International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Support Vector Machine with Grid Search Cross-Validation for Network Intrusion Detection in Cloud

N. Sujata Kumari[1], Naresh Vurukonda[2]

Submitted: 10/12/2023 **Revised**: 14/01/2024 **Accepted**: 31/01/2024

Abstract: An Intrusion Detection System (IDS) is the process of monitoring a system that detects suspicious activities and produces alerts when they are detected. The Network Intrusion Detection System (NIDS) analyzes traffic across all subnets to identify any data flows originating from subnets associated with known attack patterns. The drawback of NIDS is it gives more frequent false positives than the actual threats and it can be reduced by tuning the IDS. In this research, the Support Vector Machine (SVM) with Grid Search cross-validation (CV) is proposed for the network intrusion detection system. The dataset utilized for research is NSL-KDD and the pre-processing techniques utilized are data cleaning, label encoding and robust scalar. The Principal Component Analysis (PCA) is utilized as a feature selection technique and detection is performed by SVM that is optimized by grid search cross validation. The proposed algorithm superiorly detected the network intrusion with less noise and false positives. The proposed algorithm is evaluated by utilizing performance measures of accuracy, precision, recall and f1-score. The proposed algorithm attained high accuracy 99.53%, precision 99.26%, recall 99.18% and f1-score 99.22% which is comparatively superior to other existing methods like Naive Bayes (NB), Random Forest (RF), and Logistic Regression (LR).

Keywords: *Grid search cross validation, Intrusion detection system, Label encoding, Principal component analysis, Robust scalar, Support vector machine*

1. Introduction

The enormous development of the internet and its huge spread has a major challenge in cybersecurity to detect suspicious activities [1]. Cybersecurity is organizational, technical, and administrative for preventing unofficial use of electronic data and communication devices [2]. An Intrusion Detection System (IDS) is a difficult solution for detecting and predicting the threats of cybersecurity [3]. There are two types of IDS Host-Based IDS (HIDS) and Network-Based IDS (NIDS). IDS detects the attacks in two ways Signature (SIDS) and Anomaly-Based IDS (AIDS) [4]. The SIDA matches the messages with their database to know whether the message is malicious or normal [5]. It decides while it contains previous knowledge of attack type [6]. On the other hand, A-IDS has the behavior of network information and analyses that recognize malicious and anomaly messages [7]. It predicted zero-day attacks without previous identification, so it is low accurate and suffers from a huge false alarm rate and computational time complexity [8]. The A-IDS uses the Machine Learning (ML) method to predict if the observation is malicious or normal behavior. To develop an effective NIDS, three main challenges in network security can be addressed [9]. The first challenge is the development of information volumes over time, the penetration rate of the internet can be maximized and new techniques like IoT can maximize the speed of data production [10]. It can be a significant challenge due to IDSs must utilize the data volume [11]. The second challenge is comprehensiveness, all information's correctly reviewed and huge-level features must be ignored or less-level features must be utilized to extract them then, IDS provides superior performance and accuracy [12,13]. Ignoring huge-level features means that any modification that occurs in the network can be featured to any network attributes such as utilized protocol, part of OS, or browser [14]. A third significant challenge is the number and different protocols utilized in networks, which alone develops a huge level of complexity for IDSs [15]. Different techniques are utilized to create NIDS like Decision Trees, Navie-Bayes, and Support Vector Machine. The drawback of NIDS is it gives more frequent false positives than the actual threats and it can be reduced by tuning the IDS. The SVM and various other machine learning methods have the drawback of over or under-fitting whether the hyperparameters are not tuned properly. The overfitting occurs when the method captures the noise in training data which causes less generalization on new data. The underfitting occurs when the method is simple to capture the patterns which are underlying in data. The hyperparameter tuning helps the model balance to attain superior model complexity. In this research, the Support Vector Machine (SVM) with Grid Search cross-validation (CV) is proposed for the network intrusion detection system. The dataset utilized for research is NSL-KDD and the pre-processing techniques utilized are data cleaning, label encoding and robust scalar. The Principal Component Analysis (PCA) is utilized as a feature selection technique and detection is performed by SVM that is optimized by grid search cross-validation. The main contribution of research is as follows:

- The data cleaning, label encoding and robust scalar are utilized for the pre-processing techniques which enhance the data quality and convert the non-numerical features to numerical ones.

[1] *Research scholar, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India; nsujata02@gmail.com*

[2] *Associate Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India; naresh.vurukonda@kluniversity.in*

* *Corresponding Author Email: nsujata02@gmail.com*

- The Principal Component Analysis (PCA) is utilized for selecting relevant features that selects the relevant features with less dimensionality.
- The Support Vector Machine (SVM) is tuned by Grid Search Cross-Validation for detecting the intrusion in the network with less noise and false positives.

The rest of the research is organized as follows: Section 2 gives a literature review on network intrusion detection. Section 3 gives details of the proposed algorithm. Section 4 gives results and discussion of proposed algorithm and section 5 concludes the manuscript.

2. Literature Review

Vishwakarma and Kesswani [16] implemented a Two Stage Intrusion Detection System (IDS) to detect network intrusion using the NSL-KDD dataset. In the first phase, separate the data into four categories by data type. Then that was classified by various versions of Naive Bayes. Next, utilize major voting to select the output of classification. In the second stage, pass information that behaved normally or benign in the previous stage and classify them. The implemented method considered the benefits of unlabeled data. However, the method has insufficient characteristics of data.

Wang et al. [17] introduced a Stacked Contractive Auto Encoder (SCAE) and Support Vector Machine (SVM) method to detect network intrusion in NSL-KDD dataset. The SCAE technique was introduced to extract the unsupervised features that automatically learned superior and less dimensional features from network traffic. Then, these are classified through the introduced SVM method. The introduced method has huge precision in intrusion detection. However, the method can’t recognize certain new attacks present in the dataset.

Alfoudi et al. [18] presented an Evolving Density-Based Spatial Clustering of Application with Noise (EDBSCAN) method to detect network intrusion in NSL-KDD dataset. The presented technique used a new evolving technique for minimizing the count of clusters to highly compatible ones. Furthermore, the presented technique was developed based on cosine similarity for predicting abnormal behavior. The presented method addressed the general problem of class imbalance datasets. However, the presented method has less detection accuracy and rate.

Mohamed and Ejbali [19] suggested a State-Action-Reward-State-Action-based Reinforcement Learning (SARSA-based RL) method to detect network intrusion using the NSL-KDD dataset. The suggested technique was acquired through certain changes to the deep SARSA technique. The technique was combined with DNN which 5 layers to enhance the accuracy of multi-class classification. The suggested method attained a high detection accuracy and rate. However, the presented method has less data for training.

Kasongo [20] developed a Recurrent Neural Network (RNN) method to detect network intrusion in NSL-KDD dataset. The developed technique has various types of RNNs such as Gated Recurrent Unit (GRU), long-term memory (LSTM) and general RNN. XGBoost technique was introduced to minimize the feature space of every dataset. The developed method enhanced the performance of minority classes. However, detection accuracy for attack classes with small training data was low.

Alazzam et al. [21] implemented a One-Class Support Vector Machine (OCSVM) method to detect network intrusion in the NSL-KDD dataset. The implemented method has fusion among two major subsystems which process in parallel. Every subsystem was trained by utilizing OCSVM. One method was trained by normal packets the other was trained by attack packets. The outcome of both subsystems was integrated to provide superior judgment for every packet that passes through the network. The implemented technique has a high detection rate for known attacks. However, failed to detect unknown attacks.

Hosseini and Sardo [22] introduced a Hybrid method of Deep and Shallow Learning method to detect network intrusion in the NSL-KDD dataset. The introduced method initially utilized a spider monkey optimization algorithm for feature selection to choose many significant features, next Siamese neural network was introduced to make information much classifiable. The introduced method has less computational burden. However, the introduced method failed to identify a suitable fitness function.

Cao et al. [23] suggested a Split-Residual-Fuse Convolutional Neural Network and Bidirectional Gated Recurrent Unit (SRFCNN-BiGRU) method for detecting network intrusion in the NSL-KDD dataset. The hybrid method integrating feature selection was done by integrating Random Forest (RF) and Pearson Correlation Analysis. Then the spatial features were extracted through a Convolutional Neural Network and the next extracted features through average and max pooling and BiGRU was utilized for extracting long distance dependent data to attain efficient features. At last, the SoftMax function was utilized for classification. The suggested method can group huge datasets into little ones. However, the method increased training time and produced a huge false positive rate.

The existing methods have drawbacks like high training time, producing a high false positive rate, failure to identify suitable fitness functions, failure to detect unknown attacks, and can’t recognize certain new attacks present in the dataset. The drawback of NIDS is it gives more frequent false positives than the actual threats and it can be reduced by tuning the IDS. The SVM and various other machine learning methods have the drawback of over or under-fitting whether the hyperparameters are not tuned properly. The overfitting occurs when the method captures the noise in training data which causes less generalization on new data. The underfitting occurs while the method is simple to capture the patterns that are underlying data. The hyperparameter tuning helps the model balance to attain superior model complexity.

3. Proposed Method

In this methodology, the Support Vector Machine (SVM) with Grid Search cross-validation (CV) is proposed for the network intrusion detection system. The dataset utilized for research is NSL-KDD and the pre-processing techniques utilized are data cleaning, label encoding and robust scalar. The Principal Component Analysis (PCA) is utilized as a feature selection technique and detection is performed by SVM that is optimized by grid search cross-validation. Fig. 1 represents the process of the proposed algorithm.

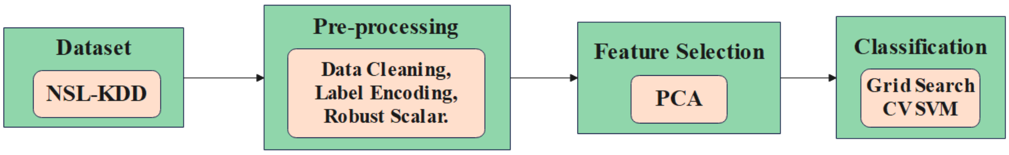

Fig 1. Process of Proposed Algorithm

3.1. Dataset

The dataset utilized for the research of intrusion detection is the NSL-KDD dataset [24] (accessed on Oct 2023) which is an enhancement over the KDD Cup 99 dataset. The dataset has 125,973 training samples of network traffic and 22,544 testing samples, every record has 41 attributes representing flow features and a label is employed for each sample. The features are divided into three types such as basic, content, and traffic features. The visualization of data, attacks in the dataset and major website service user are represented in Fig. 2, Fig. 3 and Fig. 4.

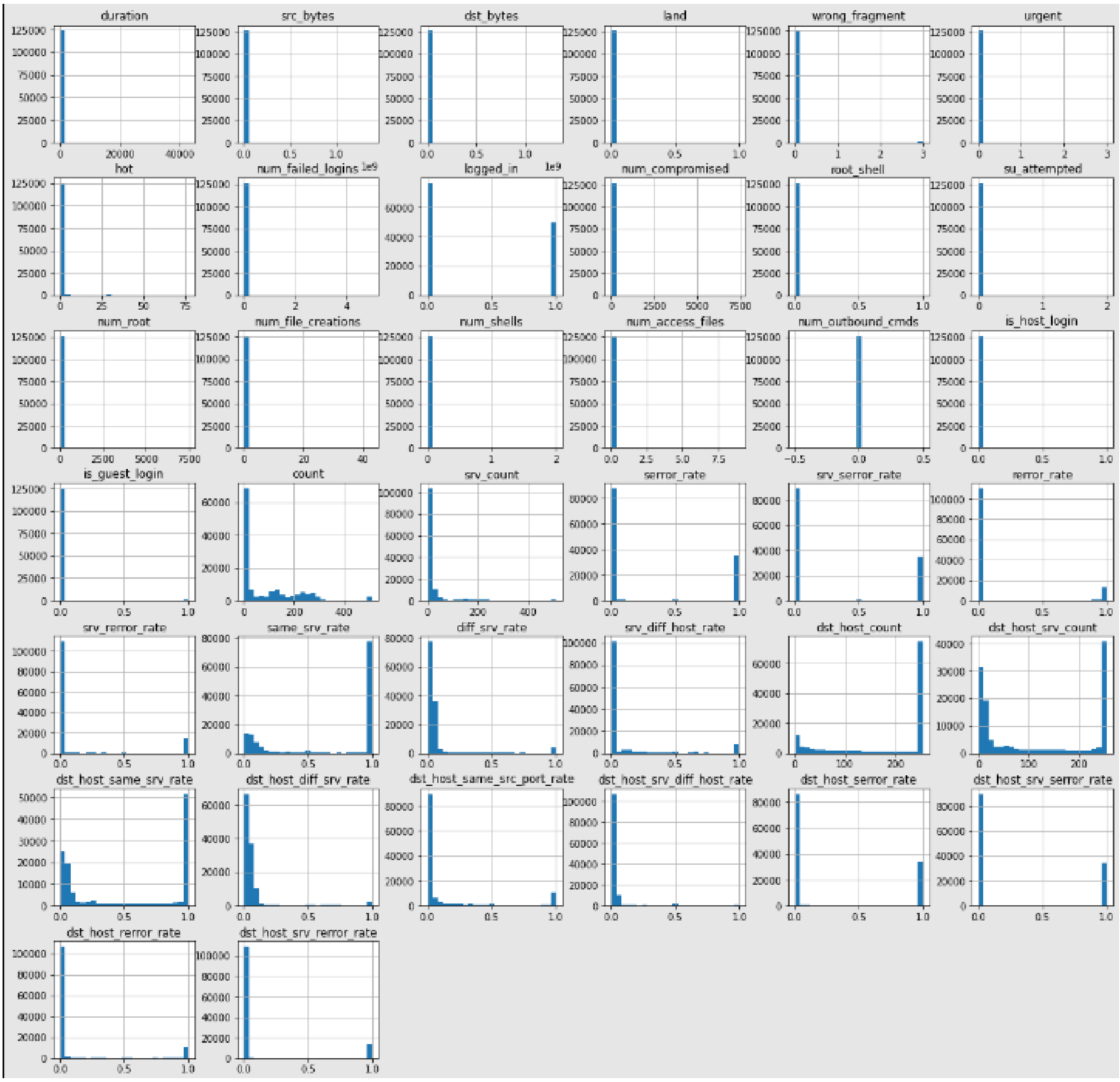

Fig 2. Data Visualization

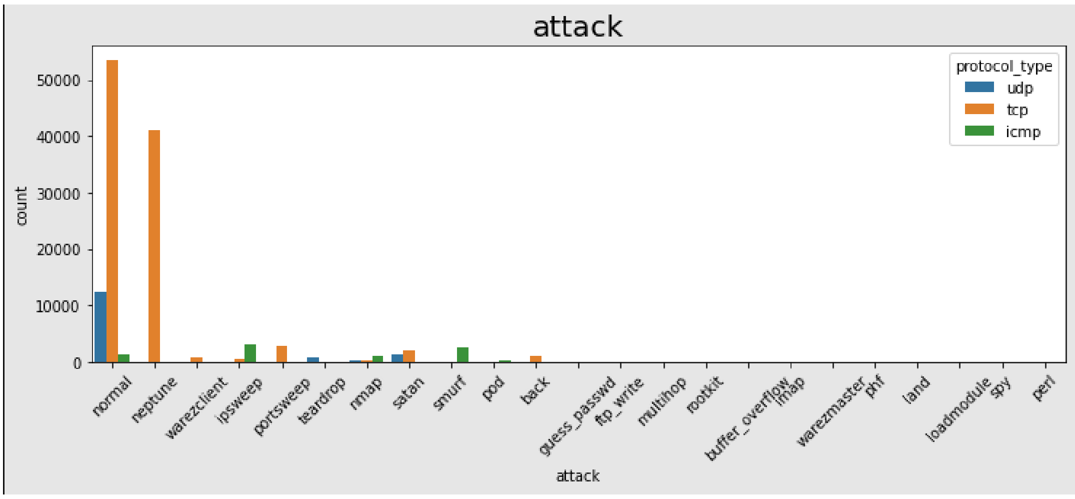

Fig 3. Attacks in dataset

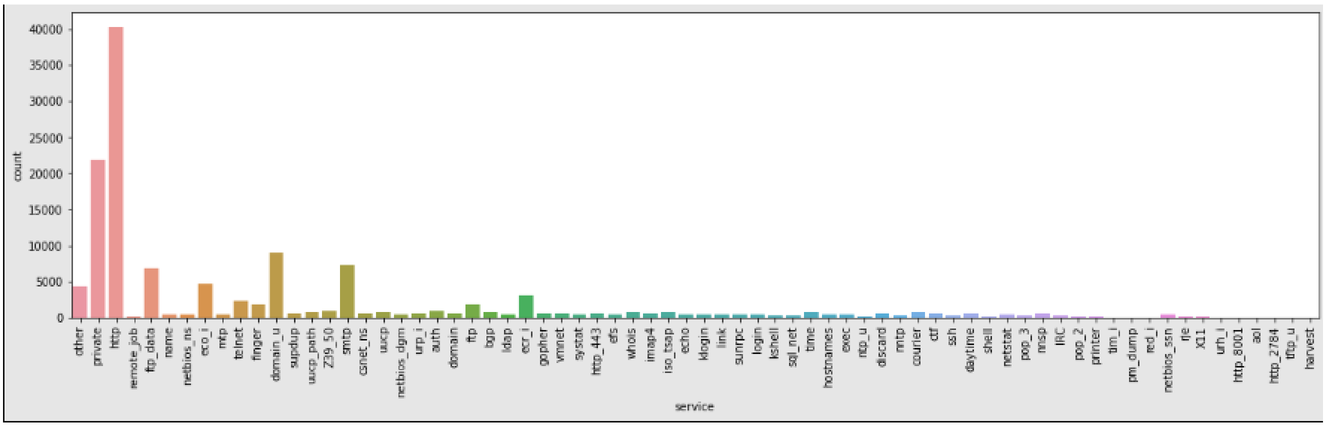

Fig 4. Major Website Services used

3.2. Pre-processing

The preprocessing techniques utilized for the research are data cleaning, label encoder and robust scalar. These pre-processing techniques enhanced the quality of data in NSL-KDD dataset.

3.2.1. Data Cleaning

Cleaning of data is a process of deleting irrelevance or duplicate records and handling missing data. The cleaning of data is a significant stage in enhancing the consistency, validity and usability of data. It is significant to delete duplicate records in the training set to ignore the classification technique to be biased to major common records and protect it from learning uncommon records.

3.2.2. Label Encoding

Label encoding is utilized for converting non-numerical features in actual data traffic to numerical ones to make the model learn features of data. In the NSL-KDD dataset, certain features are in the format of the string, those features are converted into the numerical format by using label encoding. The numerical values are encoded in place of string values and minimize the count of data which results the less memory.

3.2.3. Robust Scalar

The robust scalar utilized the same as min-max normalization. The only difference is that it uses an interquartile range instead of min max. The robust scalar scales the data by quintile range. The mathematical formula of robust scale is represented in (1)

$$x = \frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \tag{1}$$

The pre-processed data is given as input to PCA for selecting the relevant features in the dataset. By selecting relevant features that enhances the model performance.

3.3. Feature Selection

Feature selection is the process of choosing a relevant feature in a dataset that enhances the performance of the classification model. In this research, the principal Component Analysis (PCA) is utilized for selecting the features. Both feature extraction and selection are utilized for the reduction of dimensionality. The PCA-based feature extraction takes x as an eigen vector of the covariance matrix of PCA. The result of feature extraction is based on x of arbitrary sample vector a and the mathematical formula is represented in (2),

$$z = a^T x = \sum_{i=1}^{N} a_i x_i \tag{2}$$

Where $x = [x_1 \dots x_N]^T$, $a = [a_1 \dots a_N]^T$, N represents the sample vector dimensionality.

The absolute value of $x_i (i = 1,2, \dots N)$ is capable of analytically estimating the contribution to the result of feature extraction of ith feature element of samples. The lesser correct value of x_i, small the contribution of ith feature element of samples. Whether, the correct value of x_k is less enough, deleting a_k and x_k from $\sum_{i=1}^{N} a_i x_i$ which does not affect the result of feature extraction. The feature selection by PCA is performed by the following procedure:

Step 1: Measure PCA's covariance matrix utilizing actual training samples. Next, resolve whole eigenvectors and values.

Step 2: Choose eigenvectors respective to the initial m highest eigenvalues and represent these eigenvectors by $V_1, \dots, V_m$ respectively.

Step 3: Measure the contribution to the outcome of feature extraction of jth feature element as follows (3)

$$c_j = \sum_{p=1}^{m} |V_{pj}| \tag{3}$$

Where, V_{pj} represents jth entry of V_p, $j = 1,2, \dots N$ and $p = 1,2, \dots m$, $|V_{pj}|$ represents the correct value of V_{pj}.

Step 4: Sort the c_j in descending order and utilize $d_j$to save order, where $j = 1,2, \dots N$.

For instance, the c_s and c_t represents first and second highest among the whole c_j, $j = 1,2, \dots N$, then $d_1 = s$ and $d_2 = t$, where, sth and tth feature elements of actual samples are two major significant features. Whether $n -$ dimensional features are needed, then feature selection result can be $d_1th, d_2th, \dots, d_nth$ feature elements. The PCA selects the relevant features into lower-dimensional space and classifies the test samples in new space. The selected features are given as input to SVM for detecting the intrusion in the network.

3.4. Classification

A classification model is utilized for detecting the network intrusion in the NSL-KDD dataset by Support Vector Machine (SVM). The SVM model is optimized by using Grid Search Cross-Validation. The process of SVM with Grid Search Cross-Validation is discussed below:

3.4.1. Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised machine learning algorithm that is utilized for classification and regression. In SVM, data is described in $n -$ dimensional space where that predict is fresh training occurrence goes to similar or various classes. Comparing the similar or various classes, namely among categories with values 0 or 1.

SVM aims to identify hyperplane in $n -$ dimensional space which classified information points. The SVM utilizes a straight-line Kernel which separates two classes through linear (4),

$$w * x - b = 0 \tag{4}$$

Where, w represents hyperplane parameter, x represents input data and b represents bias.

The method to generate optimal hyperplane in SVM is performed by (5) and (6),

$$min \frac{1}{2} \|\omega\|^2 \tag{5}$$

$$y_i(wx_i + b) \geq 1, i = 1, \dots, \lambda \tag{6}$$

The above (6) utilized to increase of $\|\omega\|^2$ value along with concentration with $y_i(wx_i + b)$. Whether the result of the information is $y_i = +1$, then $y_i(wx_i + b)$ becomes $(wx_i + b) \geq 1$. Whether $y_i = -1$, then $y_i(wx_i + b)$ becomes $(wx_i + b) - 1$. The SVM model is hyperparameter tuned by grid search CV to enhance its performance.

3.4.2. Grid Search Cross Validation

To enhance the accuracy of the SVM model, it is optimized by utilizing a grid search CV. The grid search is the selection of integration methods and hyperparameters through testing integrations and validating every integration.

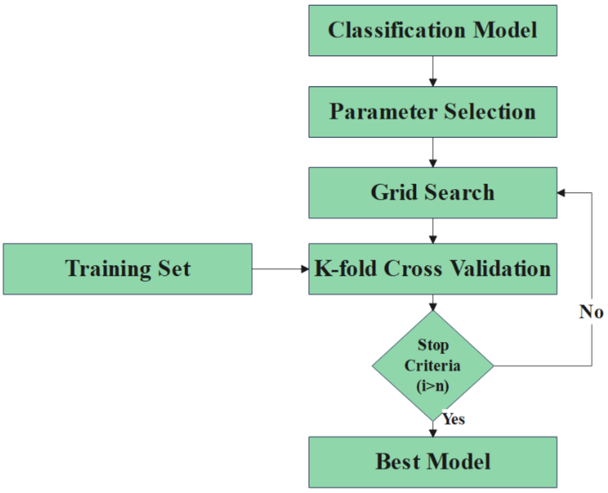

Fig 5. Process of Optimization using Grid Search Cross Validation

Grid search aims to regulate integration which generates superior performance that can taken for model prediction. Grid search is generally integrated with k-fold CV, developing an estimation index for the classification method. K-fold cv is repeated training and testing data many times till the k repetitions and $1/k$ partition of data utilized as test information. The accuracy of k method is acquired and the performance of the k-fold cv classification method is estimated by mean accuracy of the k model. Then, classification parameters are modified by grid search and the accuracy of the classification method is remeasured. The procedure of optimization is represented in Fig. 5. In above Fig. 5, i represents the current iteration and n represents the maximum number of iterations. The accuracy of the classification method along the whole integration of parameters is compared to generate the highest accuracy score. The SVM is tuned by Grid Search CV which reduces the noise and false positives to enhance the IDS performance. The proposed method provides superior performance and detects intrusion in network traffic.

4. Experimental Analysis

The detected intrusion in network traffic is simulation and evaluation is described in this section. The proposed method is simulated by python 3.7 environment, OS: Windows 10 (64 bit), Processor: intel core i7 and RAM: 16 GB. The performance measures utilized for the proposed method evaluation accuracy, precision, recall and f1-score. Tables, figures, confusion matrix and ROC curve graph are described below to evaluate the performance of the proposed method.

4.1. Quantitative and Qualitative Analysis

The performance of the proposed method is estimated with actual features, after selecting the features and SVM with various hyperparameters. The Confusion matrix and ROC curve graph are represented in this section to evaluate the method performance. The proposed method is evaluated with other existing methods like Naive Bayes (NB), Random Forest (RF) and Logistic Regression (LR).

Table 1. Performance of model with actual features.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NB	83.74	82.93	80.73	81.49
RF	88.82	85.72	84.46	84.62
LR	91.28	90.38	89.83	88.11
SVM	95.34	93.26	92.57	92.35

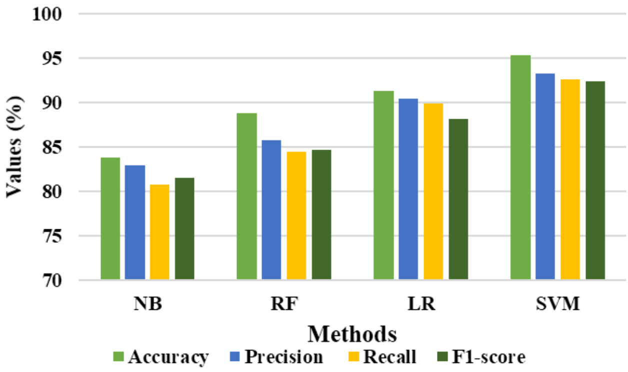

Fig 6. Performance of model with actual features

Table 1 and Fig. 6 performs the proposed method with actual features. The SVM method attained a high accuracy 95.34%, precision 93.26%, recall 92.57% and f1-score 92.35% which is superior than other existing methods utilized for evaluation. The existing methods such as NB attained the accuracy of 83.74%, RF attained the accuracy of 88.82% and LR attained the accuracy of 91.28% which is poorly performed when compared with SVM model.

Table 2. Performance of model after feature selection.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NB	90.81	91.93	90.77	92.84
RF	92.48	94.57	93.61	94.71
LR	95.27	97.03	96.49	96.37
SVM with Grid Search CV	99.53	99.26	99.18	99.22

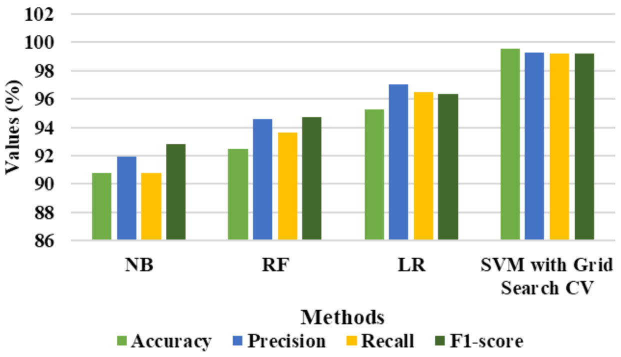

Fig 7. Performance of model after feature selection

In table 2 and Fig. 7 performs the proposed method after feature selection. After the SVM method was optimized with grid search CV attained the high accuracy 99.53%, precision 99.26%, recall 99.18% and f1-score 99.22% which is superior than other existing methods utilized for evaluation. The existing methods such as NB attained the accuracy of 90.81%, RF attained the accuracy of 92.48% and LR attained the accuracy of 95.27% which is poorly performed when compared with proposed SVM with grid search CV model.

Table 3. Performance of Proposed Method with Other Hyperparameters.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SVM with kernel	94.28	94.02	93.39	93.20
SVM with C	96.55	95.73	95.48	95.37
SVM with gamma	97.47	96.21	97.05	96.64
SVM with Grid Search CV	99.53	99.26	99.18	99.22

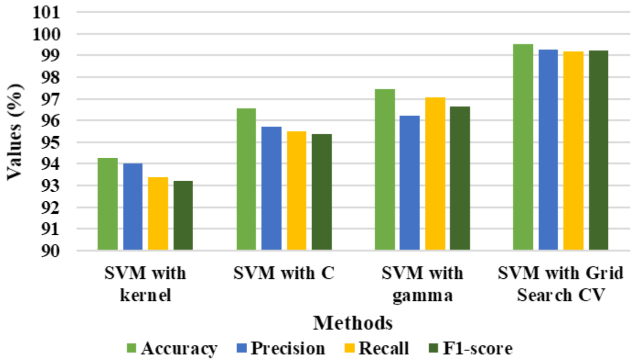

Fig 8. Performance of Proposed Method with Other Hyperparameters.

Table 3 and Fig. 8 perform the proposed method with other hyperparameters. After the SVM method optimized with grid search CV attained a high accuracy 99.53%, precision 99.26%, recall 99.18% and f1-score 99.22% which is superior than other existing methods utilized for evaluation. The existing methods such as SVM with kernel attained the accuracy of 94.28%, SVM with c attained the accuracy of 96.55% and SVM with gamma attained the accuracy of 97.47% which is poorly performed when compared with proposed SVM with grid search CV model.

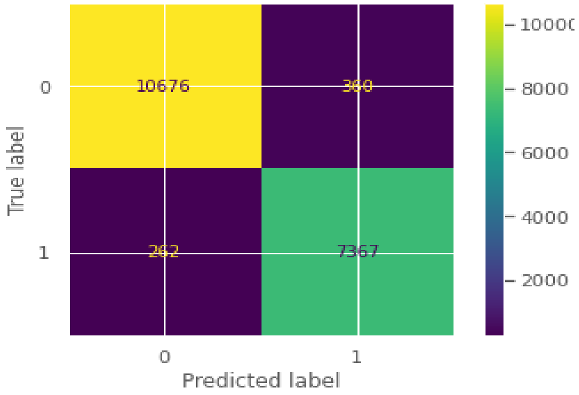

Fig 9. Confusion Matrix for Grid Search CV SVM

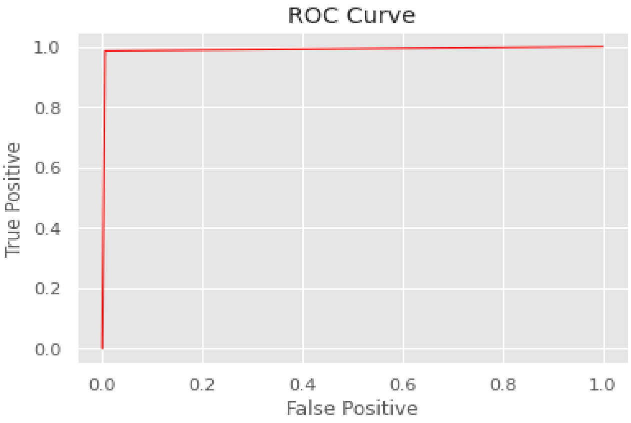

Fig 10. ROC Curve Graph for Grid Search CV SVM

Fig. 9 and Fig. 10 represents the confusion matrix and ROC Curve graph for Grid Search CV SVM respectively. A Confusion matrix is a table that is utilized to determine the performance of classification methods that visualizes and summarizes the method performance. A Receiver Operating Characteristic Curve (ROC Curve) is a graph that shows the performance of the classification method at every threshold of classification.

4.2. Comparative Analysis

The performance of the proposed algorithm is compared with other existing research and described in Table 4. The existing methods utilized for comparison are the Two Stage Intrusion Detection System (IDS) [16], SCAE + SVM [17] and EDBSCAN [18]. The proposed method attained a high accuracy of 99.53% which is superior to other existing methods like [16] attained 97.1%, [17] attained 89.93% and [18] attained 86.82% which is less than the proposed technique.

Table 4. Comparative Analysis

Method	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Two Stage Intrusion Detection System (IDS) [16]	NSL-KDD	97.1	96	99.1	97.5
SCAE + SVM [17]		89.93	91.04	89.93	86.94
EDBSCAN [18]		86.82	88.95	86.82	87.87
Proposed SVM with Grid Search CV		99.53	99.26	99.18	99.22

4.3. Discussion

In this section, the drawbacks of existing algorithms and the advantages of the proposed algorithm are discussed. The Two Stage Intrusion Detection System (IDS) [16] has insufficient characteristics of data. The SCAE + SVM [17] method can't recognize certain new attacks present in the dataset. The EDBSCAN [18] method has huge noise and false positives that reduce the IDS effectiveness. In this research, the NSL-KDD dataset is utilized which has enough characteristics and features for intrusion detection. The PCA is implemented on actual features to select the relevant features with huge quality and SVM is trained by transformed data to develop IDS which detects the new attacks present in the dataset. The SVM is tuned by Grid Search CV which reduces the noise and false positives to enhance the IDS performance. The proposed algorithm attained high accuracy 99.53%, precision 99.26%, recall 99.18% and f1-score 99.22% which is comparatively superior to other existing methods like NB, RF and LR.

5. Conclusion

The NIDS is the process of detecting malicious or suspicious activity in network traffic that helps to analyze the types and quantity of attacks. The drawback of NIDS is it gives more frequent false positives than the actual threats and it can be reduced by tuning the IDS. In this research, the SVM is proposed as IDS and it is tuned by grid search CV which reduces the noise and false positives. The dataset utilized for the research is the NSL-KDD dataset and data cleaning, label encoding, and robust scalar are utilized as pre-processing techniques. The PCA is utilized for feature selection that selects relevant features with less dimensionality. The proposed algorithm attained a high accuracy of 99.53%, precision of 99.26%, recall of 99.18% and f1-score of 99.22% which is comparatively superior to other existing methods like NB, RF and LR. The SVM is tuned by grid search CV for effective network intrusion detection which detects the intrusion with less noise and false positives. In future, deep learning algorithms can be considered for IDS with more datasets.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] M. Chalé and N. D. Bastian, "Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems," *Expert Syst. Appl.*, vol. 207, p. 117936, Nov. 2022. https://doi.org/10.1016/j.eswa.2022.117936

[2] P. B. Udas, Md. E. Karim, and K. S. Roy, "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10B, pp. 10246–10272, 2022. https://doi.org/10.1016/j.jksuci.2022.10.019

[3] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2330–2345, 2023. doi: 10.1109/JIOT.2022.3211346

[4] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Comput. Electr. Eng.*, vol. 102, p. 108156, 2022. https://doi.org/10.1016/j.compeleceng.2022.108156

[5] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," *IEEE Access*, vol. 10, pp. 99837–99849, 2022. doi: 10.1109/ACCESS.2022.3206425.

[6] S. Rizvi, M. Scanlon, J. McGibney, and J. Sheppard, "Deep Learning Based Network Intrusion Detection System for Resource-Constrained Environments," *International Conference on Digital Forensics and Cyber Crime*, vol. 508, S. Goel, P. Gladyshev, A. Nikolay, G. Markowsky, and D. Johnson, eds., Cham: Springer Nature Switzerland, 2023, pp. 355–367. https://doi.org/10.1007/978-3-031-36574-4_21

[7] E. Gyamfi and A. D. Jurcut, "Novel Online Network Intrusion Detection System for Industrial IoT Based on OI-SVDD and AS-ELM," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3827–3839, 2023. DOI: 10.1109/JIOT.2022.3172393

[8] S. Sivamohan and S. S. Sridhar, "An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework," *Neural Comput. Appl.*, vol. 35, no. 15, pp. 11459–11475, 2023. https://doi.org/10.1007/s00521-023-08319-0

[9] H. Asad and I. Gashi, "Dynamical analysis of diversity in rule-based open source network intrusion detection systems," *Empirical Software Eng.*, vol. 27, no. 1, p. 4, 2022. https://doi.org/10.1007/s10664-021-10046-w

[10] R. Chowdhury, S. Sen, A. Roy, and B. Saha, "An optimal feature based network intrusion detection system using bagging ensemble method for real-time traffic analysis," *Multimedia Tools Appl.*, vol. 81, no. 28, pp. 41225–41247, 2022. https://doi.org/10.1007/s11042-022-12330-3

[11] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le,

"Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways," *Sensors*, vol. 22, no. 2, p. 432, 2022. https://doi.org/10.3390/s22020432

[12] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems," *Symmetry*, vol. 14, no. 7, p. 1461, 2022. https://doi.org/10.3390/sym14071461

[13] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-Time Network Intrusion Detection System via Ensemble of Autoencoder in SDN," *IEEE Trans. Netw. Serv. Manage.*, vol. 19, no. 3, pp. 2269–2281, 2022. DOI: 10.1109/TNSM.2022.3175710

[14] C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms," *IEEE/ACM Trans. Networking*, vol. 30, no. 3, pp. 1294–1311, 2022. DOI: 10.1109/TNET.2021.3137084

[15] M.A. Haq, M.A. Rahim Khan and T. AL-Harbi, "Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2022. https://doi.org/10.32604/cmc.2022.018708

[16] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decision Analytics Journal*, vol. 7, p. 100233, 2023. https://doi.org/10.1016/j.dajour.2023.100233

[17] W. Wang, X. Du, D. Shan, R. Qin, and N. Wang, "Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1634–1646, 2022. Doi: 10.1109/TCC.2020.3001017

[18] A.S. Alfoudi, M.R. Aziz, Z.A.A. Alyasseri, A.H. Alsaeedi, R. R. Nuiaa, M.A. Mohammed, K.H. Abdulkareem and M.M. Jaber "Hyper clustering model for dynamic network intrusion detection," *IET Commun.*, p. cmu2.12523, 2022. https://doi.org/10.1049/cmu2.12523

[19] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 235–247, 2023. https://doi.org/10.1007/s10207-022-00634-2

[20] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, 2023. https://doi.org/10.1016/j.comcom.2022.12.010

[21] H. Alazzam, A. Sharieh, and K. E. Sabri, "A lightweight intelligent network intrusion detection system using OCSVM and Pigeon inspired optimizer," *Appl. Intell.*, vol. 52, no. 4, pp. 3527–3544, 2022. https://doi.org/10.1007/s10489-021-02621-x

[22] S. Hosseini and S. R. Sardo, "Network intrusion detection based on deep learning method in internet of thing," *J. Reliab. Intell. Environ.*, vol. 9, no. 2, pp. 147–159, 2023. https://doi.org/10.1007/s40860-021-00169-8

[23] B. Cao, C. Li, Y. Song, and X. Fan, "Network Intrusion Detection Technology Based on Convolutional Neural Network and BiGRU," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–20, 2022. https://doi.org/10.1155/2022/1942847

[24] Dataset link: https://www.kaggle.com/datasets/hassan06/nslkdd