Misan Journal of Engineering Sciences ISSN: 2957-4250
Vol. 4, No. 1, June 2025 ISSN-E: 2957-4242

Optimizing Feature Selection for IOT Intrusion Detection Using RFE and PSO

Zahraa Mehssen Agheeb Al_Hamdawaee [1]

[1]Electrical Engineering Department, University of Misan ,Misan,Iraq
*Corresponding author E-mail: zahraa.mo.eng@uomisan.edu.iq

(Received 28 May, Revised 27 June, Accepted 27 June)

Abstract: Internet of things (IoT) and DoS attacks are two of the modern subjects currently being discussed and studied. In this paper, An approach the defense algorithm of IDS for IoT networks' security development contrary to attacks of DoS applying unusual ML and diagnosis has been presented. An anomaly detection is used in the provided IDS to control network traffic in an ongoing way for deviations from usual profiles. Four observed classifier algorithms have been applied: k-Nearest Neighbor (kNN), Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). Two feature selection mechanisms, which are Particle Swarm Optimization Algorithm (PSO) and Correlation-based Feature Selection Recursive Feature Elimination (RFE) have been used to compare their performances. The dataset of IoTID20 has been used, one of the most currently used to diagnose anomalous tasks in IoT networks, for checking our model. The best results were obtained using RF and kNN classifiers that were trained with features selected by RFE. kNN benefits from the smaller feature space since it focuses on distance measures, which are more successful with a refined set of features. RF improves decision-making by focusing on the most informative features, resulting in better overall performance. RFE notably improved kNN and DT accuracy, while SVM showed consistent results regardless of the feature of selection. These results highlight the importance of feature selection in optimizing classifiers for IoT intrusion detection , and achieved perfect scores (1,00) across all metrics.The aim from this paper is to enhance intrusion detection in iot networks by designing adual stage feature selection method based on RFE and PSO.

Keywords: Internet of things, Feature selection, Machine learning, DoS attacks, RFE, PSO.

1. Introduction

The widespread adoption of IoT in smart environments has significantly increased the attack surface for cyber threats. IoT networks require reliable and effective intrusion detection solutions. Voluminous and high-dimensional data, particularly in IoT environments, can lead to reduced accuracy, computational complexity, and an increased risk of overfitting for machine learning models. The wide and quick IoT technology has revolutionized the path human beings communicate with their surroundings, increasing smart industries, cities, and homes that combine devices and communication protocols seamlessly. However, IoT presents various advantages, and its escalating interconnectivity shows essential concerns of security. Such systems are broadly vulnerable to cyber-attacks, making IoT network's security and privacy critical for widespread adaptation and successful deployment [1].

DOI: https://doi.org/10.61263/mjes.v4i1.158

Cyber-attacks pose significant security risks in iot environments, shows security issues that need a stable and efficient intrusion detection system (IDS). IDS includes Network Intrusion Detection (NIDS) and a Host-based IDS. An NIDS is broadly applied as the defense of security for inferring system network function, diagnosing Cyber-attacks pose significant security risks in iot environments, in traffic of the network, and identifying suspicious system functions. A network-centric IDS diagnoses in observation of traffic across various means of network and inspects their performance info. Presenting reliable and effective NIDS refers to one of the basic issues in the security of networks [2]. The effective model of IDS needs more info for testing and training. The quality of data is essential for the outcomes of the IDS model. After collecting statistical qualities from data, observable features, and constituent units, low-quality and unessential info could be eliminated. The data might, however, be unbalanced, incomplete, high-dimensional, and excessive. Therefore, the study of IDS requires a provided dataset analysis completely [3]. Because of IoT devices' security restrictions, it is important to make NIDS that can quickly and dependably diagnose and avoid attacks on IoT networks. To this, a lot of ML methods have been improved for IDS in IoT, with general network traffic datasets. However, such sets of data often include several unrelated/extra features that affect ML models' complexity and accuracy. A typical strategy for improving effective NIDS is via a decrease of features that reduce network traffic data dimensionality fed into the ML model. It aids lower costs of calculation and latency when increasing model generalization [4].

Traditional ML-based IDS sometimes meet complexities in processing the broad and high-dimensional data created by IoT networks. High-dimensional data, including several features, could cause concerns such as degraded model performance, overfitting, and longer processing times. So, choosing the most related features becomes important in creating effective and appropriate IDS. The present study concentrates on a 2-step strategy of FS, which integrates Recursive Feature Elimination (RFE) and Particle Swarm Optimization (PSO) mechanisms for developing the process of FS and finally increasing IDS efficiency in IoT sites.

The exponential growth of IoT devices has developed attack surfaces, making IDS more complicated and needed. The high IoT data dimensionality includes important concerns for traditional IDS, as a lot of features might not be related/extra, including a small to-diagnosis process. Effective methods of FS are essential for decreasing computational complexity and developing IDS diagnosis accuracy. RFE is a typically applied technique that iteratively eliminates the least essential features given the model's weight, however, that might not always ensure the best FS because of its deterministic aspect. In other words, PSO is a heuristic optimization technique inspired by a social manner in nature that could look for optimum/near-optimal feature subsets in a more explorative behavior. Integrating RFE and PSO in a 2-step strategy of FS leverages the two algorithms' strengths: RFE for systematic feature removal and PSO for global search abilities. Such multiple strategies could cause more appropriate and computationally effective IDS for IoT networks, considering pressing equipment for powerful IDS in this quickly developing domain.

This study's main objectives are as follows:

i. Presents a new 2-step FS technique through combining RFE with PSO for developing FS efficiency in IoT-based IDSs.
ii. Shows that multiple strategies of FS could increase different ML models' accuracy for IDS in IoT areas, choosing the most related and informative features.

Some strategies have been presented for improving FS and IDS in IoT networks; however, they meet considerable issues, especially in controlling complicated, high-dimensional data and optimizing performance. RFE enables structured elimination of irrelevant features, while PSO allows global search for optimal feature subsets, and combining them provides both local precision and global exploration, improving detection performance.

The outline of the manuscript is section 2 disscusses related work; section3 presents the methodology;section4 detail the data set and preprocessing ;section 5 describes the experimental

setup;section6 discusses the results 7 concludes the work.

2.RELATED WORK

Awad and Fraihat [5] use a decision tree model as an estimator of Recursive Feature Elimination with cross-validation (RFECV). It restricts their technique generalization to other ML models that may not equally take advantage of chosen features. The problem is reducing the dimensionality of the UNSW-NB15 dataset to improve intrusion detection. Achieved feature reduction to 15 optimal features, but the approach is heavily dependent on tree-based models like RF.

Zhang et al. [6] present the developed Whale Optimization Algorithm (WOA-HA) increased by algorithms such as a binary operator, chaotic Hénon map, and adaptive coefficient vectors. Complicated needs of tuning can restrict the practical application of these algorithms in real-life IDS scenarios. The problem is enhancing search efficiency in WOA for feature selection. It improved search efficiency but increased complexity, reducing scalability for large datasets. Fang et al. [7] incorporate GA with a feature ranking fusion algorithm and a novel task of fitness for removing extra features. However, such a technique develops a global merit-seeking pace, and computational GA overhead, integrated with clustering methods' requirement. The problem is removing redundant features using GA and ranking fusion. Enhanced global merit-seeking but high computational cost for high-dimensional data.

Alsaffar et al. [8] present multiple FS techniques (MI-Boruta), integrating Mutual Information (MI) with the Boruta algorithm to assign optimum features. Such multiple strategies, when efficient, could define important computational overhead because of both filter and wrapper techniques' integration, raising the time required for FS. The problem is optimizing feature selection by combining MI and Boruta algorithms. Improved accuracy but increased computational overhead and time requirements.

Alsaffar et al. [9] examine multi-aim FS methods like multi-aim PSO. When such methods propose superior trade-offs among feature relevance and redundancy, they sometimes need broad multi-aim fine-tuning that could complicate use in dynamic IoT areas. The problem is balancing feature relevance and redundancy with multi-aim optimization. Achieved better trade-offs but faced complexity in dynamic IoT environments.

Li and Mao [10] present a 2-step FS technique that develops convergence pace by applying a grey predictive evolutionary algorithm (IBGPEA). The problem is improving convergence speed and diagnostic accuracy in feature selection. Enhanced accuracy and speed but added computational overhead for large-scale IoT data.

Rohini et al. [11] dealt with the imbalance of class by applying the Synthetic Minority Oversampling Technique (SMOTE) as well as features with CNN. Also, their technique that integrates the Arithmetic Optimization Algorithm (AOA) and Butterfly Optimization Algorithm (BOA) meets concerns for choosing the most optimum features because of developed computational needs and combination complexity, especially while coping with hybrid classifiers. The problem is addressing class imbalance and improving feature extraction for network traffic. Improved feature relevance but increased complexity and computational requirements.

Li and Yao [12] define a 2-step IDS model given the self-supervised learning to decrease dependence on labels and raise the pace of diagnosis. When it decreases model complexity, the model's dependence on special self-knowledge distillation methods might constrain its generalizability to other IoT sets of data, potentially decreasing its capability to control different and evolving attack models in IoT networks. The problem is reducing label dependency and accelerating intrusion detection. Lowered model complexity but faced limited generalizability.

Hosseini et al. [13] improved the multi-aim MOAEOSCA mechanism by hybridizing the sine-cosine algorithm (SCA) with Artificial Ecosystem-based Optimization (AEO) mechanisms for botnet diagnosis in IoT. When the mechanism targets covering the present strategies' weaknesses through combining Bitwise functions, Opposition-based learning (OBL), and Disruption operator, it describes important complexity. The problem is enhancing botnet detection in IoT using hybrid optimization. Improved detection but increased complexity and fine-tuning requirements.

Altulaihan et al. [14] apply supervised classifier integration (DT, RF, kNN, SVM) and FS mechanisms (CFS and GA) for IDS. However, although they show accurate developments, their strategy has a shortage of adaptability to real-life shifts of the network. Problem is improving IDS accuracy through feature selection and classifier integration. Enhanced accuracy but lacked adaptability to dynamic network conditions.

Bhavsar et al. [15] created an intrusion detection system (IDS) called Pearson-Correlation Coefficient-Convolutional Neural Networks (PCC-CNN). The PCC-CNN model combines the power of convolutional neural networks with essential properties that are retrieved using linear techniques. They also trained and assessed five PCC-based machine learning models: support vector machines, logistic regression, K-nearest neighbor, linear discriminant analysis, and classification and regression trees. The study's goal is to create an IDS that identifies network irregularities using deep learning techniques. The results show that the suggested model performs well in detecting various sorts of attacks.

Choudhary et al. [16] present a framework for Adaptive IDS for IoT, which can identify and mitigate attacks. The suggested framework uses the Convolutional Neural Network-Aquila Optimization (CNN-AO) model to predict traffic as anomalous or regular. The problem of this study is to develop a framework for IoT-compatible IDS that is capable of detecting and mitigating attacks. This system is able to accurately identify anomalies and activate countermeasures, which can contribute to the safety and security of IoT-based systems.

A lot of techniques depend on special ML classifiers, restricting generalizability over various models. In addition, complicated multiple algorithms sometimes develop computational overhead, making real-life IDS complex in resource-limited IoT areas. Multi-aim optimization techniques illustrate satisfaction; however need great fine-tuning, adding complexity to their deployment. When such strategies develop state-of-the-art in FS and IDS, the requirement for measurable, effective, flexible methods remains crucial to efficiently mention unique concerns that high-dimensional IoT data possess, as well as evolving cyber threats.

Table showing a summary of key related works, comparing their methods by this study based on methods, datasets, achieved accuracy, and limitations.

Table1. comparison of related studies and the proposed approach

study	method	dataset	accuracy	limitations
Awad et al. [5]	RFE + DT	UNSW-NB15	91%	Limited to tree-based models
Zhang et al. [6]	WOA-HA (Hybrid)	Custom Dataset	94%	High complexity and parameter tuning
Fang et al. [7]	GA + Ranking Fusion	High-dimensional Data	95%	Computational overhead
Alsaffar et al. [8]	MI + Boruta	Mixed	96%	Increased feature selection time
This Study	RFE + PSO	IoTID20	100%	Needs more cross-dataset validation

3. METHODOLOGY

For diagnosis accuracy development and processing effectiveness, this research illustrates two FS mechanisms: PSO and RFE for removing unrelated and extra features, guaranteeing that just the most related features are applied in the model. 4 observed mechanisms of classification: RF, SVM, KNN, DT are used for grouping network traffic. Every classifier has its strengths, from overfitting and SVM's effectiveness to RF's decrease DT's interpretability for dividing levels with max margin. Such a technique optimizes the two FS and classification steps, presenting a powerful strategy for real-life IoT IDS, especially in contrast to DoS attacks. This study is method architecture based on Figure 1, which could be divided into 3 steps: classification, data pre-processing, and feature decrease.

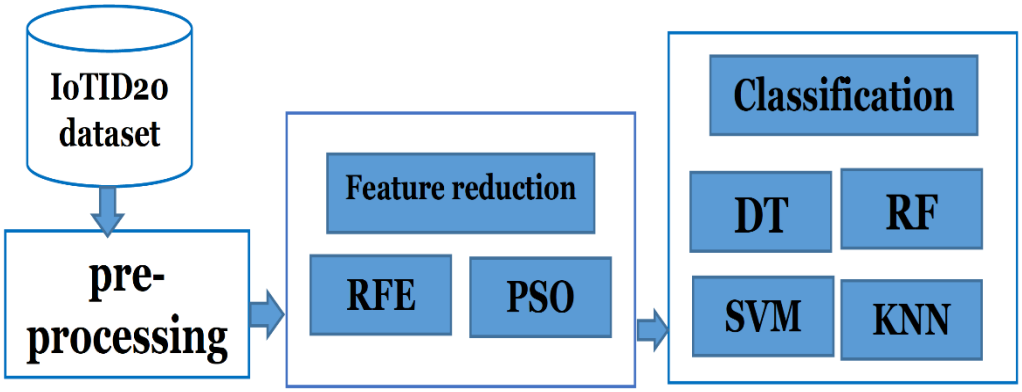

Figure 1. The framework of the proposed method.

The final evaluation stage uses accuracy, precision, recall, and F1 score as performance metrics.

3.1 IoTID20 Dataset

A lot of datasets exist that could be applied to training systems for DoS attack diagnosis. The set of data should include real-life network traffic. This is important because which set of data is versatile and broad. The dataset needs to contain the most recent DoS attacks as well as a broad variety of attack vectors. Because it includes simulated scans that are specifically directed at IoT networks as well as real-life traffic attacks, IoTID20 was chosen for this purpose to train the IDS for DoS attack detection.

The IoTID20 dataset covers many sorts of IoT assaults, including DoS, DDoS, Mirai, ARP Spoofing, and benign (regular) traffic. This dataset was obtained from smart home IoT ecosystems, which often comprise networked devices such as tablets, Wi-Fi cameras (EZVIZ), wireless access points, AI speakers (SKTNGU), laptops, and smartphones. In this configuration, the remaining devices served as attack tools, while cameras and AI speakers were identified as IoT victim devices. Nmap was used to mimic several attacks, such as scanning, distributed denial of service (DDoS), and man-in-the-middle. Furthermore, Mirai botnet assaults were created on a laptop and modified to mimic their impact on Internet of Things devices. The IoTID20 dataset was processed with CIC Flow Meter, which converted packet captures into CSV files. The CSV files were labeled based on IP addresses to indicate abnormal behavior and attack kinds. The dataset has 86 characteristics.

3.2 Data preprocessing

In this step, data is processed by partitioning, normalization, and cleansing for a standardized data format. It is shared in two sets, feature decrease and testing training, for the last model prediction.

3.2.1 Data partitioning

The dataset was split into 80% for training and 20% for testing using the hold-out method. The split was performed using the train-test split function from the Scikit-learn library with random shuffling enabled cross-validation(e.g, k-fold) was performed.

3.2.2 Normalization

To ensure consistent feature scaling, Min-Max normalization was applied to the dataset set This technique transforms each feature value to a [0,1] rang, using the equation:

$$x' = \frac{x - xmin}{xmax - xmin}$$

Where x is the original value, and x´ is the normalized value.

3.2.3 Cleaning

The preprocessing phase also included the removal of duplicate records and the handling of missing values. Records with null values were excluded to ensure data quality and consistency across the dataset.

3.3 Feature Selection Algorithms

For developing diagnosis accuracy and our system training pace, we required applying the algorithm of FS. FS includes removing unrelated and extra features and choosing those that are most pertinent and related. For the FS step, we decided to apply 2 FS mechanisms and compress among them, PSO and RFE.

3.4 D. PSO

PSO is the method of evolutionary computation (EC) presented by Kennedy and Eberhart in 1995 [17]. PSO is inspired by social behaviors like bird flocking and fish schooling. The main PSO event refers to the fact that knowledge is optimized by social communication in a population where thinking is not just private but social. PSO is given the rule that every solution could be shown as a particle in the swarm. A vector xi=($xi1$,$xi2$,…,xiD) represents each particle's location in the search space, where D indicates the search space's dimensionality. Particles hunt for the best answers by moving across the search space. Each particle is given a velocity to aid in this movement, which is expressed as vi=($vi1$,$vi2$,…,viD). Particles use their own and their neighbors' experiences to update their position and velocity while in motion. The "global best" ($g$best) is the best position found by the entire population, whereas the "personal best" ($p$best) is the best position each particle has found thus far. The Particle Swarm Optimization algorithm uses $p$best and $g$best to iteratively update each particle's position and velocity by a particular equation to find optimal solutions, as shown in [17]:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (1)$$

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * \left(p_{id} - x_{id}^{t}\right) + c_2 * r_2 * \left(p_{gd} - x_{id}^{t}\right) \quad (2)$$

Here, d∈D denotes the d-th dimension in the search space, and t denotes the t-th iteration in the evolutionary process. The inertia weight, or parameter w, regulates how much the particle's past velocity affects its present velocity. The particle's propensity to gravitate toward its own best ($k$best) and the global best ($g$best), respectively, is determined by the acceleration coefficients constants $c1$ and $c2$. The search procedure is guaranteed to exhibit stochastic behavior since the random variables $r1$ and $r2$ are independently produced values within the range [0,1]. The components of $p$best and $g$best in the d-th dimension are denoted by the words pid and pgd. The velocity is usually restricted within a predetermined range to avoid unpredictable behavior or divergence, guaranteeing algorithm stability and convergence by a predefined max velocity, vmax, and $v_{t+1}^{id} \in$ [−vmax, vmax]. Such a mechanism stops when the predefined variable is faced which can be a great amount of fitness/predefined max iterations number.

3.5 E. RFE

RFE is an RFE technique based on a wrapper that begins by removing predictors (features) recursively and creating a model relying on present predictors. That uses model performance (like accuracy) to decide which predictors involve more for showing the aimed predictor. RFE requires particular predictor numbers for maintaining, so, it is normally not known already how many predictors are optimum. For obtaining accurate predictors, the ML mechanism is applied with the RFE FS technique here. [18].compare to previous studies that combined RFE and PSO, the novelty in this work lies in the explicit two stage structure ,where RFE is applied first to remove clearly weak features based on model -derived importance scores ,followed by PSO optimization over the reduced set. This sequential structure allows the model to start from a cleaner subset, enhancing PSO convergence and reducing the search space. The RFE is applied first to eliminate irrelevant features using a random forest. After that, the number of particles was set to 20 maximum iterations to 30, in PSO, and the fitness function was based on classification accuracy.

4 CLASSIFICATION ALGORITHMS

The two ML algorithms are unsupervised and supervised. In supervised algorithms, predefined (grouped) things are applied for object-level prediction. Unsupervised algorithms, against, identify natural unlabeled things' classification. To get the best performance in our IDS, we would apply and compare four supervised learning mechanisms for classification.

4.1 A. Decision Tree

The first classification mechanism selected for assigning its performance to a group DoS attack is DT. Such a method is applied to solve the two issues of classification and regression; however in general, this is applied for issues of classification. Such a classifier is tree-structured in that the internal nodes show features of sets of data, branches show laws of decision, and leaves show results. DT possesses 2 nodes: leaf and decision nodes. Leaf nodes are decision node results and do not include any branches, but decision nodes are applied for deciding and possessing some branches. Features of dataset applied for deciding/ carrying on experiments. It is the way of getting feasible responses given the situation for a decision/issue [19].

DT is like trees in that it starts with a root node that develops branches and makes the entire tree-like structure. In a decision tree, there is one question, and given the response (yes/no), a subtree is made.

4.2 Random Forest

The second mechanism of the classifier that was chosen is RF. Applying the RF classifier, a training set subset is randomly chosen for making a decision trees set. Such a technique mainly includes creating several DTs from a randomly chosen training set subset and also integrating votes from every tree to make the last prediction. Considering the input of data, a model of classifier determines that to a group. For instance, a classifier could be applied for prediction if the image is for a dog/cat, based on an image set including dog and cat images. Initially, the mechanism of RF makes several DTs, every one of them given the random data subset. A DT is a mechanism kind that assigns which group data inputs fall into given the data inputs. By making some decision trees and averaging their outcomes, RFs go one stage above. Here, overfitting is decreased which happens when the mechanism just acts well with data of training and not with novel data.

It is feasible to consider RF as some DT ensemble. The last output is made by gathering several decision tree predictions (majority voting) and averaging them. So, the model of RF better generalizes to the broader population. In addition, the model becomes less prone to overfitting /high variance [20].

4.3 Support Vector Machine

SVM was the third classifier. SVM is comprehensively taken as a strategy of classification,

however, that could be applied to solving issues of regression. Plus, controlling ongoing criteria could control group criteria simply. For separating various levels, SVM creates a hyperplane in multidimensional space. Iterative SVM creates optimum hyperplanes that reduce errors. SVM is given the results of a max marginal hyperplane (MMH) for sharing a set of data into levels [21].

The basic aim is to divide the set of data as efficiently as feasible [21]. The SVM margin is defined as the separation between two locations. The goal is to find the hyperplane that optimizes the margin between the support vectors using the provided dataset. To do this, SVM determines the Maximum Margin Hyperplane (MMH) by following these steps:

1. Make hyperplanes that segregate levels in the best way.
2. Choose the right hyperplane which must possess max division from the closest point of data.

4.4 k-Nearest Neighbors

A supervised machine learning technique called the k-Nearest Neighbors (kNN) algorithm learns from labeled input data and applies that knowledge to forecast the right results for unlabeled data. kNNs are applied for test dataset prediction, given the training data features (labeled data). Predictions are made through computing distance among data of training and testing data, considering that points of data possess the same features. The k-Nearest Neighbors (kNN) technique is similar to a voting system in that a new data point's class label is determined by the majority class label among its k nearest neighbors. Consider, for example, that you must choose which political party to support in your tiny village with several residents. You could question your closest neighbors about their political inclinations to make this choice. You are more inclined to vote for party A if the majority of them do. Similar to this, kNN ensures a data-driven approach to categorization by assigning a new data point's class label based on the majority class label of its k nearest neighbors [22].

5 EXPERIMENTS AND RESULTS

In this section, we explore our study's general view, containing assessment metrics, features of the dataset, and the experimental area used. At last, we present our tests' meticulous testing and their outcomes' astute analysis.

5.1 Hardware and Environment Setting

ML classification models DT, RF, SVM, and KNN were used by applying Python 3.9.7. Performance applied different libraries like Numpy, Scikit-learn, and Pandas, among others, which makes FS facilitated and supported our tests' visualization as well as data processing.

Tests were performed on a computer with Windows 10 Enterprise 64-bit operating system. An NVIDIA Quadro T1000 graphics card, an Intel Core i7-10750H CPU with 16 cores and a clock speed of 2.6 GHz, and 12 GB of RAM are among the desktop hardware specifications.

5.2 Performance evaluation

In this section, we assess our presented model by applying various metrics of performance like F1 score, accuracy, AUCROC curve, recall, and precision. Such metrics are obtained from a matrix of confusion, which is a 2D table that compares certain and predicted levels and distinguishes classification results. The matrix of confusion is given the 4 values as:

i. True Negative (TN): the two basic and predicted data are false.
ii. True Positive (TP): The two basic and predicted data are true.
iii. False Negative (FN): basic data are true, and predicted data are false.
iv. False Positive (FP): basic data are false, and predicted data are true.

Accuracy refers to examples' rates which have been accurately grouped to whole samples' numbers. It could be described as:

$$\text{accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Recall or detection rate (DR), true positive rate (TPR), or sensitivity, refers to the whole TP cases percentage shared by the entire TP and FN cases number. This is computed as shown in the formula:

$$\text{Recall } = DR = TPR = \frac{TP}{TP + FN} \tag{4}$$

Precision refers to the TP cases' percentage shared by entire cases of TP and FP. The formula below shows precision:

$$\text{Precision } = \frac{TP}{TP + FP} \tag{5}$$

F1 score refers to harmonic recall and precision mean described by the equation:

$$F_1 = 2 \times \left(\frac{\text{Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} \right) \tag{6}$$

Table 2 shows the analysis's findings, which show that Random Forest is the best classifier in every situation. It performs best when paired with RFE, as demonstrated by the model's flawless accuracy, precision, recall, and F1 scores (1.00 for all metrics). This implies that Random Forest excels in differentiating between legitimate and malicious traffic and that RFE improves its performance by eliminating unnecessary features without sacrificing forecast accuracy. In a similar vein, the Decision Tree classifier has excellent performance, maintaining high precision and recall while achieving an accuracy of 0.99 when employing all characteristics. However, the accuracy of the Decision Tree marginally decreases to 0.97 when Particle Swarm Optimization (PSO) is used for feature selection, while the RFE maintains its outstanding performance (accuracy of 0.99, F1 score of 1.00). This demonstrates how RFE maintains or even increases classification accuracy, whereas PSO may cause slight performance decreases.

RFE also has a major positive impact on the k-Nearest Neighbors (kNN) classifier, whose accuracy rises from 0.96 (with all features) to 0.99 with RFE. This enhancement demonstrates how well RFE selects the most pertinent features for KNN, improving performance. PSO, however, does not appear to have an impact on KNN's performance because the outcomes are almost the same as when all characteristics are used. The Support Vector Machine (SVM) classifier, when using all features, PSO-selected features, or RFE-selected features, demonstrates consistent performance under all scenarios (accuracy of 0.94, F1 score of 0.97). This implies that SVM is less susceptible to feature selection, and thus, this model may already be at its best with the entire feature set.

Table 2. Performance Comparison of Classifiers with Different Feature Selection Methods

Classifier	**Feature Selection**	**Accuracy**	**Precision**	**Recall**	**F1 score**
DT	All features	0.99	1.00	0.99	0.99

	PSO feature	0.97	0.99	0.98	0.98
	RFE feature	0.99	1.00	0.99	1.000
RF	All features	0.99	0.99	0.99	0.99
	PSO feature	0.99	0.99	0.99	0.99
	RFE feature	1.00	1.00	1.000	1.000
CNN	All features	0.96	0.96	1.000	0.98
	PSO feature	0.96	0.96	1.000	0.98
	RFE feature	0.99	0.99	1.000	0.99
SVM	All features	0.94	0.94	1.000	0.97
	PSO feature	0.94	0.94	1.000	0.97
	RFE feature	0.94	0.94	1.000	0.97

For the models' quality quantification and comparison, we computed various scales from the matrix of confusion that contains the F1 score, accuracy, recall, and precision. Accuracy is the first scale that a model accuracy is a scale of how often it is accurate. Table 3 compares the accuracy obtained over four classifiers (kNN, DT, SVM, RF) as well as 3 cases of features (trained with whole features, trained with PSO-chosen features, trained with REF-chosen features).

Table 3. Accuracy results

Classifier	Without Feature Selection	With PSO	With REF
DT	0.9858	0.9795	0.9874
RF	0.9937	0.9874	0.9969
kNN	0.9575	0.9575	0.9921
SVM	0.9418	0.9418	0.9418

The obvious comparison can be observed in Figure 2. While trained with features chosen by mechanisms of RF and kNN, REF illustrated the best outcomes with 99% accuracy. Although the model of SVM with PSO features obtained less accuracy (94.18%), it happened because SVMs do not perform well with big sets of data with robust relations among features against classifiers of DT and RF.

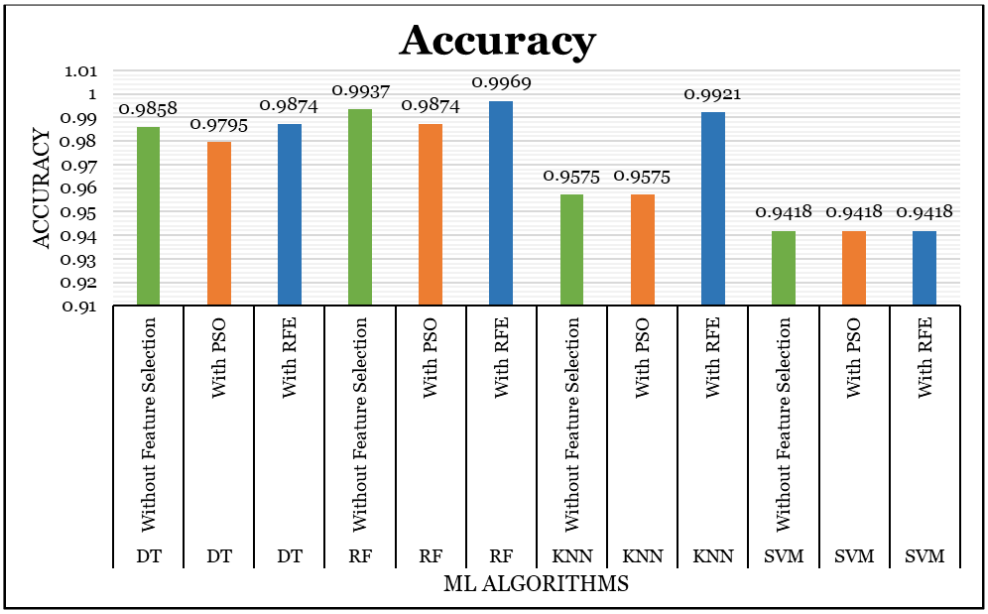

Figure 1. Accuracy Results

Table 4. Comparison classification of the proposed model with other models on the IoTID20 dataset.

Technique	Accuracy	Precision	Recall	F1-score
[15]	0.91	0.85	0.87	0.86
[16]	0.981	0.999	0.990	0.989
Proposed method	1.00	1.00	1.00	1.00

The results in Table 4 compare the performance of various strategies for a specific task (most likely an intrusion detection system or similar study). This table was analyzed as follows:

Technique [15] has the lowest accuracy (0.91) compared to other methods. However, the accuracy, recall, and F1 scores are 0.85, 0.87, and 0.86, respectively. This demonstrates that this methodology has average performance and can successfully detect samples at a lower level than other methods. This method's optimization and feature selection capabilities are most likely limited.

Technique [16] has much higher accuracy (0.981) than the prior method. This technique's performance has significantly improved, with precision, recall, and F1 values of 0.999, 0.990, and 0.989, respectively. These findings point to a more optimal algorithm with a better ability to detect threats. However, it has yet to achieve the full 1.00 level, indicating that additional improvements may be required. The proposed method achieves flawless detection accuracy, recall, and precision, with an F1 score of 1.00. This demonstrates that the suggested technique can accurately identify all samples with no positive or negative errors. This result demonstrates that the proposed method is extremely efficient and completely superior to previous techniques. The proposed method is recognized as a completely optimal and appropriate solution for the stated problem, receiving full points in all evaluation categories.

The approach [16] has extremely good performance and is similar to the proposed method, however, the technique [15] has lower performance than the other two ways. These findings demonstrate that the proposed strategy can greatly increase the accuracy and efficiency of detection systems.

5. Conclusions

The present article bolds IoTID20 dataset usage efficiency in IDS training to recognize IoT-based attacks, especially DOS ones. By leveraging real-life network traffic from smart home areas, a set of data proposes a new and general representation of the two manners which are benign and bad. The step of FS, applying PSO and REF, guarantees that just the most related data is applied, which increases the two systems' accuracy and effectiveness.

In addition, four observed learning classifiers comparison—kNN, SVM, RF, DT--- show the ML model's versatility to group network traffic. Although empirical confirmation impacted given the dataset of IoTID20 efficiently manifested presented method proficiency, getting preeminent performance in various metrics of assessment known as F1-score, accuracy, and recall.

The key findings reveal that using real network traffic data from smart home environments has resulted in a novel and comprehensive technique for identifying benign and malignant behaviors. The use of feature selection (FS) with PSO and RFE algorithms ensures that only relevant and valuable data is used for training, increasing the system's accuracy and efficiency. Also, the comparison of four machine learning algorithms (kNN, SVM, RF, and DT) shows that these algorithms have a high ability in grouping network traffic, and the proposed method has performed very well in evaluations.

These findings considerably add to existing knowledge in the field of IoT attack detection, demonstrating that feature selection optimization approaches and machine learning models can increase the accuracy and efficiency of intrusion detection systems. Future research should focus on enhancing the feature selection and classification models, testing them on various IoT datasets, and comparing the proposed method to existing optimization and machine learning methods. The perfect scores belong to several factors:

- The effective feature selection process removed irrelevant or noisy features, allowing the classifiers to train on high-quality inputs.
- The random forest classifier is known for its robustness and ability to handle complex datasets, which contributed significantly to these results.
- The IOTID20 dataset used is well-labeled and balanced, which reduces classification difficulty and enhances model performance. Despite these promising outcomes, several limitations must be

known:

* risk of overfitting: The model was trained and tested on a single dataset . while cross validation was used ,there is still a possibility that the results are overly specific to this dataset.

* lack of real-time evaluation: the proposed system was tested in an offline setting .its performance under real-time network traffic has not been validated.

* No cross-dataset validation: the ability of the system to generalize to other IOT datasets was not assessed. future work should test the model across diverse environments to verify robustness. The proposed approach can be applied in systems requiring fast and accurate feature-based filtering

Author Contributions: The author contributed to all parts of the current study.
Funding: This study received no external funding.
Conflicts of Interest: The author declare no conflict of interest.

References

[1] S. Al-Emari, Y. Sanjalawe, D. Alsmadi, E. Alduweib, and A. Alharbi, "Employing Mutual Information Feature Selection and LightGBM for Intrusion Detection in IoT," ICIC Express Letters, vol. 18, no. 6, pp. 597–606, 2024, doi: http://dx.doi.org/10.24507/icicel.18.06.597.
[2] A. MP, "Network Intrusion Detection Using Feature Selection Techniques: Bacterial Forage Optimization Algorithm.," International Journal of Intelligent Engineering & Systems, vol. 17, no. 5, p. 630, 2024, doi: 10.22266/ijies2024.1031.48.
[3] L. Zolfagharipour, MH. Kadhim, TH. Mandeel, "Enhance the Security of Access to IoT-based Equipment in Fog," In 2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT), pp. 142-146, 2023. IEEE. doi: https://doi.org/ 10.1109/AICCIT57614.2023.10218280.
[4] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," Journal of Big Data, vol. 11, no. 1, p. 36, 2024, doi: https://doi.org/10.1186/s40537-024-00892-y.
[5] M. Awad and S. Fraihat, "Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems," Journal of Sensor and Actuator Networks, vol. 12, no. 5, p. 67, 2023, doi: https://doi.org/10.3390/jsan12050067.
[6] K. Zhang, Y. Liu, X. Wang, F. Mei, G. Sun, and J. Zhang, "Enhancing IoT (Internet of Things) feature selection: A two-stage approach via an improved whale optimization algorithm," Expert Systems with Applications, vol. 256, p. 124936, 2024, doi: https://doi.org/10.1016/j.eswa.2024.124936.
[7] Y. Fang, Y. Yao, X. Lin, J. Wang, and H. Zhai, "A feature selection based on genetic algorithm for intrusion detection of industrial control systems," Computers & Security, vol. 139, p. 103675, 2024, doi: https://doi.org/10.1016/j.cose.2023.103675.
[8] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," Journal of Big Data, vol. 11, no. 1, p. 133, 2024, doi: https://doi.org/10.1186/s40537-024-00994-7.
[9] M. Zhang, J. Du, B. Nie, J. Luo, M. Liu, and Y. Yuan, "Hybrid mRMR and multi-objective particle swarm feature selection methods and application to metabolomics of traditional Chinese medicine," PeerJ Comput. Sci., vol. 10, p. e2073, 2024, doi: https://doi.org/10.7717/peerj-cs.2073.
[10] M. Li and S. Mao, "Two-Stage Feature Selection Algorithm Based on an Improved Grey Predictive Evolutionary Algorithm for the Intrusion Detection System," Jul. 11, 2023, Social Science Research Network, Rochester, NY: 4506360. doi: 10.2139/ssrn.4506360.
[11] G. Rohini, C. Gnana Kousalya, and J. Bino, "Intrusion Detection System with an Ensemble Learning and Feature Selection Framework for IoT Networks," IETE Journal of Research, vol. 69, no. 12, pp. 8859–8875, 2023, doi: https://doi.org/10.1080/03772063.2022.2098187.
[12] Z. Li and W. Yao, "A two stage lightweight approach for intrusion detection in Internet of Things," Expert Systems with Applications, vol. 257, p. 124965, 2024, doi:

https://doi.org/10.1016/j.eswa.2024.124965.
[13] F. Hosseini, F. S. Gharehchopogh, and M. Masdari, "MOAEOSCA: an enhanced multi-objective hybrid artificial ecosystem-based optimization with sine cosine algorithm for feature selection in botnet detection in IoT," Multimedia Tools and Applications, vol. 82, no. 9, pp. 13369–13399, 2023, doi: https://doi.org/10.1007/s11042-022-13836-6.
[14] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," Sensors, vol. 24, no. 2, p. 713, 2024, doi: https://doi.org/10.3390/s24020713.
[15] M. Bhavsar, K. Roy, J. Kelly, O.Olusola, "Anomaly-based intrusion detection system for IoT application," Discover Internet of things, vol. 3, no. 1, pp. 5, 2023, doi: https://doi.org/10.1007/s43926-023-00034-5.
[16] V. Choudhary, S. Tanwar, T. Choudhury, "Evaluation of contemporary intrusion detection systems for internet of things environment," Multimedia Tools and Applications, vol. 83, no. 3, pp. 7541-81, 2024, doi: https://doi.org/10.1007/s11042-023-15918-5.
[17] B. Xue, M. Zhang, and W. N. Browne, "Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach," IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1656–1671, 2013, doi: https://doi.org/10.1109/TSMCB.2012.2227469.
[18] H. A. Al Essa and W. S. Bhaya, "Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system," Bulletin of Electrical Engineering and Informatics, vol. 13, no. 1, pp. 665–676, 2024, doi: https://doi.org/10.11591/eei.v13i1.5844.
[19] S. Dasari and R. Kaluri, "An Effective Classification of DDoS Attacks in a Distributed Network by Adopting Hierarchical Machine Learning and Hyperparameters Optimization Techniques," IEEE Access, vol. 12, pp. 10834–10845, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3352281.
[20] N. Karmous, M. O. Aoueileyine, M. Abdelkader, L. Romdhani, and N. Youssef, "Software-Defined-Networking-Based One-versus-Rest Strategy for Detecting and Mitigating Distributed Denial-of-Service Attacks in Smart Home Internet of Things Devices," Sensors, vol. 24, no. 15, p. 5022, 2024, doi: https://doi.org/10.3390/s24155022.
[21] M. D. Kumar and T. J. Nagalakshmi, "Design of intrusion detection system for wireless adhoc network in the detection of DOS attack using one class SVM with random forest feature selection comparison with information gain algorithm," AIP Conference Proceedings, vol. 2853, no. 1, p. 020131, 2024, doi: https://doi.org/10.1063/5.0197410.
[22] F. Rizvi et al., "An evolutionary KNN model for DDoS assault detection using genetic algorithm based optimization," Multimedia Tools and Applications, vol. 83, no. 35, pp. 83005–83028, 2024, doi: https://doi.org/10.1007/s11042-024-18744-5.
[23] H., S. "A Novel Intrusion Detection Framework (IDF) using Machine Learning Methods," *Journal of Cybersecurity and Information Management*, vol. , no. , pp. 43-54, 2022. **DOI:** https://doi.org/10.54216/JCIM.100103
[24]Qaddos, A., Yaseen, M.U., Al-Shamayleh, A.S. et al. A novel intrusion detection framework for optimizing IoT security. Sci Rep 14, 21789 (2024). https://doi.org/10.1038/s41598-024-72049-z