Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

NIDS-ML-PSO: Network Intrusion Detection System based on Machine Learning Classifiers and Particle Swarm Optimization

Waseem Ghazi Mahdi[a,*], ***Dalal Abdulmohsin Hammood***[a], ***Leith Hamid Abed***[b], ***and Shahad Ali Sameer***[c]

[a] *Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq. Email: bbc4030@mtu.edu.iq, dalal.hammood@mtu.edu.iq*

[b] *Technical Institute of Anbar, Department of Computer Systems, Middle Technical University, Baghdad, Iraq. Email: laithhamed@mtu.edu.iq*

[c] *College of Computer Science and Information Technology . Al-Qadisah University , Diwaniyah-Iraq .Email: Shahadalisameer@gmail.com*

ARTICLEINFO

Article history:
Received: 09 /11/2024
Rrevised form: 23 /11/2024
Accepted : 17 /12/2024
Available online: 30 /12/2024

Keywords: Network intrusion detection systems (NIDs), Machine learning classifier, Feature selection, Network security, Particle Swarm Optimization (PSO).

ABSTRACT

As cyber threats continue to escalate with the rapid growth of internet usage, robust intrusion detection systems (IDSs) are essential for safeguarding network infrastructures. This study proposes an enhanced intrusion detection approach using the NSL-KDD dataset, where particle swarm optimization (PSO) is employed for feature selection to optimize machine learning classifier performance. PSO effectively reduces data dimensionality by identifying the most relevant features, improving computational efficiency and detection accuracy. Four machine learning classifiers, such as support vector machine (SVM), decision tree (DT), extra trees (ET), and random forest (RF), are evaluated with and without PSO to assess its impact. Experimental results demonstrate that PSO-based feature selection significantly improves performance, with RF achieving the highest accuracy of 98.33%. Comparative analysis with recent studies highlights the competitive advantage of the proposed method. The study concludes by identifying limitations and proposing future work, including exploring alternative feature selection techniques such as Genetic Algorithm (GA), Bat Algorithm (BA), and Cuckoo Search (CS) to further enhance IDS effectiveness.

MSC.

https://doi.org/10.29304/jqcsm.2024.16.41787

1.Introduction

The rapid growth of the internet has led to an alarming increase in cyberattacks targeting networks and computer systems, posing significant challenges to cybersecurity. To safeguard network infrastructures, Intrusion Detection Systems (IDSs) have become indispensable tools for detecting and preventing hostile activities. Ensuring that software frameworks and network environments remain secure and reliable is one of the most pressing challenges in today's technological landscape [1]. Intrusion detection involves identifying and responding to activities that exploit computing and network resources. However, increasing network traffic and the evolving sophistication of cyber threats have made it difficult for IDSs to accurately detect malicious activities. Traditional IDS methods often struggle to keep up with emerging attacks, leading to inefficiencies in detection and high false alarm rates. To

∗Corresponding author Waseem Ghazi Mahdi

Email addresses: bbc4030@mtu.edu.iq

Communicated by 'sub etitor'

address these challenges, machine learning (ML)-based IDS models have emerged as promising solutions. These models offer high detection accuracy through the ability to learn patterns from historical data, computational efficiency through optimized feature selection, and adaptability to emerging and evolving threats [2]. This paper provides a comprehensive review of current advancements in machine learning-based intrusion detection systems. It explores lightweight ML models, which have been shown to efficiently address detection accuracy, feature redundancy, and computational overhead in IDS. By laying a state-of-the-art foundation, this work aims to assist future researchers in further improving IDS methodologies and addressing ongoing cybersecurity challenges [3].

Machine learning-based IDSs provide an approach that is driven by learning to identify attack types through modeled behavior of normal and malicious activities. Most of these systems rely on a supervised learning method for effective and comprehensive representations of known threats, though periodic maintenance of the signature database is still required and adds to the workload of users [4]. Development in the field of intrusion detection techniques is achieved by improving machine learning models, which introduce a different degree of accuracy and efficiency across different systems. All in all, this is a very good trend in IDS technology because the difference in technological aspects is so vast: machine-learning-based robust and resilient systems against evolving threats. Advanced hardware architectures further allowed more integrated models that were complex and deployed, hence boosting the ability of the system in anomaly detection across datasets for proper identification of security threats.

Basically, striking the effectiveness of detection with the efficiency of resources is the key driver of using lightweight machine learning. Multilayer Perceptron (MLP), Support Vector Machines (SVM), Logistic Regression (LR), K-nearest neighbor (KNN), and Random Forest (RF) are major models that would assure high detection accuracy with least computational overhead and, as such, would be ideal for intrusion detection in real time. Another fast-growing field is machine learning for IoT security. So far, the combination of machine learning-based methods with big data analytics has led to certain successful performances in cybersecurity applications. For this reason, machine learning algorithms are under the spotlight due to their outstanding pattern-extraction capability, often superior to other approaches in several studies [5,6]. The following are the research's primary contributions:

- This study applies Particle Swarm Optimization (PSO) for feature selection on the NSL-KDD dataset, demonstrating significant enhancements in classifier accuracy and robustness. PSO effectively minimizes dataset dimensionality while retaining critical features, improving the performance and efficiency of intrusion detection systems.

- A comprehensive analysis was conducted on classifiers including Support Vector Machine (SVM), Decision Tree (DT), Extra Trees (ET), and Random Forest (RF), comparing performance with and without PSO-based feature selection. The findings highlight the advantage of using PSO for feature selection, particularly for ET and RF, which achieved high accuracy levels of 97.38% and 98.33%, respectively.

- This work includes a detailed comparison with recent studies using the NSL-KDD dataset, showcasing the effectiveness of PSO-based feature selection. The analysis demonstrates that the proposed approach sets a new standard for accuracy and reliability in detecting intrusions, providing a competitive edge over existing methods in the field.

2. Related Works

In recent years, various machine learning and ensemble approaches have been researched in order to provide an effective intrusion detection performance and overcome issues such as class imbalance and dimensionality in network traffic datasets. For instance, Maryam Samadi Bonab et al. [7] performed feature selection on NSL-KDD dataset. Their work employed the ant lion, fruit fly optimization algorithm, and a hybrid that combined the two methods in feature selection. Some of the classification algorithms used with the selected features are the KNN, NB, SVM, and DT. The best results were for the k-nearest neighbor classifier, yielding the highest accuracy of 89.0% on the NSL-KDD dataset. However, feature selection improved performance, the final accuracy remained limited. Due to the reliance on KNN, which is sensitive to feature space and data size, its scalability has been limited. Thavavel Vaiyapuri and Adel Binbusayyis [8] have applied five varieties of auto-encoders to network traffic anomaly detection in the NSL-KDD datasets. Following an unsupervised deep learning approach, they employed the strength of autoencoders in learning the representation of data in a compressed format. The best accuracy that they obtained on the NSL-KDD dataset was 87.9% using a contractive autoencoder. Feature selection methods other than those proposed further enhance performance by filtering less relevant features before feeding them into the model of the deep learning. However, the reported accuracy still shows difficulties in handling feature redundancy and rare attack classes.

Byeongjun Min et al. [9] They proposed using a memory-augmented deep autoencoder to solve the overgeneralization problem that often occurs when using an autoencoder. Their solution achieved an accuracy of 89.5% on the NSL-KDD dataset. The paper applied the capability of only a deep learning model for feature compression without using any statistical/ machine learning methods for reducing the number of input features in advance. However, their approach used DL-based feature compression; no statistical or machine learning feature reduction methods were employed to ensure interpretability and efficiency. The experiments were carried out by AK Pandey et al. [10] on UNSWNB-15 and NSL-KDD datasets. GAN was applied to generate synthetic records before the classification in order to handle class imbalance. Then, a multiclass SVM classifier was conducted based on a fine-tuning of parameters performed by Bayesian hyperparameter optimization in their proposed approach. The achieved classification accuracy using this approach is equal to 99.5% over the NSL-KDD dataset. This far exceeded our results derived from the NSL-KDD dataset and may be related to the effective handling of class imbalance within this dataset. However, the results are impressive, their dependency on GAN-generated data begs the question of how well this model generalizes to more real-life datasets with different distributions.

Kumar and Sinha [11] introduced a model combining a XGBoost classifier with Wasserstein Conditional GAN to generate synthetic IDS datasets, incorporating feature reduction through a deep autoencoder. This model achieved an F1-score of 0.96 when evaluated on the NSL-KDD dataset. However, the WCGAN’s testing for effectiveness on machine learning classifiers involved both real and synthetic data rather than solely synthetic samples. Additionally, there was inconsistency in feature alignment across datasets, and the high detection rate was mainly observed with the NSL-KDD dataset. However, the method was very dependent on synthetically generated data, and any inconsistency in the feature alignment across these datasets would defeat the purpose of their model's robustness. The works in [12] have used linear discriminant analysis, KNN certainty factor voting classifiers, and two-class machine learning classification models for feature dimensionality reduction. In handling class imbalance for anomaly datasets, the SMOTE technique is applied. This model, when trained with two newly created training datasets, showed an accuracy of 83.24% on the NSL-KDD dataset for 16 selected features. However, though these are two promising works using LDA for feature selection and SMOTE for class imbalance, the achieved accuracy is somewhat low.

A model for intrusion detection in cloud environments, including deep and machine learning, was proposed by Mourade Azrour and Abdulatif Alabdultif [13]. They used the RF classifier for filtering the features and then used RBFNN for intrusion detection. The best features were selected by the RF classifier, where the detection of intrusion in cloud computing was done by the RBFNN algorithm. Their approach, on NSL-KDD dataset, resulted in an accuracy of 94.16%, which was efficient for intrusion detection. Besides, feature selection techniques were proved very useful and important in enhancing general efficiency in the IDS. However, the reliance on RF for feature selection overlooks subtle correlations in features, and RBFNN models tend to be computationally costly. In [14] is proposed an ensemble-based machine learning approach for intrusion detection, incorporating a part of the techniques of Simple Stacking, Adaboost, RF, XGBoost, Bagging and Gradient Boosting. Their approach employed mutual information, PCA, and correlation analysis feature selections and reported accuracy above 99% on various publicly available datasets. Their method enhanced the security further against the cyber-attacks arising day by day. However, the computational cost of ensemble methods may also limit their applicability in real-time intrusion detection scenarios.

Ayuba John et al. [15] introduced a variable ensemble machine learning approach for intrusion detection, combining PCA with AdaBoost, LogitBoost, and RandomForest to address low accuracy and high false alarms in IDS models. For the NSL-KDD dataset, the AdaBoost with PCA approach achieved an accuracy of 89.0%, PCA with RandomForest reached 100% accuracy, and PCA with LogitBoost also attained a perfect accuracy of 100%, demonstrating the effectiveness of these combinations in improving detection performance. However, the generalizability of their approach to more complex or imbalanced datasets remains unaddressed, which is critical for real-world applicability. Yu Yang et al. [16] developed a method to enhance rare-class attack detection by using an optimized kernel density estimation with geometric synthetic minority oversampling technique. Balanced data was processed through denoising autoencoders to reduce redundancy and improve detection accuracy. A soft-voting ensemble method was then applied for multi-class anomaly detection, resulting in 86.39% accuracy on NSL-KDD dataset. The proposed strategy achieved improved performance in identifying unknown and rare attack types. The reviewed studies demonstrate that advanced machine learning methods, particularly ensemble and feature selection techniques, significantly improve intrusion detection performance, especially for rare and complex attack types. However, their approach improved rare-class detection, but the overall accuracy remained low due to redundancy and noise in the dataset, which could have been addressed through more effective feature selection.

3. Methodology

The methodology adopted in this study is designed to enhance the performance of intrusion detection systems using the NSL-KDD dataset. The approach is presented as a structured pipeline in Fig. 1, encompassing data preprocessing, feature selection, model training, and evaluation. The ultimate goal is to achieve a robust classification of network traffic into normal and abnormal categories, leveraging machine learning models optimized through effective feature selection. The pipeline begins with data preprocessing, where Min-Max normalization is applied to the raw dataset. This step standardizes the feature values, ensuring they fall within a specific range, which aids in improving the convergence and performance of machine learning models. Next, the preprocessed data undergoes feature selection using PSO. PSO identifies an optimal subset of features that balances model accuracy and computational efficiency by removing redundant or irrelevant features. The selected feature subset is then divided into training and testing datasets, with 80% allocated for training and 20% reserved for testing. This ensures that the models are trained effectively while maintaining a separate dataset for unbiased evaluation. Following the data preparation, multiple machine learning models, including SVM, ET, DT, and RF, are applied to the preprocessed and optimized dataset. Each model is trained and fine-tuned to maximize its predictive performance. Finally, the models are rigorously evaluated using standard performance metrics: F1-score, accuracy, precision, recall, and a confusion matrix. These metrics provide a detailed assessment of the models' effectiveness in detecting normal and abnormal network traffic, offering insights into their strengths and limitations.

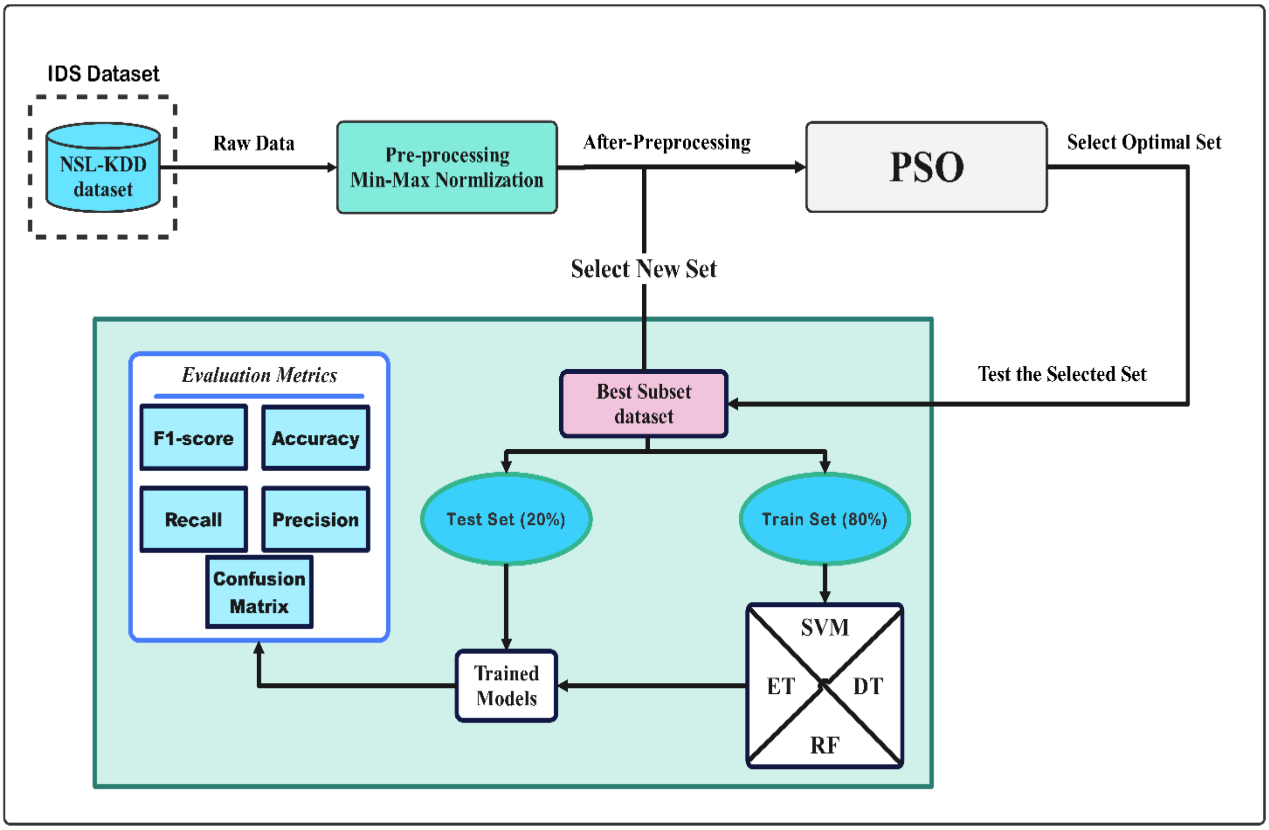

Fig. 1- Proposed IDS approach based on classifiers and PSO.

3.1. NSL-KDD Dataset Description

The selection of appropriate datasets is due to the fact that intrusion detection model evaluation requires appropriate benchmarking. This is why we employed the NSL-KDD dataset, an improved version of KDDCup'99, overcoming difficulties like redundancy and biased distribution of classes, features giving wrong indications of the model performance [17]. NSL-KDD provides 41 characteristics in basic, content, time-based, and host-based, with each record labeled as normal or one of the four attack categories: Probe, U2R, R2L, and DoS. It was divided into the NSL-KDD-Train for training and the NSL-KDD-Test for testing, thus giving a very structured basis on which model performance could be measured. Although NSL-KDD is not capable of representing the modern nature of network traffic, it is one of the most utilized benchmarks for evaluating intrusion detection techniques. The details in the dataset are represented in Table 1.

Table 1 - Description of NSL-KDD dataset.

Type of Traffic	Class	NSL-KDD-Test	NSL-KDD-Train
Normal	Normal	9711	67,343
Abnormal	Probe	2421	11,656
	R2L	2885	995
	U2R	67	52
	DOS	7460	45,927
Total records	5 classes	22,544	125,973

3.2. Preprocessing NSL-KDD Dataset

Preprocessing in this regard is the most important step to prepare the NSL-KDD dataset for training, as it greatly improves the accuracy and performance of machine and deep learning models when combined [18]. Preprocessing indeed is helpful to remove noise, balance data, and scale features consistently-all elements that will enhance good generalization and reduce overfitting of a model. Preprocessing also helps the models learn the patterns better by standardizing the data, which means better detection rates for both frequent and infrequent attack types. This is an essential phase in intrusion detection because of the high dimensions and diversity in data types that can degrade model performance if not treated.

One of the common practical preprocessing techniques is called Min-Max Normalization, in which each feature should scale in a certain range, usually within the range of [0, 1]. This kind of normalization is very helpful for any model of machine learning or deep learning because it avoids dominant features while normalizing all features on a comparable scale. Min-Max Normalization is given by the following transformation formula:

$$\boldsymbol{Z_{norm}} = \frac{y - y_{min}}{y_{max} - y_{min}} \tag{1}$$

Where $\boldsymbol{y}$ represents the original feature value, $\boldsymbol{y_{max}}$ and $\boldsymbol{y_{min}}$ are the minimum and maximum values of the feature, respectively, and $\boldsymbol{Z_{norm}}$ is the normalized value. It normalizes all the feature values in a uniform range so that models converge better during the training process, which enhances the efficiency in intrusion detection.

3.3. Feature selection on the NSL-KDD Dataset based on PSO

Feature selection is one of the most important steps for fine-tuning both machine and deep learning models. For example, in cases of high-dimensional datasets like the NSL-KDD, feature selection keeps only the relevant features and reduces the noise of the data [19,20]. This reduces computational complexity because now the models will focus on the most informative attributes. It impacts directly the accuracy and efficiency of the model by reducing overfitting and speeding up the convergence of training. That is to say, intrusion detection systems with feature selection will enhance their detection capability by focusing on patterns that are very important in the given data.

Feature selection is done on the NSL-KDD dataset using PSO in this paper. PSO is a population-based optimization algorithm inspired by the collective behavior of swarms, for instance, the flocking of birds or schooling of fish. A particle in PSO corresponds to one candidate solution-a subset of the features in this case-and navigates through the search space in search of an optimal solution for the predefined objective, normally maximizing the accuracy of the model [21]. Each particle updates its position based on two important components: a personal best position and a global best position in the swarm. The position and the velocity of each particle are updated in every iteration using the following formulae:

$$\boldsymbol{v_i^{(t+1)}} = w\, v_i^{(t)} + c_1\, r_1 (p_i^{best} - x_i^{(t)}) + c_2\, r_2\, (g^{best} - x_i^{(t)}) \tag{2}$$

$$\boldsymbol{x_i^{(t+1)}} = \boldsymbol{x_i^{(t)}} + \boldsymbol{v_i^{(t+1)}} \tag{3}$$

Where $\boldsymbol{v}_{\boldsymbol{i}}^{(t)}$ and $\boldsymbol{x}_{\boldsymbol{i}}^{(t)}$ represent the velocity and position of particle $\boldsymbol{i}$ at iteration $\boldsymbol{t}$, respectively. Parameters $\boldsymbol{w}$, $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ influence the particle's movement, while random values $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ add variability. By applying PSO to identify a subset of critical features, we achieve improved model accuracy and efficiency, enabling more focused and effective intrusion detection on the NSL-KDD dataset.

3.4. Assessment Criteria

In this respect, intrusion detection models' performance will be compared in our study concerning some metrics that will assess their performance of correctly classifying normal network traffic as a positive class and abnormal network traffic as a negative class, which will give an indication of the integral performance of the model in identifying benign network activities from malicious ones [22].

The confusion matrix will be very useful in this evaluation, describing the results in terms of four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [23,24]. Here, TP means that the normal traffic has been identified, TN indicates the abnormal traffic that was correctly classified; FP refers to the abnormal traffic misclassified as normal, and FN is the normal traffic which has been misclassified as abnormal.

- Accuracy gives the proportion of all correctly classified instances (both normal and abnormal) out of the total predictions, providing an overall measure of the model's effectiveness [25]:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{True\ Positive} + \boldsymbol{True\ Negative}}{False\ Negative + True\ Positive + False\ Positive + Ture\ Negative} \tag{4}$$

- Recall (Sensitivity, for normal traffic) assesses the model's ability to detect all actual normal instances, showing the proportion of correctly identified normal traffic out of all actual normal cases [26]:

$$\boldsymbol{Recall} = \frac{True\ Positive}{False\ Negative + True\ Positive} \tag{5}$$

- Precision (for normal traffic) calculates the proportion of correctly classified normal instances out of all instances predicted as normal [27]:

$$\boldsymbol{Precision} = \frac{True\ Positive}{False\ Positive + True\ Positive} \tag{6}$$

- F1-Score balances Precision and Recall, providing an overall measure that is useful for evaluating model performance when there is a need to consider both correct detections and false classifications [28]:

$$\boldsymbol{F1 - Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

This set of metrics will, therefore, enable the estimation of the strength of the model in differentiating between normal and abnormal traffic, further into how it will behave in real network scenarios. It therefore follows that abnormal and normal traffic classification provides some reasonable balance in analysis.

4. Results and Discussion

Experiments were conducted on an Intel-based Windows system based on the specifications shown in Table 2. For the programming part, Python was used, with most of the code being written and executed in the Visual Studio Code.

This section presents the performance evaluation of the considered classifiers in this study, such as SVM, DT, ET, and RF, with and without PSO for feature selection. There are five important metrics that have been employed in this study for the performance evaluation of each of these classifiers, including confusion matrix, recall, precision, F1-score, and accuracy. These experiments have been carried out in two different scenarios: first, by applying PSO for feature selection, and second, without using PSO-to clearly observe the performance differences of the classifiers with and without the optimized subset of features.

Table 2 - System Configuration.

Specification	Description
Processor	Intel Core i7, 6th Generation
Graphics	8 GB VRAM
Memory Capacity	16 GB RAM
Programming Language	Python
OS Environment	Windows 10 Home

These reflect respective performances for each classifier with regard to the detection of normal and abnormal traffic. The results confirm that through the application of PSO for feature selection, improvement is given to the models by letting the classifier focus on the most relevant features, hence increasing its accuracies and efficiency with regard to different metrics under consideration.

4.1. Evaluation of machine learning classifier without PSO

This subsection evaluates the performance of various ML classifiers on the NSL-KDD dataset without applying PSO for feature selection. The classifiers assessed include SVM, DT, ET, and RF. Table 3 summarizes the results based on key metrics: Precision, Accuracy, Recall, and F1-Score. Without PSO, all features are included in training, potentially impacting both computational efficiency and model accuracy. As shown, RF achieved the highest accuracy at 92.04% and an F1-Score of 92.27, demonstrating robustness even without feature optimization. ET also performed well, with an accuracy of 90.48% and an F1-Score of 91.05. In comparison, DT and SVM exhibited slightly lower accuracy and F1-Scores, suggesting some sensitivity to feature redundancy.

Table 3 - Classifiers performance without using PSO.

Model	Precision	Accuracy	Recall	F1-Score
SVM	95.56	85.24	78.98	86.48
DT	96.27	87.81	82.14	88.65
ET	98.05	90.48	84.99	91.05
RF	96.18	92.04	88.67	92.27

The confusion matrices for each classifier (SVM, DT, ET, and RF) are shown in Fig. 2, providing a detailed view of their performance in classifying normal and abnormal traffic within the NSL-KDD dataset without PSO for feature selection. Each matrix includes four key outcomes: TP (normal traffic correctly classified as normal), TN (abnormal traffic correctly classified as abnormal), FP (normal traffic misclassified as abnormal), and FN (abnormal traffic misclassified as normal). Analyzing these matrices highlights each classifier's strengths and limitations, particularly in accurately identifying abnormal instances, which is crucial for effective IDS.

In Fig. 2, the SVM classifier correctly identified 3,228 normal instances and 2,599 abnormal instances, with 150 FP and 859 FN, indicating a moderate tendency to misclassify abnormal traffic as normal. This result reflects SVM's moderate effectiveness in detecting attacks, as a substantial number of abnormal instances were not detected. DT classifier shows some improvement over SVM, with 3,252 TP and 2,751 TN, and fewer FP (126) and FN (707). This suggests that DT is slightly more effective at identifying abnormal traffic compared to SVM, as seen in Figure 1. The improvement in abnormal detection demonstrates DT's enhanced capability in handling complex traffic patterns, though some misclassification still occurs.

ET demonstrates the highest accuracy among the classifiers, as shown in Fig. 2, with 3312 TP and 2873 TN, along with minimal FP (66) and FN (585). This matrix underscores ET's strength in distinguishing between normal and abnormal traffic, making it highly reliable for IDS with balanced classification results. ET's low misclassification rate is particularly suitable for scenarios where accuracy is critical. RF also performs well, as illustrated in Figure 1, with 3,249 TP and 3,043 TN, along with 129 FP and 415 FN. Although RF exhibits slightly more misclassification of abnormal traffic as normal compared to ET, it still maintains high accuracy, showing its effectiveness in intrusion detection. These results establish a baseline, showing that while feature selection can further enhance performance, RF and ET classifiers already demonstrate strong accuracy and reliability without PSO. This baseline provides a point of comparison for models with PSO in subsequent sections.

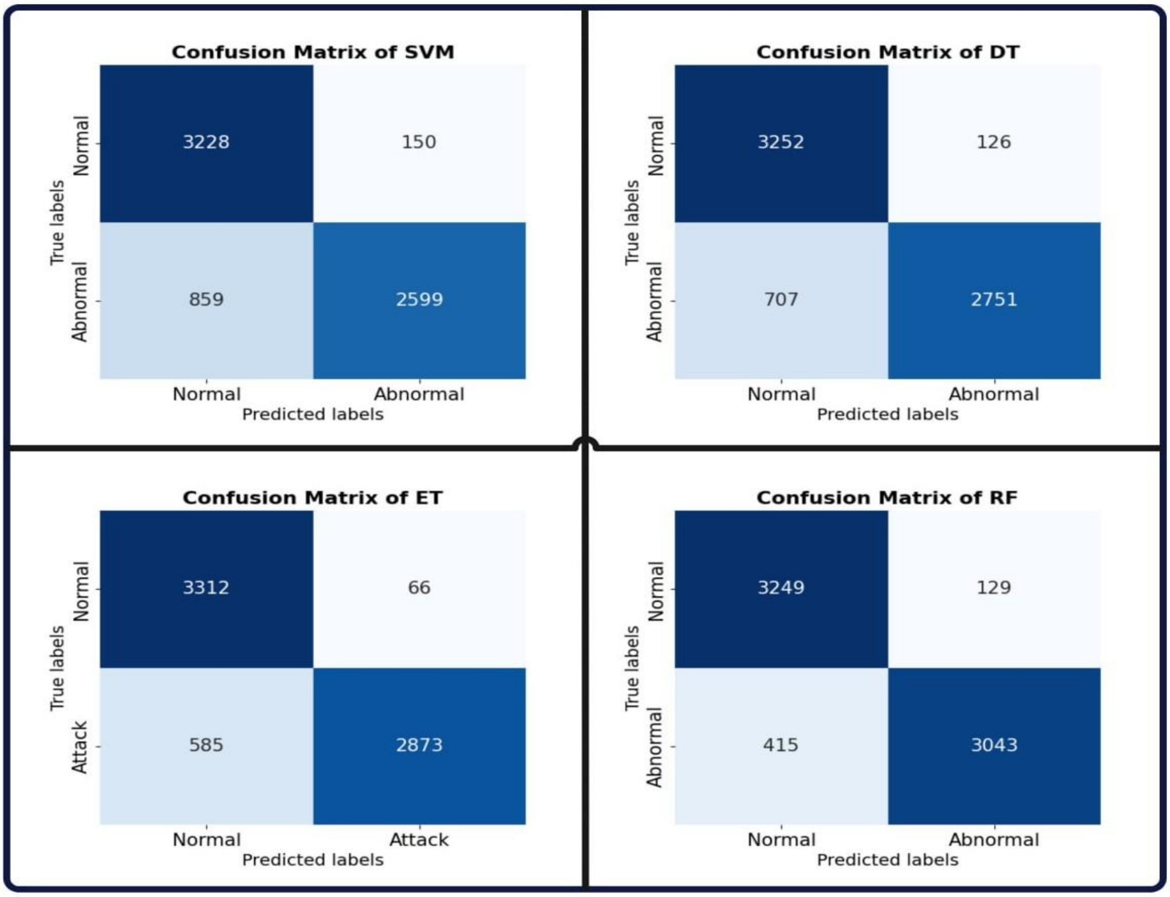

Fig 2- Confusion matrices of classifiers (SVM, DT, ET, RF) without PSO feature selection on NSL-KDD dataset.

4.2. Evaluation of machine learning classifier with PSO

In this subsection, we assess the performance of various ML classifiers on the NSL-KDD dataset after applying PSO for feature selection. The classifiers evaluated include SVM, DT, ET, and RF. Table 3 summarizes each classifier's performance using key metrics: Precision, Accuracy, Recall, and F1-Score.

The application of PSO significantly enhances the classifiers' effectiveness by selecting the most relevant features, which helps in reducing noise and focusing the models on critical attributes. As observed in Table 4, RF achieves the highest accuracy, reaching 98.33%, with an F1-Score of 98.32, indicating robust performance in detecting both normal and abnormal traffic. ET also performs exceptionally well, with an accuracy of 97.38% and an F1-Score of 97.35, showing strong reliability post-feature selection. DT and SVM also exhibit improved metrics, achieving accuracies of 96.91% and 92.12%, respectively, demonstrating that PSO enhances the model's ability to classify abnormal instances accurately.

Table 4 - Classifiers performance with PSO.

Classifier	Precision	Accuracy	Recall	F1-Score
SVM	96.54	92.12	88.54	92.37
DT	95.47	96.91	98.23	96.83

ET	97.22	97.38	97.48	97.35
RF	98.93	98.33	97.72	98.32

The confusion matrices in Fig. 3 display the performance of each classifier (SVM, DT, ET, and RF) on the NSL-KDD dataset after applying PSO for feature selection. These matrices provide insights into each classifier’s effectiveness in distinguishing between normal and abnormal traffic following feature optimization. Each matrix includes key elements: TP (correctly identified normal traffic), TN (correctly identified abnormal traffic), FP (normal traffic misclassified as abnormal), and FN (abnormal traffic misclassified as normal). For SVM, the matrix shows 3,261 TP and 3,036 TN, with 117 FP and 422 FN. This reflects an improvement in abnormal traffic classification compared to the results without PSO, as there are fewer FP and FN. DT's confusion matrix indicates strong performance with 3,225 TP and 3,400 TN, along with 153 FP and 58 FN. This substantial reduction in FN after PSO application enhances DT’s reliability in intrusion detection.

ET demonstrates high accuracy, with 3,284 TP and 3,373 TN, while recording only 94 FP and 85 FN. This outcome underscores ET's robustness in accurately classifying both normal and abnormal traffic, further strengthened by PSO’s feature selection, which minimizes misclassification. RF achieves the highest accuracy among the classifiers in Figure 2, with 3,342 TP and 3,380 TN. It has the fewest misclassifications, with only 36 FP and 78 FN, showing that PSO has significantly enhanced RF’s ability to identify both types of traffic accurately. Figure 2 highlights how PSO-driven feature selection improves each classifier’s performance, with RF and ET emerging as the most effective models due to their low FN and FP counts. These matrices underscore the impact of feature selection on model accuracy, particularly in identifying abnormal instances, which is essential for intrusion detection systems.

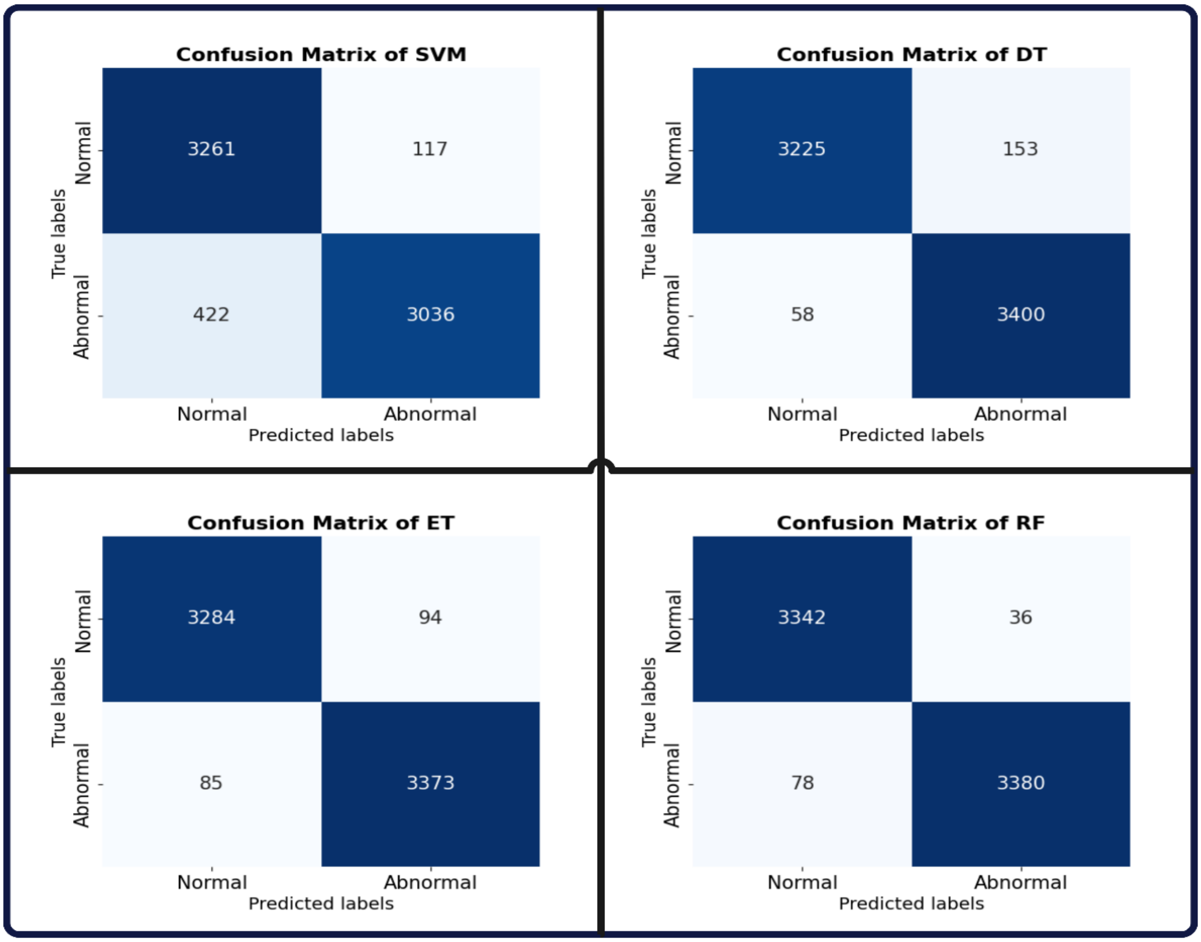

Fig 3- Confusion matrices of classifiers (SVM, DT, ET, RF) with PSO feature selection on NSL-KDD dataset.

4.3. Comparison of proposed approach with recent studies

This proposed approach offers an innovative method for intrusion detection, utilizing PSO for feature selection and leveraging various ML classifiers such as SVM, DT, ET, and RF. PSO significantly enhances feature selection by identifying the most relevant attributes, which directly contributes to the high classification accuracy observed in these models. Among the classifiers tested, RF, combined with PSO-selected features, achieved a remarkable accuracy of 98.33%, demonstrating the effectiveness of this methodology over other recent techniques.

Table 5 - Comparison with recent studies on the UNSW-NB15 dataset.

Author/year	Feature selection technique	Classifier	Recall	Precision	F1-Score	Best accuracy (%)
Amol D. Vibhute et al. [29] (2023)	Random forest	KNN	97.91	97.99	98.00	98.24
Shiravani et al. [30] (2023)	Fuzzy-based approach	SVM	97.4	97.5	97.44	96.89
Kasongo. [31] (2023)	XGBoost	LSTM	N/A	N/A	99.58	88.13
Fuat Türk. [32] (2023)	Ensemble learning	RF + MLP	98	97	98	97.8
Proposed	**PSO**	**RF**	97.72	**98.93**	98.32	**98.33**

As shown in Table 5, several feature selection techniques have been utilized in recent studies, including methods such as Random Forest, Fuzzy-based approaches, and XGBoost. Random Forest has been a common choice for feature selection across studies, achieving an accuracy of 98.24% when paired with KNN. Additionally, DL methods such as LSTM (88.13%) and ML methods like KNN and SVM (reaching up to 96.89% with Fuzzy-based selection) have also been used for classification. Ensemble models, like RF combined with MLP, have shown competitive performance, achieving an accuracy of 97.8%. However, the proposed approach, utilizing PSO for feature selection, outperforms these recent techniques, achieving the highest accuracy of 98.33% with the RF classifier and 97.38% with ET. The application of PSO significantly enhances the classifiers' effectiveness by selecting the most relevant features, which helps in reducing noise and focusing the models on critical attributes. Unlike traditional methods such as RF, which retain redundant features, PSO minimizes noise and ensures a compact feature subset. The above result depicts the efficiency of the proposed approach in eliciting better accuracy for intrusion detection on the NSL-KDD dataset..

5. Conclusion

This work proposes a new paradigm of intrusion detection, which is based on PSO-driven feature selection combined with different machine learning classifiers-SVM, DT, ET, and RF-to make the performance optimal. The obtained results indicate an improved model accuracy from PSO-driven feature selection since it selects only relevant features, reduces noise, and enhances computational efficiency. Among the classifiers tested, RF and ET achieved the highest accuracy, with RF reaching 98.33% and ET achieving 97.38%, highlighting the effectiveness of the proposed method compared to recent approaches in intrusion detection. Despite the promising results, this study has some limitations. The current approach relies solely on PSO for feature selection, which may limit the exploration of other potential feature subsets. Additionally, while the NSL-KDD dataset provides a solid benchmark, it does not fully capture the complexity of modern network environments, potentially affecting the generalizability of the model. For future work, it would be valuable to explore other feature selection techniques, such as Bat Algorithm (BA), Genetic Algorithm (GA), and Cuckoo Search (CS), to potentially enhance feature selection and further improve model accuracy. Integrating these techniques could provide a more comprehensive comparison and potentially uncover even more effective feature subsets. Furthermore, testing the proposed method on newer, more complex datasets could offer insights into its applicability in real-world, dynamic network environments, ensuring the robustness and adaptability of the intrusion detection system.

Acknowledgements

We would like to express our gratitude to all the individuals and institutions who supported and contributed to this research.

References

[1] S. Zavrak and M. Iskefiyeli, “Anomaly-based intrusion detection from network flow features using variational autoencoder,” IEEE Access, vol. 8, pp. 108346–108358, 2020.

[2] T. Kim and W. Pak, “Early detection of network intrusions using a GAN-based one-class classifier,” IEEE Access, vol. 10, pp. 119357–119367, 2022.

[3] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, “Toward a lightweight intrusion detection system for the internet of things,” IEEE access, vol. 7, pp. 42450–42471, 2019.

[4] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” IEEE communications surveys & tutorials, vol. 21, no. 1, pp. 686–728, 2018.

[5] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, “A novel two-stage deep learning model for network intrusion detection: LSTM-AE,” Ieee Access, vol. 11, pp. 37131–37148, 2023.

[6] P. Sun et al., “DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System,” Security and communication networks, vol. 2020, no. 1, p. 8890306, 2020.

[7] M. Samadi Bonab, A. Ghaffari, F. Soleimanian Gharehchopogh, and P. Alemi, “A wrapper-based feature selection for improving performance of intrusion detection systems,” International Journal of Communication Systems, vol. 33, no. 12, p. e4434, 2020.

[8] T. Vaiyapuri and A. Binbusayyis, “Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation,” PeerJ Computer Science, vol. 6, p. e327, 2020.

[9] B. Min, J. Yoo, S. Kim, D. Shin, and D. Shin, “Network anomaly detection using memory-augmented deep autoencoder,” IEEE Access, vol. 9, pp. 104695–104706, 2021.

[10] A. K. Pandey, P. Singh, D. Jain, A. K. Sharma, A. Jain, and A. Gupta, “Generative Adversarial Network and Bayesian Optimization in Multi-class Support Vector Machine for Intrusion Detection System,” Int. J. Intell. Eng. Syst, vol. 16, pp. 110–119, 2023.

[11] V. Kumar and D. Sinha, “Synthetic attack data generation model applying generative adversarial network for intrusion detection,” Computers & Security, vol. 125, p. 103054, 2023.

[12] L. Elmoiz Alatabani, E. Sayed Ali, R. A. Mokhtar, R. A. Saeed, H. Alhumyani, and M. Kamrul Hasan, “Deep and Reinforcement Learning Technologies on Internet of Vehicle (IoV) Applications: Current Issues and Future Trends,” Journal of Advanced Transportation, vol. 2022, no. 1, p. 1947886, 2022.

[13] H. Attou et al., “Towards an intelligent intrusion detection system to detect malicious activities in cloud computing,” Applied Sciences, vol. 13, no. 17, p. 9588, 2023.

[14] M. A. Hossain and M. S. Islam, “Ensuring network security with a robust intrusion detection system using ensemble-based machine learning,” Array, vol. 19, p. 100306, 2023.

[15] A. John, I. F. B. Isnin, S. H. H. Madni, and F. B. Muchtar, “Enhanced intrusion detection model based on principal component analysis and variable ensemble machine learning algorithm,” Intelligent Systems with Applications, vol. 24, p. 200442, 2024.

[16] Y. Yang, Y. Gu, and Y. Yan, “Machine learning-based intrusion detection for rare-class network attacks,” Electronics, vol. 12, no. 18, p. 3911, 2023.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” presented at the 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, 2009, pp. 1–6.

[18] M. N. Kadhim, D. Al-Shammary, and F. Sufi, “A novel voice classification based on Gower distance for Parkinson disease detection,” International Journal of Medical Informatics, vol. 191, p. 105583, 2024.

[19] D. Al-Shammary, M. N. Kadhim, A. M. Mahdi, A. Ibaida, and K. Ahmed, “Efficient ECG classification based on Chi-square distance for arrhythmia detection,” Journal of Electronic Science and Technology, vol. 22, no. 2, p. 100249, 2024.

[20] M. Sadiq, M. N. Kadhim, D. Al-Shammary, and M. Milanova, “Novel EEG Classification based on Hellinger Distance for Seizure Epilepsy Detection,” IEEE Access, 2024.

[21] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” presented at the 2007 IEEE swarm intelligence symposium, IEEE, 2007, pp. 120–127.

[22] M. Y. Hassan, A. H. Najim, K. A. Al-Sharhanee, M. N. Kadhim, N. F. Soliman, and A. D. Algarni, “A Hybrid Cuckoo Search-K-means Model for Enhanced Intrusion Detection in Internet of Things,” 2024.

[23] M. N. Kadhim, A. H. Mutlag, and D. A. Hammood, “Vehicle detection and classification from images/videos using deep learning architectures: A survey,” presented at the AIP Conference Proceedings, AIP Publishing, 2024.

[24] M. N. Kadhim, A. H. Mutlag, and D. A. Hammood, “Multi-models Based on Yolov8 for Identification of Vehicle Type and License Plate Recognition,” presented at the National Conference on New Trends in Information and Communications Technology Applications, Springer, 2023, pp. 118–135.

[25] A. Hussein, A. T. Abdulameer, A. Abdulkarim, H. Husni, and D. Al-Ubaidi, “Classification of Dyslexia Among School Students Using Deep Learning,” Journal of Techniques, vol. 6, no. 1, pp. 85–92, Mar. 2024, doi: 10.51173/JT.V6I1.1893.

[26] N. N. Ali, A. Hameed, A. G. Perera, and A. Al_Naji, “Custom YOLO Object Detection Model for COVID-19 Diagnosis,” Journal of Techniques, vol. 5, no. 3, pp. 92–100, Sep. 2023, doi: 10.51173/JT.V5I3.1174.

[27] A. S. Amsalam, A. Al-Naji, A. Y. Daeef, and J. Chahl, “Computer Vision System for Facial Palsy Detection,” Journal of Techniques, vol. 5, no. 1, pp. 44–51, Mar. 2023, doi: 10.51173/JT.V5I1.1133.

[28] S. I. Ibrahim, D. A. Hammood, and L. H. Abed, “Unconstrained face identification using machine learning classification,” AIP Conf Proc, vol. 3232, no. 1, Oct. 2024, doi: 10.1063/5.0236373/3316643.

[29] A. D. Vibhute, C. H. Patil, A. V. Mane, and K. V. Kale, “Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets,” Procedia Computer Science, vol. 233, pp. 960–969, 2024.

[30] A. Shiravani, M. H. Sadreddini, and H. N. Nahook, “Network intrusion detection using data dimensions reduction techniques,” Journal of Big Data, vol. 10, no. 1, p. 27, 2023.

[31] S. M. Kasongo, “A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework,” Computer Communications, vol. 199, pp. 113–125, 2023.

[32] F. Türk, “Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms,” Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, vol. 12, no. 2, pp. 465–477, 2023.