13*(Suppl 1), 576-582.

[13] Hammad, M., Hewahi, N., & Elmedany, W. (2022). MMM-RF: A novel high accuracy multinomial mixture model for network intrusion detection systems. *Computers & Security*, *120*, 102777.

[14] Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, *55*(1), 453-563.

[15] Zebin, T., Rezvy, S., & Luo, Y. (2022). An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, *17*, 2339-2349.

[16] Selva, D., Nagaraj, B., Pelusi, D., Arunkumar, R., & Nair, A. (2021). Intelligent network intrusion prevention feature collection and classification algorithms. *Algorithms*, *14*(8), 224.

[17] Guezzaz, A., Benkirane, S., Azrour, M., & Khurram, S. (2021). A reliable network intrusion detection approach using decision tree with enhanced data quality. *Security and Communication Networks*, *2021*, 1-8.

[18] Ibrahim, K. K., & Obaid, A. J. (2021). Fraud usage detection in internet users based on log data. *International Journal of Nonlinear Analysis and Applications*, *12*(2), 2179-2188.

[19] Ibrahim, K. K., & Obaid, A. J. (2021). Fraud usage detection in internet users based on log data. *International Journal of Nonlinear Analysis and Applications*, *12*(2), 2179-2188.

[20] Seth, S., Singh, G., & Kaur Chahal, K. (2021). A novel time efficient learning-based approach for smart intrusion detection system. *Journal of Big Data*, *8*(1), 1-28.