Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

INTRUSION DETECTION SYSTEM USING BINARY PSO AND SUPPORT VECTOR MACHINE ALGORITHMS

G. Stalin Babu[1], T. Sri kanya[2], S. Manaswini[3], K. Swathi[4], P. Pavan kumar[5], R. Manideep[6]

[1, 2, 3, 4, 5] Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh

Abstract

During the past few years, huge amount of network attacks have increased therequirement of efficient network intrusion detection techniques. Different classification techniques for identifying various attacks have been proposed in the literature. In this project we proposed hybrid classifier BPSO (Binary Particle Swarm Optimization)is used to select a subset of features from the NSL KDD dataset, and SVM(Support Vector Machine) is used to classify the network traffic as normal or an attack of one of four categories. BPSO is a metaheuristic algorithm that is commonly used for feature selection in classification problem whereas SVM, a highly accurate classifier, is used here for classification and regression tasks. The BPSO-SVM hybrid classifier aims to improve the classification task. Using the NSL KDD dataset in MATLAB, we show how well our method works. The NSL KDD dataset is a benchmark dataset that is commonly used for evaluating intrusion detection systems. It is a modified version of the KDD Cup 1999 dataset. The NSL KDD dataset contains a large number of features that describe network traffic, as well as labels indicating whether each connection in normal or an attack of one of four categories(Dos,Probe,R2L,U2R).

Keywords: IDS, BPSO, LS-SVM, NSL KDD

1. Introduction

Due to the exploding use of networks in recent years, Internet and computer systems have produced a number of security concerns. According to CERT data (CERT), the number of intrusions has disproportionately grown over time. Any hostile entry or assault on computers, networks, or information systems might result in significant catastrophes and be a violation of the computer security policies, which include Confidentiality, Integrity, and Availability (CIA).

An intrusion detection system (IDS) is a software or hardware-based security tool designed to detect and prevent unauthorized access or malicious activity on a computer network or system. An IDS works by monitoring network traffic and system activity for signs of suspicious behavior, such as unauthorized access attempts, malicious code, or unusual activity patterns.

IDS can be classified into two main categories:

Host-based IDS: This type of IDS is installed on individual computers or servers within a network and monitors their activity for signs of intrusion or compromise. It typically analyzes log files, system calls, and other data generated by the operating system or application to detect suspicious activity.

Network-based IDS: This type of IDS is deployed at strategic points within a network, such as routers, switches, or firewalls, and monitors network traffic for signs of intrusion or compromise. It typically analyzes packets of data passing through the network and looks for anomalies or patterns that indicate suspicious activity.

IDS can operate in two modes: signature-based and anomaly-based detection.

Signature-based detection: This mode compares network traffic or system activity against a database of known attack signatures or patterns. If a match is found, the IDS raises an alert or takes some action to prevent the attack.

Anomaly-based detection: This mode establishes a baseline of normal network or system behavior and looks for deviations or anomalies from that baseline. If an anomaly is detected, the IDS raises an alert or takes some action to investigate the event.

IDS is an important component of network and system security and can help organizations to identify and respond to security incidents in a timely manner, preventing potential damage to their systems and data.

In this paper we propose and implement a hybrid classifier based on binary particle swarm optimization

(BPSO) and Support Vector Machine algorithm for the classification of cyber attacks in a network. Support Vector Machine is an ensemble classification and regression approach.Support Vector Machine or SVM is one of the most popular Supervised algorithms, which is used for Classification as well as Regression problems.An SVM is based on statistical learning theory and classifies data by determining a set of support vectors.The main objective of an SVM is to find an optimal hyperlane for the classification of new data pointsAfter reducing the features we need to classify the records among different attcaks. We further optimize the selected features with the help of PSO algorithm and compare our results with several different classification techniques. We evaluate our proposed approach on NSL KDD dataset.

The rest of the paper is organized as follows: Section 2 contains the literature survey. Section 3 contains dataset information In section 4 the proposed technique is presented along with an overview of the Particle Swarm Optimization and Support Vector Machine algorithms. Section 5 shows experimental setup. The results obtained are shown in section 6.

2. Literature Survey

In this section provides various research works in terms of methodology, year and accuracy.

Table 1: Literature Survey on various Machine learning techniques

REF. NO.	AUTHOR NAME	TITLE	METHODOLOGY	YEAR	ACCURACY
[1]	K. Raj, et al.	Decision Tree Based algorithm for Intrusion Detection	C4.5 Decision Tree Split (DTS)	2016	79.52%
[2]	S.Duque , et al.	Using Data Mining Algorithms for Developing a model for Intrusion Detection System (IDS)	K-means	2015	81.61%
[3]	Ravipati Rama Devi, et al.	Intrusion Detection System Classification Using Different Machine Learning Algorithms On KDD- 99 And NSL-KDD Datasets	Naive Bayes, Adaboost, Multi-Layer Perception	2019	89.5% 89.3% 88.9%
[4]	ManjulC Belavagi ,et al.	Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection	Logistic Regression, Gaussian Naïve Bayes, Support Vector Machine	2016	84% 79% 75%
[5]	Alireza Osareh, et al.	Intrusion Detection in Computer Networks based on Machine Learning Algorithms	Nueral Networks, Support Vector Machine	2008	66.6% 65.7%
[6]	SarthakRastogi, et.al	An Analysis of Intrusion Detection Classification using Supervised Machine Learning Algorithms on NSL-KDD Dataset	Gaussian Naïve Bayes	2022	84.3%

[7]	Jiadong Ren, et al.	Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based onMachine Learning Algorithms	Support Vector Machine, Logistic Regression, Decision Tree	2019	62.5% 83.2% 85.6%
[8]	Alkasassbeh , et al.	Machine Learning Methods for Network Intrusions	Multilayer Perceptron (MLP), Bayes Network	2018	91.9% 90.73%
[9]	Vinoth Y, et al.	Anomaly Based Network Intrusion Detection using Ensemble Machine Learning Technique	DecisionTree Bayes Classifier, RNN-LSTM, Random Forest, Ensemble of the 4 classifiers	2020	85.20%
[10]	Ponthapall i R. et al.	Implementation of Machine Learning Algorithms for Detection of Network Intrusion	Decision Tree, Logistic Regression, Random Forest, SupportVector Machine	2020	72% 68.67% 73.78% 71.77%
[11]	Verma P, et al.	Network IntrusionDetection usingClustering and Gradient boosting	Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost)	2018	84.253% 80.238% 82.011% 80.731%

[12]	Marzia Z, et al.	Evaluation of Machine Learning Techniques for Network Intrusion Detection	Fuzzy C-Means, K-means	2018	83.60% 83.60%

3. DATASET

The NSL-KDD dataset is a widely used benchmark dataset for evaluating intrusion detection systems (IDS). It was created as an extension to the original KDD Cup 1999 dataset, which had several limitations and did not reflect the current state of network attacks. The NSL-KDD dataset overcomes these limitations by introducing a new set of attack scenarios and providing a more comprehensive and diverse set of network traffic data.

The NSL-KDD dataset contains a total of 125,973 labeled instances of network traffic, which are classified into four main categories of attacks and normal traffic. The four attack categories are:

A .Denial of Service (DoS) attacks: These attacks involve overwhelming a target system with traffic or requests, causing it to become unresponsive.

B. Probe attacks: These attacks involve attempting to gather information about a target system, such as its vulnerabilities and weaknesses.

C.User to Root (U2R) attacks: These attacks involve exploiting a vulnerability in a system to gain privileged access and control.

D.Remote to Local (R2L) attacks: These attacks involve gaining unauthorized access to a system from a remote machine.

4. Methodology

In our proposed technique we have implemented a binary PSO algorithm for feature selection and Least Square SVM for classification. In this section, an introduction to particle swarm optimization and Least Square SVM is given and then proposed technique is elaborated.

4.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a nature-inspired optimization algorithm that was first proposed by Eberhart and Kennedy in 1995. It is a population-based optimization technique that is used to find the optimal solution to a problem by iteratively adjusting the positions of a group of particles in a search space. In PSO, a swarm of particles moves through a search space, where each particle represents a potential solution to the problem. The movement of the particles is guided by their own experience (i.e., their personal best) and the experience of the swarm (i.e., the global best). Each particle updates its position and velocity based on its personal best and the global best, and this process is repeated until a satisfactory solution is found. PSO has been successfully applied in various fields, such as engineering, finance, and biology, to solve problems such as function optimization, classification, and clustering. Its simplicity, effectiveness, and flexibility make it a popular choice among researchers and practitioners in optimization and machine learning.

Role of BPSO

The goal of BPSO is to find the optimal binary string that maximizes or minimizes a given objective function. The particles move through the search space by updating their positions and velocities at each iteration based on their own position, the best position found by the particle swarm so far, and the best position found by the particle itself.

In BPSO, the velocity of each particle is a binary vector that determines the direction and magnitude of its movement in the search space. The velocity is updated at each iteration based on the position of the particle and the best position found by the swarm so far.

Pseudocode of the BPSO algorithm

```
1.  Initialize particles' positions and velocities randomly within the search space
2.  Initialize personal best positions and objective function values for each particle
3.  Initialize global best position and objective function value
4.  Set maximum number of iterations
5.  For each iteration do:
6.  For each particle do:
7.  Evaluate the objective function for the current position
8.  If the objective function value is better than the personal best:
9.  Update the personal best position and objective function value
10. If the objective function value is better than the global best:
11. Update the global best position and objective function value
12. Update the velocity of the particle using the current position, personal best position, and global best position
13. Update the position of the particle using the new velocity
14. End For
15. End
```

The key advantage of BPSO is that it can handle binary optimization problems, which are common in many applications such as genetic algorithms, logic circuits, and digital signal processing, among others. BPSO can efficiently search the binary search space and find the optimal solution, even in cases where the search space is high-dimensional.

BPSO has been applied in various fields such as feature selection, data clustering, and image processing, among others. Its simplicity, fast convergence, and robustness to noise and outliers make it a popular choice for solving binary optimization problems.

Support Vector Machine

In SVM, the objective is to find the best separating hyperplane that can classify the data points into two or more classes with maximum margin. The margin is defined as the distance between the decision boundary and the closest data points from each class. To find the best separating hyperplane, SVM uses a mathematical optimization technique that involves finding the hyperplane that maximizes the margin while minimizing the classification error. This hyperplane is also known as the maximum margin hyperplane (MMH). In cases where the data is not linearly separable, SVM uses a technique called kernel trick to transform the data into a higher dimensional space where it becomes linearly separable. This helps to improve the classification accuracy of SVM. SVM is widely used in many applications such as text classification, image classification, bioinformatics, and data mining, among others. Its popularity is due to its ability to handle high-dimensional data and its robustness to noise and outliers.

Least Square SVM

Least-squares support-vector machines (LS-SVM) for statistics and in statistical modeling, are least squares versions of support-vector machines (SVM), which are a set of related supervised learning methods that analyze data and recognize patterns, and which are used for classification and regression analysis. In this version one finds the solution by solving a set of linear equations instead of a convex quadratic programming (QP) problem for classical SVMs. Least-squares SVM classifiers were proposed by Johan Suykens and Joos Vandewalle. LS-SVMs are a class of kernel-based learning methods.LS-SVM is used for regression and classification problems, just like SVM. However, LS-SVM uses a different approach to find the optimal separating hyperplane. In traditional SVM, the optimization problem involves finding the hyperplane that maximizes the margin between the classes while minimizing the misclassification error. On the other hand, LS-SVM aims to find a linear or nonlinear function that maps the input data to the output variable using a least squares approach. In LS-SVM, the optimization problem is formulated as a linear system of equations, which can be solved using matrix algebra. This approach is computationally efficient and allows for faster training of the model, especially in cases where the number of training examples is large. LS-SVM also has some advantages over traditional SVM. For instance, LS-SVM does not require the selection of the regularization parameter, which can be a challenging task in traditional SVM. Additionally, LS-SVM can handle noisy data and outliers better than traditional SVM. LS-SVM has been successfully applied in many applications, including system identification, time series prediction, and image classification, among others.

The process of the proposed technique involves following steps:

- Data Preprocessing
- Feature Selection
- Classification

Data Preprocessing

The NSL-KDD dataset contains a mix of categorical and continuous features. The first four columns represent continuous features, while columns 5 to 41 are categorical features. The categorical features in columns 5 to 41 are represented as integers, with each integer corresponding to a specific category or value. In this paper preprocessing steps include normalization and shuffling of the data.

Each feature is normalized to have a minimum value of 0 and a maximum value of 1 using the formula: Y = (X - X_min) / (X_max - X_min) where X is the original feature value, X_min is the minimum value of that feature across all samples, X_max is the maximum value of that feature across all samples, and Y is the normalized feature value. Then, the normalized feature values are scaled to a range between 0.1 and 0.99 using the formula:

$$\text{scaled\_value} = Y * (b - a) + a$$

where $a = 0.1$ and $b = 0.99$.

The target variable is set to 1 if the 42nd column of the table is 'normal' and -1 otherwise. Then the target is transposed so that it is a column vector.As 2nd, 3rd, 4th columns are continuous numerical values, numerical data is extracted from these columns. Then all the columns are concatenated including first column and 5 to 41 columns.

An equal number of normal and attack instances are splitted to avoid any bias towards one class during the training of the model. After balancing the dataset, the attacks and normal instancres are merged together to create a single dataset for further processing. This merged dataset is then shuffled randomly to avoid any biases and to ensure that each sample has an equal chance of being selected during training or testing.

Feature selection and Classification

In this paper, three cases of implementation is done on the NSL-KDD Dataset.

Case 1: Least Squares SVM performed on all the features

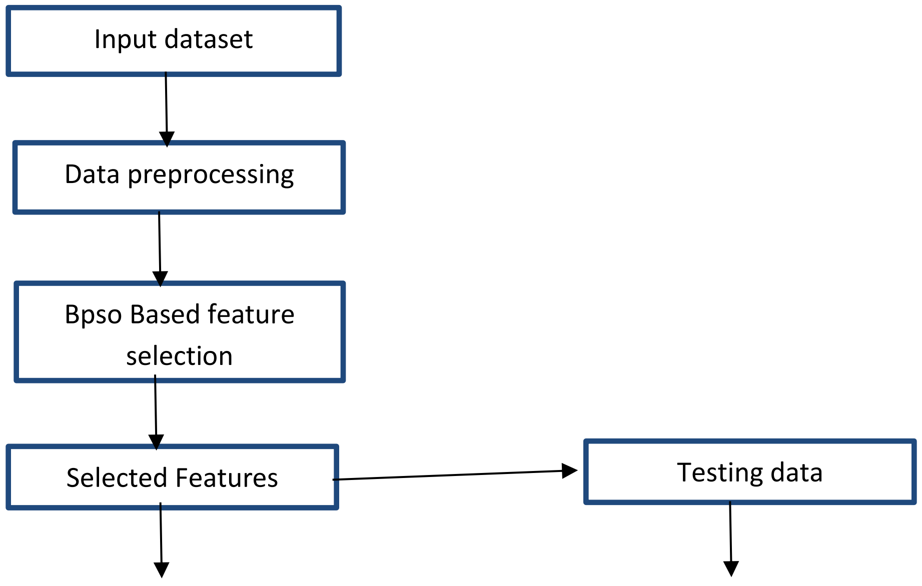
In this case, Least Square SVM is performed on all the features of training data and testing data for training and testing respectively. Initially data train and data test files are loaded and samples are taken from the dataset for training and testing which are 500 and 100 respectively.

Case 2: Least Squares SVM performed on 25 features

In this case, Least Squares SVM is performed on only 25 features of training data and testing data. The 25 features on which Least Squares SVM .In this case also data train and test files are loaded and samples are taken from the dataset for training and testing which are 500 and 100 respectively.

Case 3: Binary PSO and LS-SVM

Binary Particle Swarm Optimization (BPSO) to select a subset of features that are most relevant to the classification problem at hand. The selected features are then used to train a Least Squares Support Vector Machine (LS-SVM) for classification.



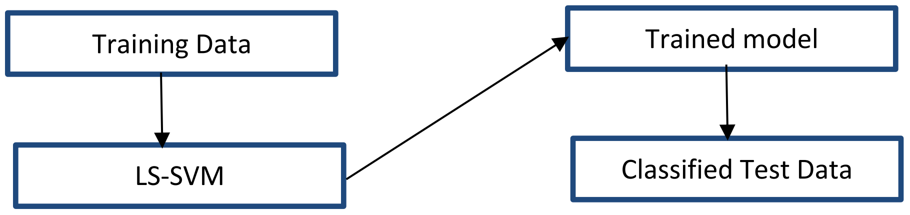


Fig 1: Frame work of Intrusion Detection System

6. Results

The performance of our proposed model was evaluated using various performance metrics, including Accuracy, AUC, F1-Score, Specificity, and Sensitivity. These metrics provide a comprehensive evaluation of the model's performance in terms of different aspects such as the model's ability to correctly classify positive and negative samples, the trade-off between sensitivity and specificity, and the overall prediction accuracy.

Table 2- Performance matrices

Algorithms	**Accuracy**	**AUC**	**F1 Score**	**Specificity**	**Sensitivity**
SVM	85%	97%	84.54%	95.65%	75.93%
SVM(25_Char)	89%	98.1%	89.11%	95.65%	83.33%
SVM-BPSO	93%	97.7%	93.33%	95.65%	90.7%

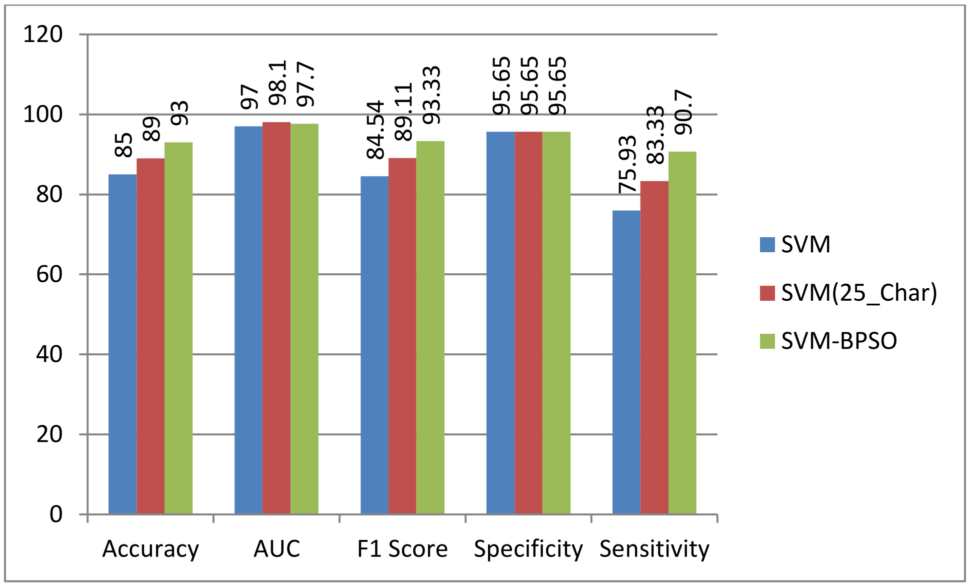


Figure 2: Comparing Performance Metrics for 3 Models

When all the features from the dataset were used, LS-SVM achieved an accuracy of 85% and an AUC-RUC Curve of 97%. This suggests that the model was able to correctly classify a large proportion of the samples, but there is still room for improvement in terms of its discriminatory power.

In contrast, when only 25 features from the dataset were used, LS-SVM achieved an accuracy of 89% and an AUC-RUC Curve of 98.1%. This indicates that the subset of features used in the model was more informative and contributed more to the model's discriminatory power.

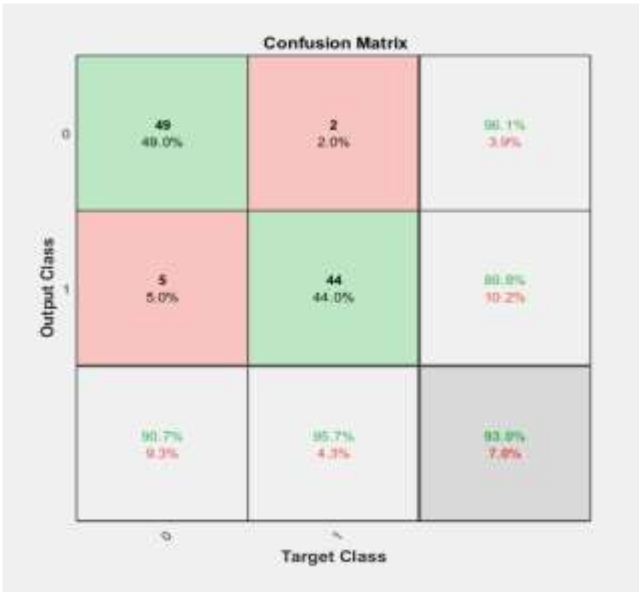


Figure 3: Confusion Matrix for BPSO and LS-SVM Model

To further improve the model's performance, a subset of features was selected using BPSO, and a model was created using LS-SVM. This approach resulted in an accuracy of 93% with an AUC-ROC curve of 97.7%.

Figure 2 shows the confusion matrix for this model. It is labeled with output class and target class. The resulting model achieved an accuracy good accuracy with an AUC-ROC curve of 97.7%, indicating that the model was able to accurately distinguish between positive and negative samples and achieved a high level of discriminatory power.

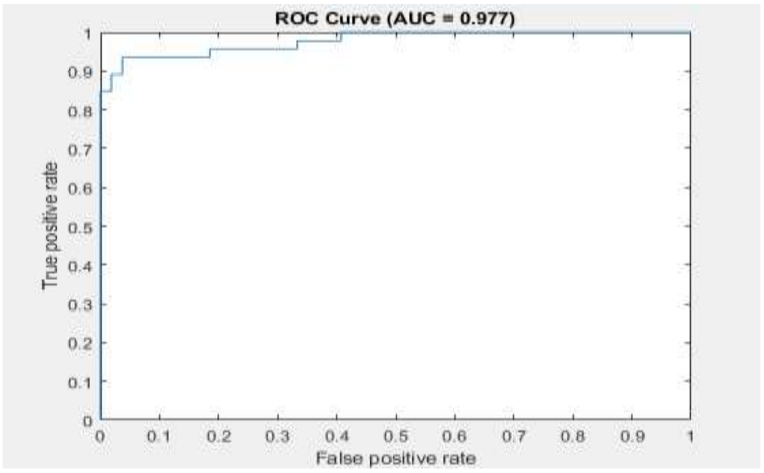


Figure 4: AUC-ROC Curve for BPSO and LS-SVM

These results demonstrate the effectiveness of feature selection methods in improving the performance of machine learning models by reducing the dimensionality of the dataset and removing irrelevant or redundant features.

5. Conclusion and Future Work

The achieved accuracy of 93% with an AUC-ROC curve of 97.7% suggests that the proposed model is highly effective in predicting the classes of new, unseen samples with a high degree of confidence. AUC-ROC curve can be seen Figure 3. The high AUC score indicates that the model is well-suited for imbalanced datasets, where the number of samples in one class is significantly higher than the other. In such cases, AUC is a better metric than accuracy as it considers the model's ability to correctly rank the samples rather than just its overall accuracy. The effectiveness of BPSO for feature selection is due to its ability to explore the search space efficiently and effectively. BPSO can identify the most informative features that contribute significantly to the model's discriminatory power while removing redundant and irrelevant features that may cause overfitting or increase the model's complexity. LS-SVM, on the other hand, is a powerful classification algorithm that can handle non-linear and high-dimensional datasets. The use of LS-SVM in our proposed model helped in reducing overfitting and improving the generalization performance of the model. This is because LS-SVM has a regularization term that penalizes the model for having too many non-zero coefficients, resulting in a more parsimonious model that can generalize better to new samples. Overall, the achieved results demonstrate the effectiveness of using advanced techniques like BPSO for feature selection and LS-SVM for classification in improving the performance of machine learning models. By combining these techniques, we were able to build a model that achieved a high level of discriminatory power and accurately predicted the classes of new, unseen samples with a high degree of confidence.

In our model, we have used some of the samples from the both training and testing datasets. In future, we would like to test on more samples and also test for the each attack individually like DoS, Probe, etc.

References

[1] Rai, Kajal, M. Syamala Devi, and Ajay Guleria. "Decision tree based algorithm for intrusion detection." *International Journal of Advanced Networking and Applications* 7.4 (2016): 2828.

[2] Duque, Solane, and Mohd Nizam bin Omar. "Using data mining algorithms for developing a model for intrusion detection system (IDS)." *Procedia Computer Science* 61 (2015): 46-51.

[3] Ravipati, Rama Devi, and Munther Abualkibash. "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper." *International Journal of Computer Science & Information Technology (IJCSIT) Vol* 11 (2019).

[4] Belavagi, Manjula C., and Balachandra Muniyal. "Performance evaluation of supervised machine learning algorithms for intrusion detection." *Procedia Computer Science* 89 (2016): 117-123.

[5] Osareh, Alireza, and Bita Shadgar. "Intrusion detection in computer networks based on machine learning algorithms." *International Journal of Computer Science and Network Security* 8.11 (2008): 15-23.

[6] Rastogi, Sarthak, et al. "An analysis of intrusion detection classification using supervised machine learning algorithms on NSL-KDD dataset." *Journal of Computing Research and Innovation (JCRINN)* 7.1 (2022): 124-137.

[7] Ren, Jiadong, et al. "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms." *Security and communication networks* 2019 (2019).

[8] Alkasassbeh, Mouhammad, and Mohammad Almseidin. "Machine learning methods for network intrusion detection." *arXiv preprint arXiv:1809.02610* (2018).

[9] Vinoth, Y., and K. Kamatchi. "Anomaly Based Network Intrusion Detection using Ensemble Machine Learning Technique." *International Journal of Research in Engineering, Science and Management. IJRESM* (2020): 290-296.

[10] Raviteja, Ponthapalli, et al. "Implementation of machine learning algorithms for detection of network intrusion." *International Journal of Computer Science Trends and Technology (IJCST).(163-169)* (2020).

[11] Verma, Parag, et al. "Network intrusion detection using clustering and gradient boosting." *2018 9th International conference on computing, communication and networking technologies (ICCCNT)*. IEEE, 2018.

[12] Zaman, Marzia, and Chung-Horng Lung. "Evaluation of machine learning techniques for network intrusion detection." *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018.