


Research Article


Deep Packet Inspection Model Based on Support Vector Machine for Anomaly Detection in Local Area Networks

¹Margaret Moronke DOSUNMU 

Department of Educational Management and Business Studies
Olabisi Onabanjo University
Ago-Iwoye, Nigeria
dosunmu.margaret@oouagoiwoye.edu.ng

²Femi Emmanuel AYO 


Department of Mathematical Sciences
Olabisi Onabanjo University
Ago-Iwoye, Nigeria
ayo.femi@oouagoiwoye.edu.ng

³Lukman Adebayo OGUNDELE 

Department of Mathematical Sciences
Olabisi Onabanjo University
Ago-Iwoye, Nigeria
ogundele.lukman@oouagoiwoye.edu.ng

⁴Abass Ishola TAIWO 

Department of Mathematical Sciences
Olabisi Onabanjo University
Ago-Iwoye, Nigeria
taiwo.abass@oouagoiwoye.edu.ng

⁵Timothy Olabisi OLATAYO 

Department of Mathematical Sciences
Olabisi Onabanjo University
Ago-Iwoye, Nigeria
bisi.olatayo@oouagoiwoye.edu.ng

ARTICLE INFO

Article History

Received: 28/05/2024

Accepted: 10/07/2024

Published: 24/09/2024

This is an open-access article under the CC BY 4.0 license:

<http://creativecommons.org/licenses/by/4.0/>

**ABSTRACT**

Deep packet inspection is a network security solution that identifies and flags anomalous network traffic patterns in a local network environment. Traditional signature-based techniques for intrusion detection are limited in identifying different attacks or completely new kinds, which makes them unsuitable in some situations. In addition, most previous methods for anomaly detection have low detection rate and high false alarm. In this study, a deep packet inspection model based on support vector machine (SVM) for anomaly detection in local area networks was proposed. The proposed method combined the SelectKBest method and SVM for the categorization of anomaly in a local network environment. Results showed that the proposed method outperformed other related machine learning methods with accuracy, precision, recall, and F1-score of 94.81%, 94.03%, 94.13%, and 94.0799%, respectively. The accuracy result shows that most network traffic can be correctly identified by the SVM using the SelectKBest approach, with minimal false positives or negatives.

Keywords: Deep packet inspection; Anomaly detection; Local area network; Support vector machine; SelectKBest

1. INTRODUCTION

Deep packet inspection (DPI) is a network security solution that identifies and flags anomalous or suspicious network traffic patterns in a local network environment for traffic anomaly identification in local area networks (LANs) [1, 2]. This system typically operates at the network level and monitors every data packet that travels over the LAN. A DPI system is a sophisticated network security method that provides an in-depth degree of network traffic scanning. Real-time packet content inspection enables the detection of network anomalies, breaches, and assaults that may be missed

by traditional security solutions. Understanding the patterns and behaviors of LAN traffic can help organizations strengthen their network security against cyber threats including malware, data exfiltration, and unauthorized access [3, 4].

The exponential growth of internets and networked systems has increased the amount of data transferred via LANs. With increased LAN traffic, maintaining and securing these networks become gradually challenging [5, 6]. Firewalls and intrusion detection systems (IDSs) are two examples of conventional security measures that are essential but might be insufficient to fend off complex and modern cyberattacks [7]. Comprehending the peculiarities of network traffic is a crucial undertaking. Analyzing the abnormalities is vital regardless of their harmful nature for two reasons: detecting anomalies is important because they can increase router resource use and create network congestion. Although certain abnormalities do not directly disrupt the network, they can still negatively affect clients or end users.

The identification and categorization of anomalies necessitates an ongoing process of event monitoring in computer systems and networks, which calls for the analysis of large amounts of data produced by many sources [8, 9]. An effective way to identify network anomalies and attacks is to use a DPI that combines the features of an intrusion prevention system with an IDS. DPI-based solutions support the creation or optimization of service offerings, enhance service quality, manage service policies, and safeguard the network and its users. They do so by providing a comprehensive picture of network usage, identifying subscribers who consume large amounts of traffic, and efficiently managing traffic in real time. DPI systems are usually implemented in the busiest parts of the network (trunk links and links to other networks) and where service control is necessary. DPI uses criteria specified by the administrator and decides which application type a given session belongs to by analyzing the packets going through it, including the data field. By examining packet data, the administrator can prioritize, block, limit, or reroute traffic to other systems based on predefined criteria. DPI determines the protocols using heuristic and statistical technologies in addition to rules based on signature analysis. Therefore, the rules set by the administrator will be applied based on the traffic being transmitted, even if that traffic is transmitted on ports that are not standard for the application being used.

Current IDSs rely on signature-based analytic techniques in recognizing and precisely defining known types of assaults [10, 11]. However, these algorithms are limited in classifying different attacks or completely new kinds, which makes them unsuitable in some situations. No universal mechanism can detect attacks that have not been discovered before despite the existence of solutions for detecting network anomalies [12, 13]. Real-time network traffic monitoring using commercial DPI software tools is used to spot anomalies and possible new threats. DPI methods are quite good at finding well-known malware signatures. DPI is also capable of analyzing attack patterns, which include the sequence in which attacks are launched, the path taken by the attacker, and the methods used in the network flow. This study proposes a DPI model based on support vector machine (SVM) to discover anomalies in LANs. A hybrid feature selection technique will be incorporated into the suggested DPI model based on SVM to improve the accuracy of the base classifier.

The rest of the paper is designed as follows. Section 2 presents the related works. The materials and method are presented in Section 3. The results and discussion are presented in Section 4. Section 5 elaborates the conclusions and the future work.

2. RELATED WORK

Anomaly detection methods can be broadly classified into two major groups based on the environment and detection mechanism.

2.1 Nature of Environment

- Host-based IDS: It is responsible for spotting and recognizing harmful activities on the host computer. One advantage of host-based IDS over network IDS is its capacity to identify malicious traffic coming from the host itself, as well as suspicious attacks that are generated inside the company or harmful activities that go undetected by NIDS [14].
- Network IDS: It is a device that watches the entire network from its installation location and is equipped with an enhanced detection mechanism to identify potentially malicious users [15].

2.2 Detection Mechanisms

- Anomaly-based IDS: It is also known as behavior-based IDS. This robust technique uses behavioral deviations from a typical profile to identify unusual, hostile acts, and new attacks [16].

- Signature-based IDS: also known as the misuse-based IDS or knowledge-based IDS. These systems examine previous attacks to extract the discriminating patterns and traits known as signatures, along with signature patterns.

2.3 Review on DPI Methods

Liu et al. [17] propose a deep architecture for anomaly detection in packet payloads. Preprocessing, feature construction, model construction, and anomaly detection comprise the four modules of the suggested architecture. The payload is retrieved and labeled in the first module. The feature construction module uses a sliding block to create a block sequence. A dictionary is then used to choose the high-frequency elements in the block sequence, which eliminates unnecessary information. By encoding each item in the block sequence into an embedded vector, the block-based features are created throughout the block embedding process. The next two modules comprise the final section of the suggested architecture, which attempts to detect anomalies for packet payload in an adaptive manner. In particular, a multilayer perceptron is used as a classifier to identify anomalies in each sample, and a neural network based on long short-term memory (LSTM) and a convolutional neural network (CNN) is designed to learn the long- and short-term dependency relationships in the block-based features.

Summerville et al. [18] offer an extremely lightweight deep packet anomaly detection system for Internet of Things (IoT) devices. The method allows the n-gram size to vary by dimension and models payloads in an efficient and flexible manner using n-gram bit-patterns. The detector can quickly determine packet classification by employing a direct representation of the feature space for the discrimination function. The method exhibits substantial parallelism that makes it suitable for effective implementation in software and hardware. The detectors have demonstrated exceptional efficacy in detecting anomalous packets from various sorts of attacks and in differentiating between device-specific and general Internet traffic.

Spiekermann and Keller [19] suggest using packets to detect anomalies in virtual networks without supervision. The suggested unsupervised packet-based anomaly detection for network analysis is broken down into six parts: formulation of the problem, data collection and analysis, building the model, validating the model, deployment, and inference. During the phase of problem formulation, the researchers establish several parameters for the analysis. Given that machine learning (ML) algorithms can be time consuming, providing detailed definitions of the ML categories and specifying the required input data for subsequent steps are important. During the period of data gathering, the pertinent packets are obtained. This stage is simple to implement in traditional networks, but the complexity of network packet capture procedures increases in virtual networks. The data analysis phase includes all the actions required to convert the recorded packets into a format that can be used for the next phases. The learning model is trained, tested, and fine-tuned as part of the model creation process. The model is checked for quality by validation. The deployment and inference phase includes performance, accuracy, and resource management.

Basumallik et al. [20] develop a CNN-based state estimator based on phasor measurement units (PMUs) for packet-data anomaly detection. This work addresses a family of fake data injection attacks (FDIAs) that attempt to alter PMU measurements for producing inaccurate answers for state estimation (SE). Prior to each cycle of SE, the authors extract multi-variate time-series signals from PMU data packets pooled in phasor data concentrators corresponding to various occurrences such as line faults and trips, generation and load variations, shunt disconnections, and FDIA. A CNN data filter with Nesterov Adam gradient descent and category cross entropy loss is suggested to validate the PMU data. The performance of the filter is contrasted with that of (a) classic classifiers like SVM and ensemble techniques, as well as (b) deep learning algorithms like recurrent neural networks and LSTM. Among all classifiers, the suggested CNN-based filter produces superior classification accuracies.

Liu and Wang [21] establish a CNN-based real-time anomaly detection system for network traffic. The system conducts online real-time packet extraction and identification by directly extracting the original features of the network flow through the use of a CNN. It utilizes software-defined networking to enable a zero-configuration anomaly detection system by allowing it to react flexibly to network changes. The experimental results fully demonstrated that the proposed method can detect abnormal traffic problems and enhance the security performance of edge clustering networks.

Lara et al. [22] offer a modular architecture made of building blocks. This architecture is appropriate for identifying irregularities in network traffic and application-layer data that are shared across IoT devices in a smart home setting.

In keeping with this architecture, the writers specify a specific IDS for a case study that makes use of a publicly available dataset that tracks the electricity usage of 21 household appliances over the course of a year. Specifically, the authors establish two anomaly detectors to identify bogus command or data injection attacks and 10 indicators of compromise to identify network attacks. In addition, the authors expand the detection range to cover known attacks with a signature-based IDS (Snort). To evaluate our the effectiveness of IDS, the authors replicate four false command or data injection attacks and eight network assaults, such as DoS and scanning. The authors further mention that, given that they worked with a publicly available dataset, their contribution can be utilized as a benchmark when comparing it to fresh methods that enhance detection performance.

Song et al. [23] propose a system for network traffic analysis and anomaly detection using software-based DPI. They address the issue of identifying network anomalies by suggesting a technique for constructing a collection of useful features that formalize the normal and anomalous behavior of the system based on an assessment of the Hurst parameter of the network traffic. In addition, the criteria for utilizing the Hurst parameter and the three-sigma rule to identify and stop different kinds of network anomalies are established. The Hurst parameter is assessed by the authors using the rescaled range approach. Several parameters, including minimal computation time, minimal monitoring time, self-training potential, and the ability to observe various traffic types, are necessary for the practical utility of the suggested method. The authors devise an algorithm for protocol detection and analysis of collected data to determine statistical load parameters for the new DPI. Furthermore, flow control algorithms are built to guarantee the quality of service. These algorithms are developed by utilizing the results of a static analysis of flows and a suggested way of anomaly identification using the Hurst parameter. The authors evaluate the potential for network traffic anomaly identification and prevention between the suggested software-based DPI system and the current SolarWinds DPI system. The results show that the former is better than the latter.

Deri et al. [24] suggest a high-speed, DPI (nDPI) tool that is open-source. The current version is deficient in a number of significant areas. The quality of the findings produced is the main flaw in the current implementation (as well as in most of the traffic classification systems in use at present). Typically, classifiers yield a single result per flow, which is meant to provide the most complete flow characterization. As a result, the output is a combination of results on different levels: content types (such as MPEG or Flash), application protocols (such as DNS, HTTP, SSL, BitTorrent, or SMTPS), IP protocols (such as TCP or UDP), and service providers (such as Facebook or YouTube). Such a result is incredibly limited in utility. As a result, the authors recommend updating the results format to offer all possible classes in a consistent manner.

Doroud et al. [25] develop a workable method for enhancing the effectiveness of conventional traffic classification techniques by connecting quick classification stages (ML and port-based). These combined stages help reduce false-positive rates. They also introduce a more exact stage based on DPI, which requires additional time and resources. According to experimental data, chain achieves precision, recall, accuracy, and area under the curve (AUC) scores that are consistent with DPI techniques, and it is 45% faster than nDPIng, which is a well-known DPI implementation. The viability of the suggested method in network function virtualization scenarios is verified. The state-of-the-art DPI classifier, nDPIng, is compared with the authors' chain classifier, which increases classification efficiency over it. Chain classifier accurately classifies network traffic 45% faster than nDPIng while maintaining a comparable classification performance. Chain can be used as an approach to conduct online categorization more quickly and with less processing resources because DPI is frequently employed in application firewalls. In addition to being more privacy-respecting than pure DPI, chain significantly lessens the effects of circumvention techniques that have gradually made port-based classification unreliable over time. Furthermore, chain's multi-stage design is particularly well suited for deployment as virtual network function chaining. In this approach, stages with varying memory and computational costs can be separately scaled and deployed based on the dynamic mix and rate of network traffic. Future works will present this implementation, which is presently under development.

2.4 ML Algorithms on DPI System

- SVM
SVM represents each item to be classified in an n-dimensional space, and the coordinates of these points are referred to as features in some cases. First, SVM draws a hyperplane on which points belonging to one group are placed on one side, and points belonging to the other group on the other [26].
- Feature Selection

One of the problems encountered by ML algorithms is high dimensionality. These days, it's typical to have datasets with a large number of fields or columns [27, 28, 29]. Large datasets are typically employed in IDSs. Employing ML techniques to train the system on datasets with many features can be time consuming, increase learning complexity, and negatively affect the system if irrelevant characteristics are included. Thus, feature extraction or selection aids in eliminating extraneous features or noise from the data, which increases the classification rate.

3. MATERIALS AND METHOD

This study suggests a deep packet inspection approach for SVMs to discover anomalies in LANs. The preprocessing of the dataset, feature selection, normalization, and classification make up the suggested method. Preprocessing the dataset entails removing noise from the modified dataset and redistributing the class distribution in the dataset using random undersampling to prevent class imbalance. The features are chosen based on the k highest score using Python's SelectKBest method for feature selection. The values of the chosen features are normalized to a common scale using the normalization phase. SVM is utilized in the classification phase for anomaly detection in LANs based on the normalized selected characteristics. The architecture of the suggested DPI model based on SVM for anomaly detection in LANs is depicted in Figure 1.

3.1 Dataset Preprocessing

Data preprocessing utilizes NumPy and Pandas. NumPy can handle multidimensional arrays quickly. The data frame architecture of pandas eases the conversion of CSV files. It can manage big data collections. The open-source Scikit-Learn framework, which is compatible with NumPy and Pandas, is used for training models. It provides several different ML functions. Using the CICIDS-2018 dataset as an input, an ML approach is utilized to train the anomaly inference engine unit. This dataset is utilized because of its popularity for modern attack representation of wide area networks and LANs. It contains the infiltration of the network from inside attacks, which aligns with the scope of this study for anomaly detection within the LAN.

- **Cleaning:** The eight traffic data files have been combined, and rows containing the values “Infinity,” “Null,” or “NaN” have been removed. An overview of the label distribution of the dataset is provided in Table 1. Recurring columns are removed. Zero-valued columns are also eliminated.
- **Undersampling method:** Table 1 shows that the number of “BENIGN” records is significantly higher than the number of attack records. Consequently, random undersampling is used to shift the class distribution of the dataset to a proportion of 30% assaults and 70% BENIGN records for preventing model bias. Using the random undersampling strategy, instances from the majority class are chosen randomly and then removed from the training dataset. In addition, datasets with an uneven distribution of observations for the target class—that is, one class label having a very high number of observations while the other has a very low number of observations—are referred to as imbalanced data. Random undersampling technique is used to regularize the minority or majority class. Rows from the majority class can be randomly removed using this strategy to match the majority class with the minority class. A balanced dataset for majority and minority classes can be obtained after sampling the data. Given that both classes have an equivalent number of records in the dataset, the classifier will assign equal weights to each class.

3.2 Feature Selection

Feature selection chooses the most significant features while reducing the dimensionality of the data. The SelectKBest method is used to accomplish this task. For feature selection and dimensionality reduction on sample sets, the classes in the `sklearn.feature_selection` module can be used to increase the accuracy scores of estimators or improve their performance on extremely high-dimensional datasets. The SelectKBest class is provided by the Scikit-Learn API to extract the best features from a given dataset. SelectKBest chooses features based on their highest scores (k). The Chi-Square test for feature selection is used to evaluate the performance score of each feature, as shown in Equation (1). Equation (2) shows the SelectKBest method for choosing the most important features. A 95% confidence interval is used by the proposed Chi-Square test for feature selection. In other words, a 5.0% value ($p < 0.05$) is used to determine if observed values are significant to the expected value. Any feature with a value greater than the 5.0% level will be rejected, and the features less than the 5.0% level will be selected as the topmost features needed for intrusion detection. This study uses $k = 14$ as the number of selected features based on the Chi-Square test evaluation. If all features achieve the Chi-Square test value, then the first 14 topmost features will be selected. However, if none achieve the 5.0% level of significance, all features will be used. The Chi-Square test is used because of its ability to recognize the most

important features for predicting anomaly in a network. In addition, the Chi-Square test can aid in simplifying the developed model, enhancing its accuracy, and improving its interpretability.

$$x_c^2 = \sum \frac{(o_i - E_i)^2}{E_i} \quad (1)$$

where c is the degree of freedom, O is the observed value(s), and E is the expected value(s).

$$select = SelectKBest(score_{func} = x_c^2, k = n) \quad (2)$$

where $k = n$ is the number of selected best scores of features based on the Chi-Square test x_c^2 .

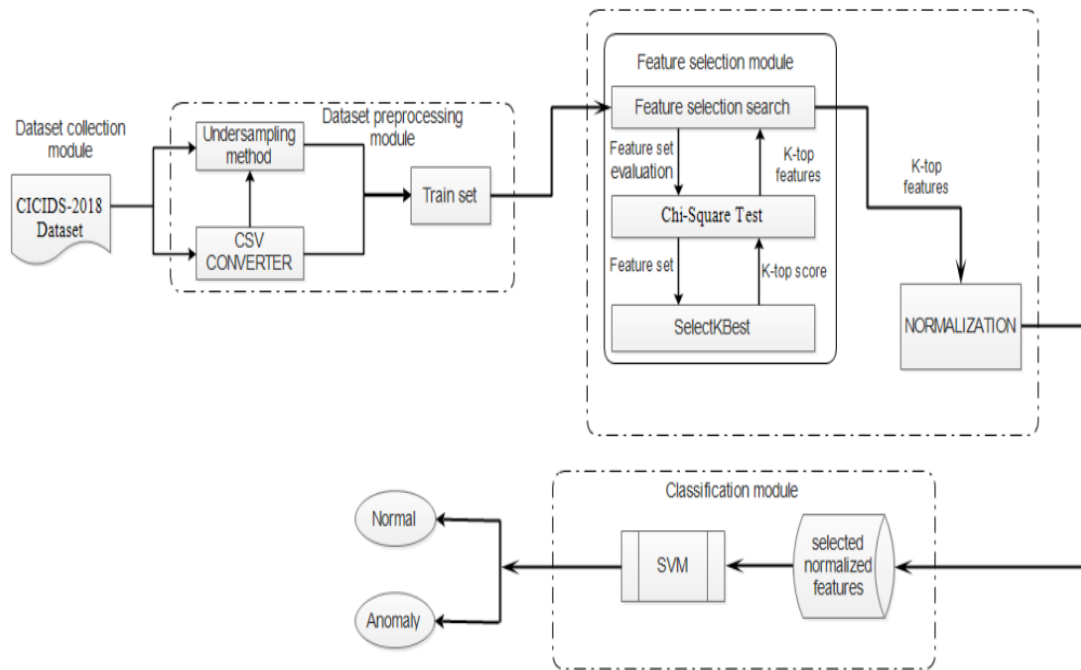


Fig. 1. Architecture of the proposed DPI model based on SVM for anomaly detection in LANs

3.3 Normalization

A data preprocessing method called normalization is used to bring feature values in a dataset to a standard scale. The goal is to ensure numbers fall within the range of 0 to 1. Specifically, min-max scaling is utilized, as shown in Equation (3). The standard scale for a particular input value in the dataset depends on its minimum and maximum values.

$$X = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

where x is the input value. The maximum and minimum values of a feature are denoted by X_{max} and X_{min} , respectively. The numerator will be zero and X will be zero when the value of x is the lowest value in the column. By contrast, the value of X equals 1 when it is the largest value in the column given that the numerator and denominator are equal. The value of X is between 0 and 1 if it falls between the lowest and maximum values.

TABLE I. LABEL DISTRIBUTION IN THE CICIDS-2018 DATASET.

S/N	Label	Instances
1	BENIGN	2272487
2	DoS Hulk	230123
3	Port-Scan	158930
4	DDoS	128027
5	DoS-GoldenEye	10293
6	FTP-Patator	7938
7	SSH-Patator	5897
8	DoS-slowloris	5796
9	DoS-Slowhttptest	5499
10	Bot	1966
11	Web Attack-Brute Force	1507
12	Web Attack-XSS	652
13	Infiltration	36
14	Web Attack-Sql Injection	21
15	Heartbleed	11

3.4 Classification

The non-probabilistic binary classifier known as SVM is based on the structural risk minimization theory from statistical learning. The goal is to build and locate the ideal margin or hyperplane that produces a satisfactory separation in an extremely high-dimensional space. More formally, given training instances as $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $x_i \in R^n$ is the n dimensional feature array, $y_i \in \{-1, +1\}$ is the class label and N is the number of instances in the dataset. A new instance x can be classified using the function described in Equation (4).

$$g(x) = \text{sign}(w \cdot x + b) \quad (4)$$

where b is the bias and w is the weight vector. The hyperplane dividing the two classes is defined by the two coefficients. When the distribution of the dataset prevents the classification problem from being linearly separable, the method can map the dataset into a higher-dimensional feature space and attempt to build the hyperplane that linearly separates the mapped vectors. As a result, a kernel function K that can replace x_i with $K(x_i)$ is adopted. This study uses SVM with a radial-basis kernel function (RBF) due to its universal kernel function and a smaller number of configurable parameters. In addition, RBF networks are easy to design, exhibit good tolerance to input noise, and possesses suitable abilities for network learning. The kernel function is defined in Equation (5).

$$K(x_i, x) = \exp\left(-\frac{1}{2\sigma^2} \|x_i - x\|^2\right) \quad (5)$$

This study aims to optimize the parameters C and σ of the RBF for enhancing the performance of the SVM. The proposed SVM is used as a binary classification for the classification of benign and anomaly classes. Given that the training complexity of SVM is highly dependent on the size of the dataset, the Chi-Square test and SelectKBest feature selection method are applied to reduce the dataset dimensionality. The overall algorithm describing the proposed method, as discussed in the entire materials and method section, is shown in Algorithm 1 below:

Algorithm 1: Algorithm for the proposed method for anomaly detection in LANs

Input: X_train; feature matrix of the training set

Y_train; labels of the training set

Output: X_resampled

Y_resampled

Y; class prediction

1. Start
 2. For 1 to N_majority samples of Y_train
 3. Count the number of N_majority
 4. Calculate the number of majority class samples to keep after undersampling
 5. $N_keep = \text{round}(\text{ratio} * N_majority)$
 6. Let N-keep = 0
 7. For each unique class label c in Y_train
 8. If c is the minority class
 9. Add all cases of minority class samples to N-keep
 10. Else if c is the majority class
 11. Randomly select N_keep cases from the majority class samples and add them to the list.
 12. Extract the corresponding feature for X_resampled and Y_resampled
 13. Return X_resampled and Y_resampled
 14. For each X_resampled and Y_resampled
 15. $x_c^2 = \sum \frac{(o_i - E_i)^2}{E_i}$
 16. $select = \text{SelectKBest}(score_{func} = x_c^2, k = n)$
 17. $X = \frac{x - x_{min}}{x_{max} - x_{min}}$
 18. $g(x) = \text{sign}(w \cdot x + b)$
 19. $K(x_i, x) = \exp\left(-\frac{1}{2\sigma^2} \|x_i - x\|^2\right)$
 20. Return Y
-

4. RESULTS

Python 3.10 is used for the implementation, given that it is an object-oriented language with a large library and documentation that offers less production code. Editing is conducted using the Jupyter Notebook IDE. PyNumPy and Pandas are used for data preprocessing. NumPy allows efficient processing of multidimensional arrays, while Pandas simplifies CSV conversion with its data frame architecture. Pandas also handles large datasets effectively. For model training, NumPy and Pandas can be utilized along with the open-source Scikit-Learn framework. Many ML functionalities are available.

4.1 Discussion

Figure 2 visually explores the relationships and correlations within the dataset. According to the graphic, no strong correlation or relationship is observed among the factors. However, after data cleaning procedures are applied, the previously noted lack of association is neutralized, and this observation changes. Figure 3 depicts normalized labels, where all attack packets are represented as 1 and all benign packets are represented as 0. The target variable used to train the SVM algorithm consists of these modified labels. In addition, Figure 3 shows that the proposed method can clearly distinguish between benign and anomalous traffic in the LAN. The feature extraction is visualized in Figure 4. Selecting important features from a large set of data entails reducing its dimensionality. The min-max scalar technique is utilized to determine the connection of each feature with the target variable. Figure 4 also demonstrates that, using

the proposed feature selection technique, the top 14 features that fall within the p-value threshold of the Chi-Square test are recommended for selection by the SelectKBest method. The bars with thicker colors represent the attributes considered for selection based on the Chi-Square test and the SelectKBest feature selection method. Table 2 shows the top features selected by the developed feature selection method.

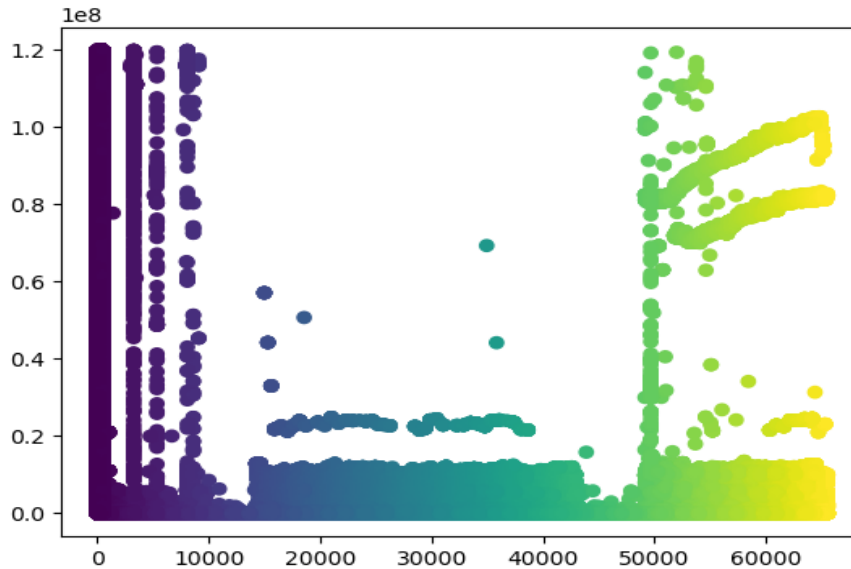


Fig. 2. Scatter plot of the dataset

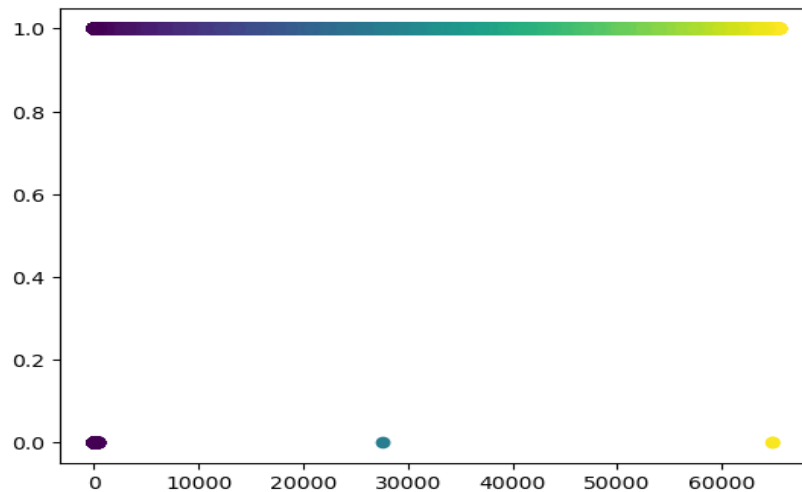


Fig. 3. Plot of label normalization to 0 and 1

Figure 5 shows a receiver operating characteristic (ROC) curve of the trained SVM model. The performance of a classification model can be assessed graphically using the ROC curve. When the discrimination threshold changes, it illustrates how a binary classifier system can be diagnostic. The trade-off between true positive rate and specificity false positive rate is graphically represented by the ROC curve as the categorization threshold varies. An AUC value closer to 1 indicates that the model can discriminate better across a range of threshold settings. An AUC value of 1 denotes a perfect classifier, whereas an AUC value of 0.5 signifies a classifier with no classification ability (equivalent to random guessing). Therefore, the model is correctly trained, and its classification performance is encouraging with an AUC value of 0.93 from the trained model. Considering that the AUC value of the trained model is close to 1, the proposed model is effective for anomaly classification in LANs.

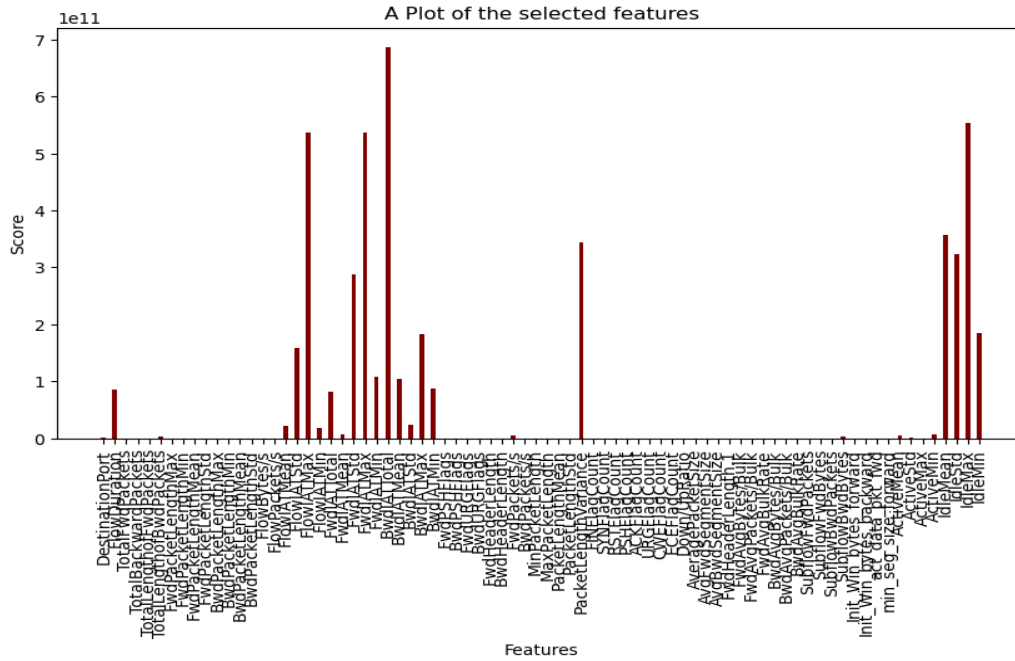


Fig. 4. Plot of selected features

TABLE II. TOP FEATURES SELECTED

S/N	Features
1	Flow Duration
2	Flow AT Std
3	Flow AT Max
4	Flow IAT std
5	Flow IAT max
6	Flow IAT min
7	Bwd IAT total
8	Bwd IAT max
9	Bwd IAT min
10	Packet length variance
11	Idle mean
12	Idle std
13	Idle max
14	Idle min

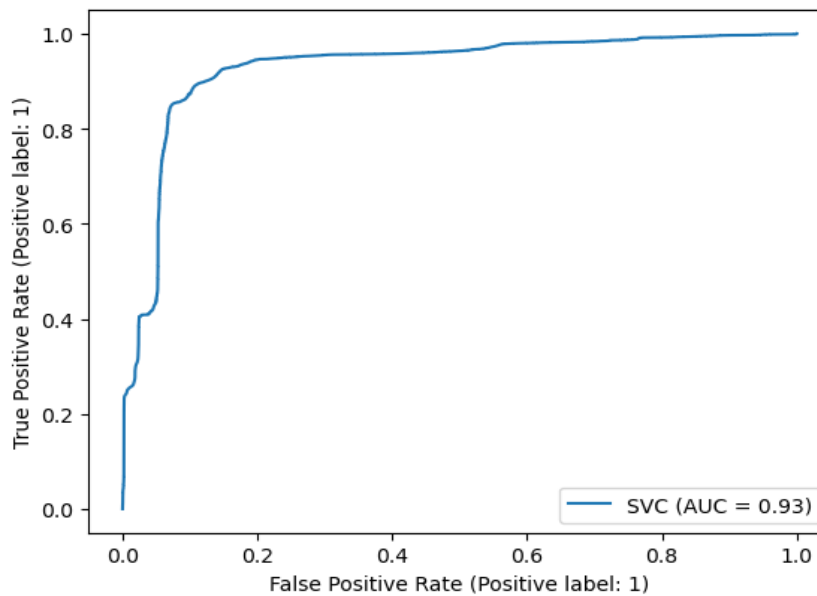


Fig. 5. Plot of selected features

Figure 6 displays the confusion matrix, which reflects the classification performance of the proposed method between benign and anomalous traffic. Out of 87,039 packets, it correctly identifies 37,253 as legitimate traffic and 40,000 as malicious. However, it incorrectly classifies 3,571 benign packets as benign when they are actually attacks, and 6,215 benign packets as attacks.

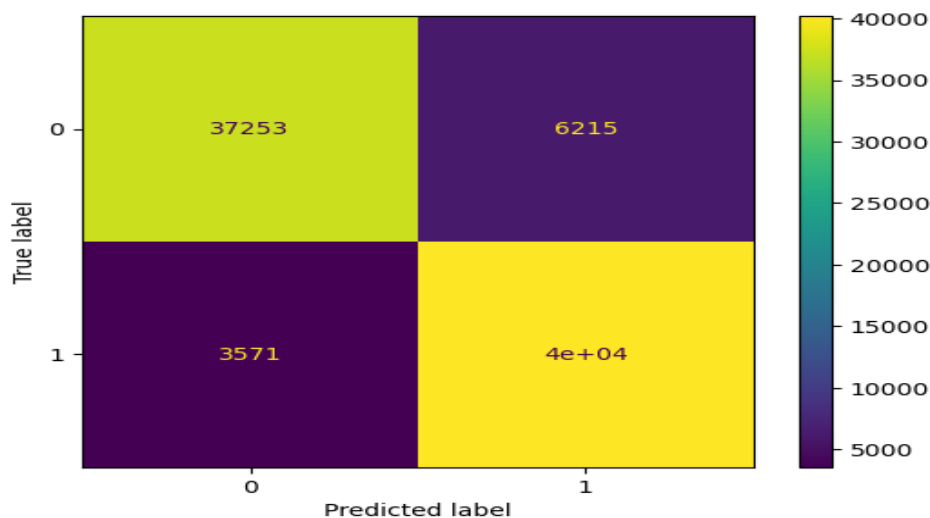


Fig. 6. Confusion matrix of test dataset on the trained SVM algorithm

Table 3 shows the performance evaluation for the study. The results show that the constructed SVM using the SelectKBest method has an accuracy of 94.81%. With minimal false positives or negatives, the accuracy results show that the SVM with the SelectKBest approach can accurately classify most network traffic. A 94.81% accuracy score is generally regarded as rather high and indicates that the system is operating at a high level. With a precision score of 94.03%, the system can accurately identify the majority of real intrusions and only a tiny number of false positives. Therefore, the system is effective in decreasing false alarms, which can help network administrators or users by lessening workload and preventing pointless inquiries. The developed SVM with the SelectKBest algorithm can properly recognize most actual intrusions, but it also misses a tiny fraction of intrusions, as indicated by the recall score of 94.13%. This result implies that additional tuning of the system might be necessary to reduce its vulnerability to specific types of attacks.

TABLE III. PERFORMANCE EVALUATION

ML Algorithm	Evaluation Metric			
	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SelectKBest + SVM	94.81	94.03	94.13	94.0799
J48 tree	92.37	93.30	94.20	93.7478
kNN	90.74	92.20	92.50	92.3498
Gradient boosting	91.92	91.70	95.30	93.4654
k-means	89.00	90.70	91.20	90.9493
Random forests	92.19	93.30	94.20	93.7478
Multilayer perceptron	90.23	93.50	90.20	91.8204

5. CONDITIONS AND CONSIDERATIONS OF THE NETWORK

The application of feature-based SVM with RBF for anomaly detection in LANs involves computer networks and routing packets in a LAN. In this study, emphasis is placed on extracting only the important features from the network dataset that affect the speed of the network in detecting anomalies. Reducing network traffic helps accelerate the detection rate and reduce delay and congestion on network paths. Thus, the Chi-Square test and the SelectKBest feature selection method is used to build an SVM classifier for real-time and accurate anomaly detection in LANs. The proposed SVM method aims to accurately distinguish between benign and anomalous traffic in the LAN.

A case study of small business networks is used as an example of a LAN to illustrate the proposed model. A small business network with multiple LANs connecting different branches is considered. These LANs connect to the Internet via a router for resource sharing. Various devices from different branches use the LAN connections to connect with one another. The network aids access to business activities and serves their respective customers. Given the numerous business activities within the local network, an effective mechanism for anomaly detection and prevention is essential. The proposed method aims to alert different branches of the business, even those located in areas away from the customers, which prompts timely action. By ensuring security within the LAN, the proposed method allows various departments from different branches to connect with one another and with customers for business transactions. In addition, it can be used in congested business branches to improve service provision to customers while securing business data. Ultimately, this method reduces network complexity and secures private information of customers and the business through the LANs.

6. CONCLUSION

This study mainly aims to implement a DPI system for anomaly detection in a local area network environment. The adopted strategy includes using the SVM method in conjunction with the CICIDS-2018 dataset to enhance network security by detecting and reacting to anomalous patterns in network traffic. The outcomes demonstrate how well the SVM algorithm classifies typical and unusual network behaviors. By leveraging DPI-derived features, the model exhibits flexibility and generalization abilities in response to a range of attack scenarios. Therefore, it is a trustworthy IDS. A thorough performance evaluation of the DPI and SVM model is made feasible using the CICIDS-2018 dataset, which offers a diverse range of scenarios. Remarkably, the system successfully identifies known and unexpected anomalies, which adds strength to the network security solution. The study results demonstrate a minimal false positive rate, which guarantees the suitability of the detection system for practical implementation in real-world scenarios.



Conflicts of Interest

The authors declare no conflicts of interest.

Funding

None

Acknowledgment

None

References

- [1] N. Miloslavskaya and A. Tolstoy, "Internet of Things: information security challenges and solutions.," *Cluster Computing*, vol. 22, pp. 103-119, 2019.
- [2] R. K. Sharma, B. Issac, Q. Xin, T. R. Gadekallu and K. & Nath, "Plant and Salamander Inspired Network Attack Detection and Data Recovery Model.," *Sensors*, vol. 23, no. 12, p. 5562, 2023.
- [3] M. S. Pour, C. Nader, K. Friday and E. & Bou-Harb, "A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security.," *Computers & Security*, p. 103123., 2023.
- [4] V. Vasani, A. K. Bairwa, S. Joshi, A. Pljonkin, M. Kaur and M. & Amoon, "Comprehensive Analysis of Advanced Techniques and Vital Tools for Detecting Malware Intrusion.," *Electronics*, vol. 12, no. 20, p. 4299, 2023.
- [5] K. Gaur, A. Kalla, J. Grover, M. Borhani, A. Gurtov and M. & Liyanage, "A survey of virtual private LAN services (VPLS): Past, present and future.," *Computer Networks*, vol. 196, p. 108245, 2021.
- [6] E. Fazel, H. E. Najafabadi, M. Rezaei and H. & Leung, "Unlocking the Power of Mist Computing through Clustering Techniques in IoT Networks.," *Internet of Things*, p. 100710, 2023.
- [7] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz and E. & Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions.," *Electronics*, vol. 12, no. 6, p. 1333, 2023.
- [8] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi and M. L. & Proença, "A comprehensive survey on network anomaly detection.," *Telecommunication Systems*, vol. 70, pp. 447-489, 2019.
- [9] S. Aminizadeh, A. Heidari, S. Toumaj, M. Darbandi, N. J. Navimipour, M. Rezaei and M. ... & Unal, "The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things.," *Computer Methods and Programs in Biomedicine*, p. 107745, 2023.
- [10] A. Heidari and M. A. & Jabraeil Jamali, "Internet of Things intrusion detection systems: A comprehensive review and future directions.," *Cluster Computing*, vol. 26, no. 6, pp. 3753-3780, 2023.
- [11] A. Khraisat, I. Gondal, P. Vamplew and J. & Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges.," *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019.
- [12] N. Moustafa, J. Hu and J. & Slay, "A holistic review of network anomaly detection systems: A comprehensive survey.," *Journal of Network and Computer Applications*, vol. 128, pp. 33-55, 2019.
- [13] N. Jeffrey, Q. Tan and J. R. & Villar, "A review of anomaly detection strategies to detect threats to cyber-physical systems.," *Electronics*, vol. 12, no. 15, p. 3283, 2023.
- [14] N. Ashraf, W. Ahmad and R. & Ashraf, "A comparative study of data mining algorithms for high detection rate in intrusion detection system.," *Annals of Emerging Technologies in Computing (AETiC)*, pp. 2516-0281, 2018.
- [15] S. A. Agah, "Investigating identification techniques of attacks in intrusion detection systems using data mining algorithms.," *International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 174-181, 2017.
- [16] Y. Zhou, G. Cheng, S. Jiang and M. & Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier.," *Computer networks*, vol. 174, p. 107247, 2020.



- [17] J. Liu, X. Song, Y. Zhou, X. Peng, Y. Zhang, P. Liu and C. ... & Zhu, "Deep anomaly detection in packet payload," *Neurocomputing*, vol. 485, pp. 205-218, 2022.
- [18] D. H. Summerville, K. M. Zach and Y. & Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," in 34th international performance computing and communications conference (IPCCC), Nanjing, China, December 14 – 16, 2015.
- [19] D. Spiekermann and J. & Keller, "Unsupervised packet-based anomaly detection in virtual networks," *Computer Networks*, vol. 192, p. 108017, 2021.
- [20] S. Basumallik, R. Ma and S. & Eftekharnjad, "Packet-data anomaly detection in PMU-based state estimator using convolutional neural network," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 690-702, 2019.
- [21] H. Liu and H. & Wang, "Real-Time Anomaly Detection of Network Traffic Based on CNN," *Symmetry*, vol. 15, no. 6, p. 1205, 2023.
- [22] A. Lara, V. Mayor, R. Estepa, A. Estepa and J. E. & Díaz-Verdejo, "Smart home anomaly-based IDS: Architecture proposal and case study," *Internet of Things*, vol. 22, p. 100773, 2023.
- [23] W. Song, M. Beshley, K. Przystupa, H. Beshley, O. Kochan, A. Pryslupskyi and J. ... & Su, "A software deep packet inspection system for network traffic analysis and anomaly detection," *Sensors*, vol. 20, no. 6, p. 1637, 2020.
- [24] L. Deri, M. Martinelli, T. Bujlow and A. & Cardigliano, "ndpi: Open-source high-speed deep packet inspection.," in International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 04-08 August 2014.
- [25] H. Doroud, G. Aceto, W. de Donato, E. A. Jarchlo, A. M. Lopez, C. D. Guerrero and A. & Pescape, "Speeding-up dpi traffic classification with chaining," in IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9-13 December 2018.
- [26] R. Patel, A. Thakkar and A. & Ganatra, "A survey and comparative analysis of data mining techniques for network intrusion detection systems," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 1, pp. 265-260, 2012.
- [27] G. Chandrashekar and F. & Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16-28, 2014.
- [28] U. Stańczyk, "Feature evaluation by filter, wrapper, and embedded approaches," *Feature Selection for Data and Pattern Recognition*, pp. 29-44, 2015.
- [29] T. Zhao, Y. Zheng and Z. & Wu, "Feature selection-based machine learning modeling for distributed model predictive control of nonlinear processes," *Computers & Chemical Engineering*, Vols. 169, p. 108074, 2023.
- [30] J. Liu, X. Song, Y. Zhou, X. Peng, Y. Zhang, P. Liu and C. ... & Zhu, "Deep anomaly detection in packet payload," *Neurocomputing*, vol. 485, pp. 205-218, 2022.