INTRUSION DETECTION SYSTEM IN WEB MINING USING CONTINUOUS LEARNING VECTOR QUANTIZATION

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

Article History: Received: 23.02.2023 **Revised:** 08.04.2023 **Accepted:** 07.06.2023

Abstract

Cyber-attack detection by an intrusion detection system (IDS) in web mining involves utilizing the IDS to identify and mitigate malicious activities and threats specifically targeting web mining operations. The IDS monitors the network traffic associated with web mining operations. It analyses the data packets exchanged between the web mining server and clients, looking for any suspicious or anomalous patterns. This IDS combines the principles of intrusion detection with the analysis of web data to identify potential threats or anomalies. In this article, the IDS is implemented through Continuous Learning Vector Quantization (CLVQ) algorithm for identifying and classifying the intrusions presented in web based systems. Also, the Association Rule Mining (ARM) for similarity determination is introduced. Based on the experimental results, such pre-processing and combination of similarity determination method are applied with CLVQ classification algorithm in machine learning models using KDD' 99 dataset. The proposed CLVQ classification achieves higher classification accuracy.

Keywords: Cyber-attack, Intrusion Detection System, Web Mining, CLVQ, ARM.

[1]Research Scholar, Dept. of Computer Science Engineering,
Don Bosco Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University. [2]Professor, Dept. of Computer Science Engineering, Don Bosco Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University.

Email: [1]bbbsag@gmail.com, [2]ashokdbit2017@gmail.com

DOI: 10.31838/ecb/2023.12.s3.474

1. INTRODUCTION

An IDS in the context of web mining refers to a system or set of techniques used to identify and respond to unauthorized activities or malicious behavior within a web mining environment. Data from the web, including web pages, social media, and web logs may be mined for valuable information via web mining, for various purposes including data analysis, knowledge discovery, or recommendation systems [1]. The IDS in web mining aims to detect anomalies, patterns, or specific types of attacks within the web mining process. It helps ensure the reliability, availability, and privacy of the web mining system and the data being processed [2]. IDS can analyze logs generated by web servers, web crawlers, or web mining tools to identify abnormal patterns or suspicious activities. For example, it can detect an unusually high number of requests from a single IP address, frequent access to sensitive areas, or unexpected changes in web server behaviour [3].

In association rule mining (ARM), common co-occurring itemsets are discovered by mining transactional datasets for associations [4]. ARM entails two significant steps, namely frequent item recognition and generation of association rules. Analyze and interpret the discovered association rules to gain insights into the data. Apply the rules to real-world scenarios, such as market basket analysis, recommendation systems, or decision-making processes [5]. Evaluate the generated rules based on various criteria, such as support, confidence, lift, or interestingness measures. Filter and select the rules that meet the desired criteria [6].

IDS can monitor the behavior of web mining components, such as crawlers, data extraction modules, or data processing pipelines [7]. It looks for deviations from normal behavior that may indicate an intrusion or compromise. For instance, sudden changes in the crawling pattern, unexpected resource consumption, or unauthorized access to data repositories [8]. IDSs are also powerful and effective network security tools that allow unauthorized and abnormal network traffic flow to be detected [9]. Network-based Intrusion Detection Systems (NIDS) are security systems that monitor and analyze network traffic to identify and respond to potential intrusions or malicious activities [10].

Socially aware web page recommendation refers to a recommendation system that takes into account social factors and user preferences to provide personalized web page recommendations [11]. It leverages social information, such as user social networks, social interactions, and social context, to enhance the accuracy and relevance of recommendations. Learning Vector Quantization (LVQ) is a machine learning algorithm that falls under the category of supervised learning and is commonly used for classification tasks. LVQ is a variant of the more general Self-Organizing Map (SOM) algorithm and it is part of the artificial neural network family [12].

RELATED WORKS

Fuzzy linguistic summarization is presented in this work [13] for extracting rules that are fuzzy in the format of rules that relate emotional demands into the form of product design, components of the two different backward and forward models. In order to efficiently obtain language summaries based on sufficient evidence, brute force techniques and genetic algorithms are utilized. The method that uses time sequence characteristics and association rules aspects of this research provides a deep-learning-based classification of malware strategy. The enhanced LSTM is used in this approach [14] to extract time sequence information from diverse protocol data.

It is critical to use realistic data sets of typical and harmful networking

accomplishments in order to test machine learning methods since they need to reflect real-world network situations and it depends on real-time network activity [15]. This research [16] presents a unified assessment of current security approaches, including their advantages and disadvantages. It covers security problems in cloud service architecture, the necessity of dimensionality reduction, and intrusion detection systems (IDS).

The purpose of this research [17] was to determine if sophisticated web metrics collected from Google Analytics software might be utilised to assess the general interface of e-commerce sites and to detect possible usability concerns. By analysing structured data, the apriori technique is utilised to mine the correlations among "locations" and "categories" further. A network diagram is used to visualise the association findings [18]. The goal of this paper [19] is to build a classification system depends on principal component analysis is used to detect cyber-attacks in wireless adhoc networks. ANN-based intrusion detection systems will be compared with its performance.

In this study [20], a data mining methodology is suggested For the purpose of improving the accurateness of prediction and discovering rules for normal item sets. The K-means clustering approach was also used in order to minimize the dataset extent in order to increase the efficiency of the suggested model. For document clustering, this framework [21] is supplied in an interpretable format. It iteratively decreases the mistake rate during data relationship mapping with the use of weighted document content. This research [22] focuses on the use of Federated Learning methods in Intrusion Detection. Both technologies are thoroughly discussed, and recent scientific progress is examined and classified. Finally, the study discusses the limits of current efforts and possible future options for this technology.

2. PROPOSED METHODOLOGY

In this proposed research work, Kalman filter is used for preprocessing method and Association Rule Mining technique used for similarity determination. A network traffic associated with web mining operations are detected and classified by CLVQ algorithm. Preprocessing, similarity determination, and Classification are the three components that make up the system architecture as shown in Figure 1.

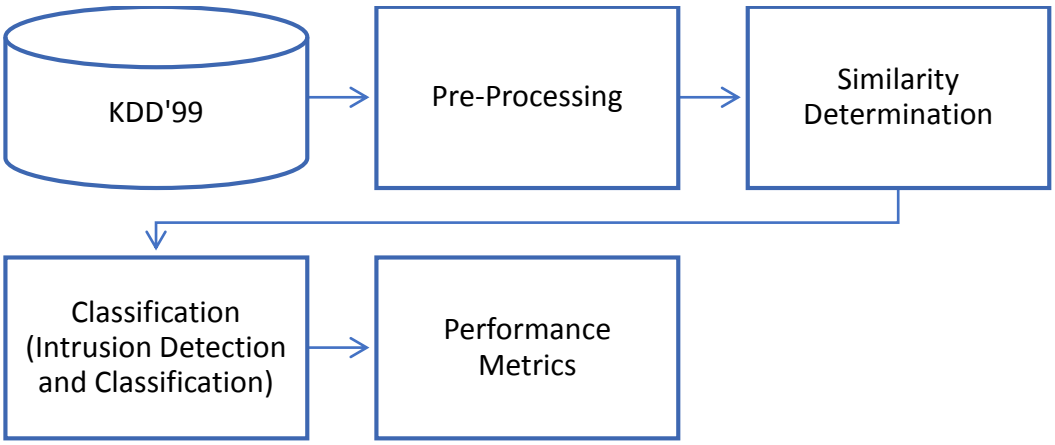

Figure 1. Overall architecture of Proposed System

3.1 Preprocessing

Based on noisy observations, the Kalman filter estimates the state of a system mathematically. While it is not directly used for preprocessing in web mining or IDS (Intrusion Detection System) tasks, it

can be employed as a component within a larger system. Steps involved in Kalman filter of an IDS web mining system:

Data Collection: Gather the network traffic data, such as network packets, HTTP requests, or log files, as the input for the IDS web mining system.

Feature Extraction: Extract relevant features from the collected data. These features can include source/destination IP addresses, packet sizes, timestamps, HTTP request methods, URLs, etc. The specific features depend on the nature of the IDS and the web mining task.

Noise Filtering with Kalman Filter: Apply the Kalman filter to filter out noise and enhance the quality of the extracted features. The Kalman filter can help smooth out noisy measurements, remove outliers, and estimate the true state of the observed features.

Kalman Filter Preprocessing Steps: a. Initialize Kalman filter parameters: Set up the initial state estimation, covariance matrix, transition matrix, and measurement matrix for the Kalman filter. b. Iterate through the feature sequence: For each extracted feature, perform the following steps: (i) Predict step: Use the transition matrix to predict the next state and update the covariance matrix. (ii) Update step: Compare the predicted state with the actual observed feature value. Use the measurement matrix to update the state estimation and covariance matrix based on the measurement residual. c. Obtain filtered features: After processing all the features with the Kalman filter, the resulting estimates from the filter represent the filtered and enhanced features.

Preprocessing Outcome: The filtered features obtained from the Kalman filter can be used as input for subsequent steps in the IDS web mining system, such as anomaly detection, classification, or pattern recognition algorithms. The noise reduction and state estimation performed by the Kalman filter can help improve the accuracy and effectiveness of these subsequent steps.

3.2 Association Rule Mining for Similarity Determination

ARM is a method to find the relation between the variables in large volumes of data. ARM can be applied to an IDS to enhance its capabilities in detecting and analyzing patterns of intrusions or malicious activities. ARM is a data mining technique that identifies relationships and patterns within datasets. The first step is to collect and preprocess the data for analysis. This includes gathering relevant security logs, network traffic data, or any other sources of information that capture the behavior of the system under surveillance. The data is then preprocessed to remove noise, handle missing values, and transform it into a suitable format for association rule mining.

From the preprocessed data, relevant features or attributes are extracted to represent the characteristics of the system and its activities. These features can include source and destination IP addresses, timestamps, protocol types, port numbers, or any other attributes that provide insights into the system's behavior. ARM is applied to the extracted features to discover patterns or relationships between different attributes. The IDS aims to identify associations between attributes that indicate potentially malicious activities or intrusions. For example, certain combinations of IP addresses, port numbers, and protocol types may frequently occur together during known attacks, forming association rules.

Rule Generation: The association rule mining process generates a set of rules that describe the relationships or associations found in the data. Each rule consists of an antecedent (a set of conditions) and a consequent (the outcome or conclusion). The rules are typically measured based on metrics such as support (frequency of the

rule), confidence (likelihood of the consequent given the antecedent), or lift (degree of association between the antecedent and consequent).

Rule Evaluation: The generated rules are evaluated based on predefined criteria or expert knowledge. Rules that meet certain thresholds or criteria are considered significant and potentially indicative of intrusions or malicious activities. The evaluation can include considering the domain knowledge, relevance to known attack patterns, or statistical measures of the rule quality.

Rule Application and Detection: The evaluated rules are applied to incoming data in real-time to detect potential intrusions or malicious activities. The IDS monitors the system's behavior, and when it identifies patterns that match the antecedents of the discovered rules, it generates alerts or triggers appropriate response actions. These actions can include blocking IP addresses, terminating sessions, or raising alarms for further investigation.

The concept of association rule mining refers to the process by which a dataset is mined in order to reveal interesting relationships or associations between items. The mathematical model for association rule mining involves several key components and measures. Let's explore them:

Itemsets: An itemset is a collection of items that co-occur together in a transaction or dataset. In association rule mining, itemsets are represented as sets of items. A mathematical notation to represent an itemset is $\{item1, item2, \ldots, itemn\}$.

Support: Support is a measure that quantifies the regularity or occurrence of an itemset in the dataset. It signifies the proportion of communications that contain the itemset. Mathematically, support is calculated as:

$$\text{Support}(A) = (\text{Number of transactions containing itemset A}) / (\text{Total number of transactions})$$

Support is used to filter out infrequent itemsets and focus on those that occur frequently.

Confidence: Confidence measures the strength of the association between two itemsets. It denotes the restrictive probability of finding the consequent itemset given the antecedent itemset. Mathematically, confidence is calculated as:

$$\text{Confidence}(A \rightarrow B) = (\text{Support}(A \cup B)) / (\text{Support}(A))$$

Confidence is used to identify strong associations between itemsets.

Lift: Lift is a measure that indicates the significance of an association rule by comparing the observed support with the expected support if the items were independent. Mathematically, lift is calculated as:

$$\text{Lift}(A \rightarrow B) = (\text{Support}(A \cup B)) / (\text{Support}(A) \times \text{Support}(B))$$

Lifts greater than 1 indicate positive associations, lifts equal to 1 indicate independence, and lifts less than 1 indicate negative associations.

Minimum Support and Minimum Confidence: These are threshold values set by the user to filter out uninteresting or weak associations. Itemsets and rules that do not meet the minimum support and minimum confidence thresholds are typically discarded.

The mathematical model for association rule mining involves using these measures, thresholds, and algorithms to discover frequent itemsets and generate association rules that exhibit interesting relationships among the items in the dataset. The goal is to identify associations that have high

support, confidence, and lift, indicating strong relationships and potential insights.

3.3 Continuous Learning Vector Quantization algorithm

Continuous Learning Vector Quantization (CLVQ) algorithm that enables incremental learning and adaptation in response to new data. CLVQ is a supervised learning algorithm that depends to the family of competitive learning algorithms. It is primarily used for pattern recognition and classification tasks. Steps invlovled in CLVQ are:

Initialization: CLVQ starts by initializing a set of codebook vectors or prototypes. These codebook vectors represent the classes or categories in the training data. Each codebook vector is a multidimensional vector in the feature space.

Training Phase: During the training phase, the algorithm processes the examples at a time. In each case of training, CLVQ computes the similarity or distance between the input vector and the codebook vectors. The similarity is typically measured using a distance metric such as Euclidean distance.

Winner Selection: CLVQ identifies the codebook vector that is closest or most parallel to the input vector. This codebook vector is referred to as the winner or best matching unit (BMU). The distance among the input vector and the BMU determines the degree of similarity.

A learning algorithm, denoted as L (θ, X, Y), updates the model's parameters θ based on the input data X and corresponding labels Y.

Adaptation: After selecting the BMU, CLVQ adjusts the BMU's position in the feature space to become more similar to the input vector. This adaptation process involves moving the BMU closer to the input vector in the feature space. By updating the BMU, CLVQ learns to better represent the input vectors from the same class.

Given the current set of parameters θ(x), the learning algorithm updates the parameters based on new data (X, Y) as follows:

$$\theta(x+1) = L(\theta(x), X, Y)$$

Neighborhood Adaptation: In addition to updating the BMU, CLVQ also adjusts the neighboring codebook vectors in the feature space. This neighborhood adaptation helps in preserving the topological relationships among the codebook vectors. The neighboring codebook vectors are moved closer to the input vector, but to a lesser extent compared to the BMU.

Iteration: Steps 3-5 are repeated for a fixed number of iterations or until convergence is achieved. Each iteration updates the codebook vectors based on the training examples, gradually refining their positions in the feature space.

The main advantage of CLVQ over traditional LVQ is its ability to adapt incrementally to new data. As new training examples become available, CLVQ can update the codebook vectors without retraining the entire model from scratch. This enables CLVQ to handle dynamic datasets where the distribution of classes or patterns may change over time.

Pseudocode for CLVQ

```
InitializeParameters(parameters) // Initialize model parameters
For each training iteration: // Training phase
    newData = GetNewTrainingData()// Get new labeled training data
   UpdateModel(parameters, newData) // Update model with new data
    trainingPerformance = EvaluateModel(parameters, newData)
      If convergenceCriteriaMet(trainingPerformance):
```

```
        Break
While more data is available or feedback is provided:
   additionalData = GetAdditionalData()
   feedback = GetFeedback()
   UpdateModel(parameters, additionalData, feedback)
   adaptationPerformance = EvaluateModel(parameters, additionalData)
   If convergenceCriteriaMet(adaptationPerformance):
      Break
testingData = GetTestingData() // Testing phase
testingPerformance = EvaluateModel(parameters, testingData)
FinalParameters = parameters
```

In this pseudocode, the IDS is continuously learning and adapting its model based on new labeled training data and any feedback provided. The training phase involves iteratively updating the model with new training data and evaluating its performance on the training set. The process continues until a convergence criterion is met. The adaptation phase allows the IDS to incorporate additional labeled data or feedback to further improve its model. The model is updated with the additional data/feedback, and its performance is evaluated on the adaptation data. The adaptation process also stops when a convergence criterion is met. Finally, the testing phase evaluates the performance of the final model on separate testing data to assess its effectiveness in real-world scenarios.

4. RESULTS AND DISCUSSIONS

4.1 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 depends 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 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.

4.2 Performance analysis

When evaluating the performance of a continuous learning and adaptation system for classifying IDS (Intrusion Detection System), several performance measures can be used to assess its effectiveness. There are:

Accuracy: Accuracy processes the overall perfection of the classification results by comparing the number of suitably classified illustrations to the total number of instances as shown in fig 2. It is computed as:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

Where, TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

Precision: In terms of precision, it represents the percentage of correctly identified positive incidents (intrusions) out of all positive incidents as shown in fig 3. It is computed as:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Precision is useful in situations where the focus is on minimizing false positives.

Recall (Sensitivity or True Positive Rate): It is the percentage of correctly identified positive incidents (intrusions) out of all the instances that are in fact positive as shown in fig 4. It is computed as:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Recall is important when the emphasis is on minimizing false negatives.

F1-Score: In order to combine precision and recall into a single measure, the F1-score is calculated in a way that is balanced between both metrics as shown in fig 5. This is a measure of precision and recall calculated as follows:

$$\text{F1} - \text{Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

The F1-score provides a balanced measure of the classifier's performance on both precision and recall.

Specificity: Specificity measures the proportion of correctly identified negative instances (non-intrusions) out of all actual negative instances as shown in fig 6. It is computed as:

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

Specificity is relevant when the focus is on minimizing false positives.

These performance measures provide different perspectives on the enactment of the continuous learning and adaptation system for IDS classification as given in table 1. It's important to select the appropriate performance measures based on the specific requirements and objectives of the IDS system and consider the trade-offs between different metrics depending on the application context.

Table 1: Comparison of Proposed model with existing model

Classifier	Results (%)				
	Accuracy	Precsion	Recall	F1-Score	Specificity
k-NN	98.62	93.43	94.69	93.8	89.91
SVM	99.96	95.60	96.13	95.7	90.20
Naïve Bayes	92.26	94.73	95.14	94.3	89.90
Proposed	99.98	96.35	96.45	96.9	91.3

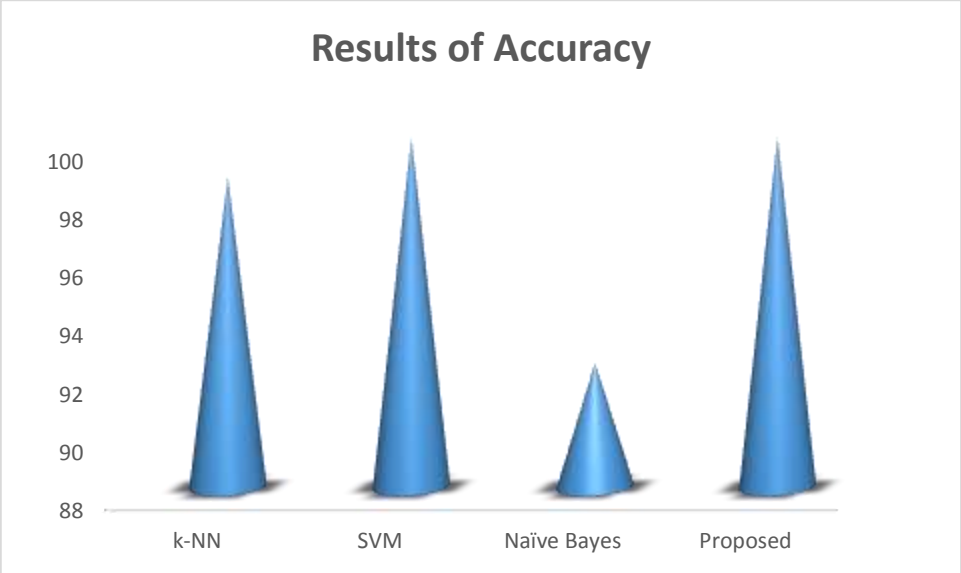

Figure 2. Accuracy Comparison

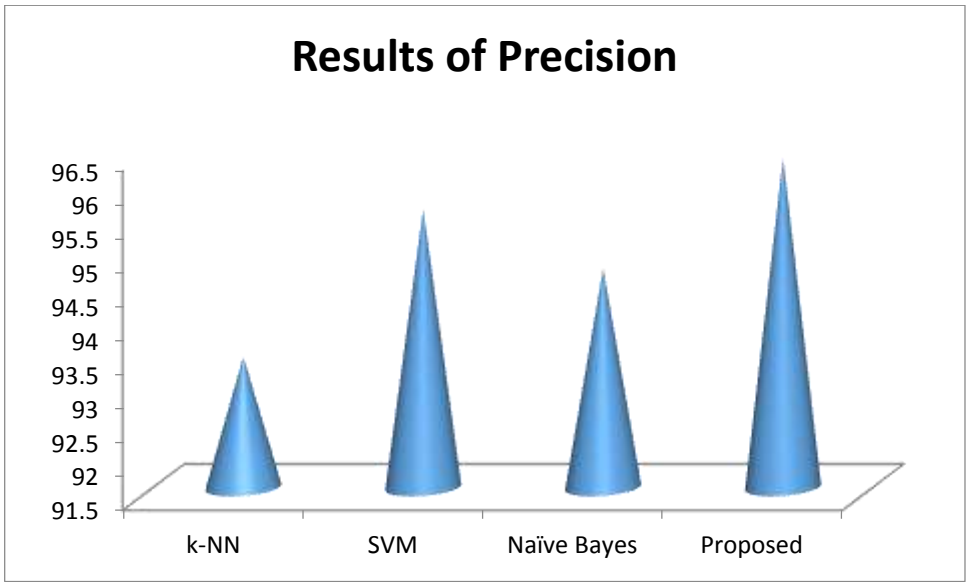

Figure 3. Precision Comparison

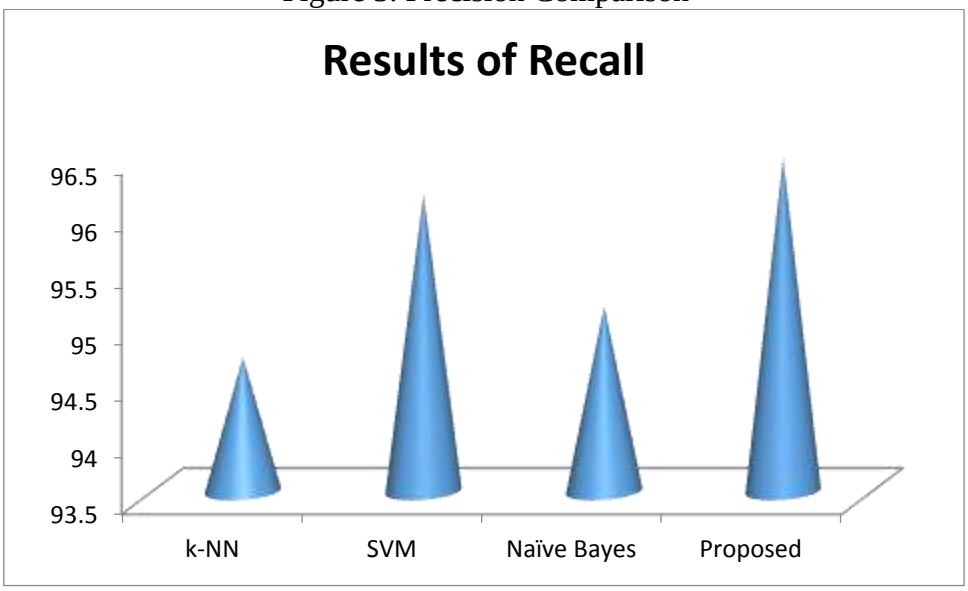

Figure 4. Comparison of Recall

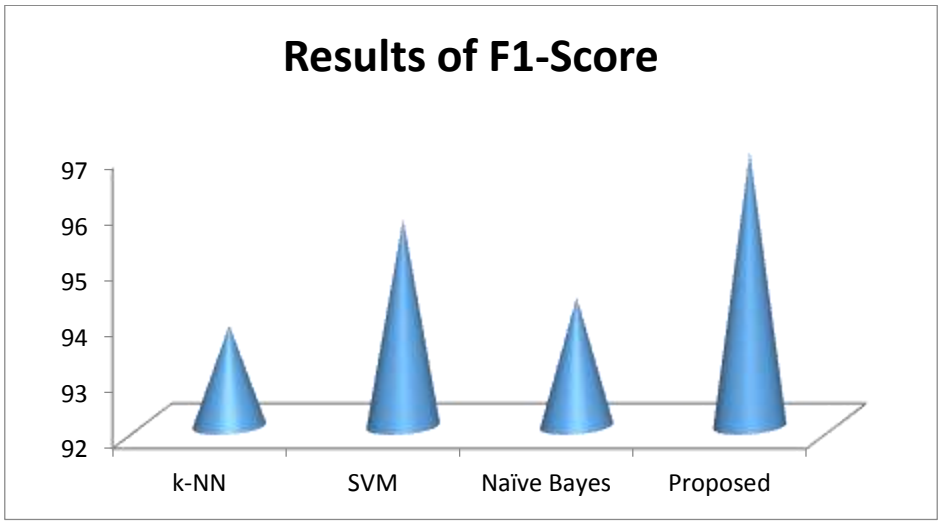

Figure 5. Results of F1-Score

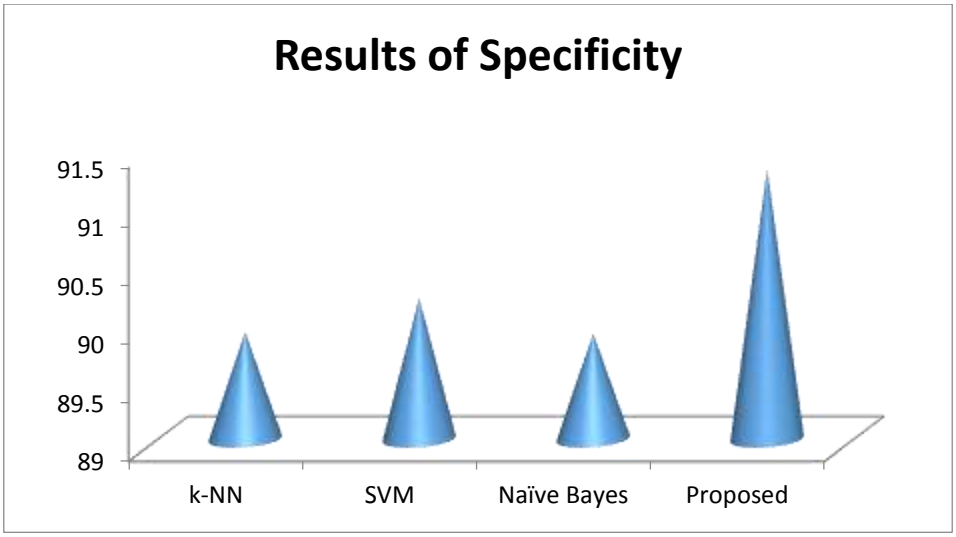

Figure 6. Results of Specificity

From the above figure, clearly shows that proposed classifier gives better accuracy for Intrusion Detection in a web mining than other classifiers.

5. CONCLUSION

The cyber-attacks detection becomes more essential in Web systems. IDS aims to identify and respond to malicious activities or unauthorized access attempts. To effectively protect an organization from attacks, IDS must be able to handle large amounts of data without impacting network performance or dropping important data. In this research, propsoed a kalman filtering as preporcessing method and ARM for similarity determination, which were then evaluated on distinct data set of KDD'99 using CLVQ classifying algorithm for detecting intrusion in a web mining. It has been determined that the proposed method is 99.98% accurate based on the evaluation results. The results of this study are also evaluated using the KDD'99 dataset, which contains a variety of records. Several parameters were estimated in order to perform a comparative analysis such as accuracy, precision, recall, F1-score, specificity respectively. As a result of the performance analysis, the proposed algorithm has a much higher performance level than other existing algorithms, thus making it superior to them.

6. REFERENCES

Kumar, S., Gupta, S., & Arora, S. (2021). Research trends in network-based intrusion detection systems: A review. IEEE Access, 9, 157761-157779.

Chhatwal, G. S., & Deepak, G. (2022). IEESWPR: An Integrative Entity Enrichment Scheme for Socially Aware Web Page Recommendation. In Data Science and Security: Proceedings of IDSCS 2022 (pp. 239-249). Singapore: Springer Nature Singapore.

Hadi, R. M., Abdullah, S. H., & Abedi, W. M. S. (2022). Proposed neural intrusion detection system to detect denial of service attacks in MANETs. Periodicals of Engineering and Natural Sciences, 10(3), 70-78.

Sharmila, S., & Vijayarani, S. (2021). Association rule mining using fuzzy logic and whale optimization algorithm. Soft Computing, 25, 1431-1446.

Imran, Jamil, F., & Kim, D. (2021). An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. Sustainability, 13(18), 10057.

Tama, B. A., & Lim, S. (2021). Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. Computer Science Review, 39, 100357.

Torabi, M., Udzir, N. I., Abdullah, M. T., & Yaakob, R. (2021). A review on feature selection and ensemble techniques for intrusion detection system. International Journal of Advanced Computer Science and Applications, 12(5).

Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M., & Li, M. (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile networks and applications, 26, 808-829.

Sharmila, S., & Vijayarani, S. (2021). Association rule mining using fuzzy logic and whale optimization algorithm. Soft Computing, 25, 1431-1446.

Ghazal, T. M., Al-Dmour, N. A., Said, R. A., Omidvar, A., Khan, U. Y., Soomro, T. R., ... & Ali, L. (2023). DDoS Intrusion Detection with Ensemble Stream Mining for IoT Smart Sensing Devices. In The Effect of Information Technology on Business and Marketing Intelligence Systems (pp. 1987-2012). Cham: Springer International Publishing.

Liu, H., Zhong, C., Alnusair, A., & Islam, S. R. (2021). FAIXID: A framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. Journal of network and systems management, 29(4), 40.

Kayacik, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2007). A hierarchical SOM-based intrusion detection system. Engineering applications of artificial intelligence, 20(4), 439-451.

Akgül, E., Delice, Y., Aydoğan, E. K., & Boran, F. E. (2022). An application of fuzzy linguistic summarization and fuzzy association rule mining to Kansei Engineering: a case study on cradle design. Journal of Ambient Intelligence and Humanized Computing, 13(5), 2533-2563.

Niu, W., Zhou, J., Zhao, Y., Zhang, X., Peng, Y., & Huang, C. (2022). Uncovering APT malware traffic using deep learning combined with time sequence and association analysis. Computers & Security, 120, 102809.

Antunes, M., Oliveira, L., Seguro, A., Veríssimo, J., Salgado, R., & Murteira, T. (2022, March). Benchmarking Deep Learning Methods for Behaviour-Based Network Intrusion Detection. In Informatics (Vol. 9, No. 1, p. 29). MDPI.

Lata, S., & Singh, D. (2022). Intrusion detection system in cloud environment: Literature survey & future research directions. International Journal of Information Management Data Insights, 2(2), 100134.

Kumar, B., Roy, S., Sinha, A., Iwendi, C., & Strážovská, Ľ. (2022). E-Commerce Website Usability Analysis Using the Association Rule Mining and Machine Learning Algorithm. Mathematics, 11(1), 25.

Chen, S., Xi, J., Chen, Y., & Zhao, J. (2022). Association mining of near misses in hydropower engineering construction based on convolutional neural network text classification. Computational Intelligence and Neuroscience, 2022.

Kumar, K. S., & Nagalakshmi, T. J. (2022, November). Design of Intrusion Detection System for Wireless Ad Hoc Network in the Detection of Man In The Middle Attack using Principal Component Analysis Classifier Method Comparing with ANN classifier. In 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS) (pp. 1-6). IEEE.

Alghanam, O. A., Al-Khatib, S. N., & Hiari, M. O. (2022). Data mining model for predicting customer purchase behavior in e-commerce context. International Journal of Advanced Computer Science and Applications, 13(2).

Peruma, P. (2022). Document Clustering Using Graph Based Fuzzy Association Rule Generation. Comput. Syst. Sci. Eng., 43(1), 203-218.

Belenguer, A., Navaridas, J., & Pascual, J. A. (2022). A review of federated learning in intrusion detection systems for iot. arXiv preprint arXiv:2204.12443.