# An Efficient Feature Selection Method for Network Attack Detection Using PSO-Based Wrapper Technique

**Alaa Shareef Shalef [1], Razieh Asgarnezhad[2*]**

[1] *Alaa Shareef Shalef, Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.*

[*2] *Razieh Asgarnezhad, Department of Computer Engineering, Aghigh Institute of Higher Education, Shahinshahr, Isfahan, Iran.*

*Corresponding Author: r.asgarnezhad@aghigh.ac.ir*

**Abstract: -** Given that internet usage and connectivity are in such great demand right now, the steep increase in network attacks has been a big source of concern for cyber security. Fog computing can provide low-latency services for cloud and mobile users as an add-on to cloud computing. Fog devices may experience security difficulties due to the proximity of the end users to the fog nodes and the lack of suitable computing capacity. Fog computing system destruction may result from conventional network threats. Applying Intrusion Detection Systems (IDS) directly to the fog computing platform may be inappropriate given the vast research on their use in conventional networks. It is crucial to construct an intrusion detection system model over huge datasets in the fog computing environment because nodes of the fog frequently generate enormous amounts of data. An intrusion detection system (IDS), a strategic intrusion prevention innovation that can be used in the fog computing platform and utilize machine learning techniques for network anomaly detection and network event classification threat, has shown to be effective and efficient in defending against some of these network attacks. In order to reduce time complexity and create an improved model that can predict outcomes more accurately, this paper presented a Particle Swarm Optimization (PSO) Algorithm Wrapper-Based feature selection and Nave Bayes for Anomaly Detection Model in a Fog Environment. It uses the Security Laboratory knowledge Discovery Dataset (NSL-KDD). According to the data, the developed system performs better overall, with an accuracy rate of 98.27 percent and a false positive rate of just 1.6%. The results demonstrate that the suggested strategy outperforms comparable approaches in the literature.

**Key Words—** ***Intrusion Detection System, Particle Swarm Optimization, Wrapper technique.***

## I. Introduction

The Internet has quickly evolved into one of the most significant innovations in human history. The importance and influence of the Internet can be observed in a variety of areas, including travel, commerce, research, and education. IoT is one of the Internet's newest and most popular uses. IoT has developed as an industrial revolution in the past ten years as a result of the popularity and use of inexpensive, effective devices like sensors, actuators, and other similar devices, along with a number of communication media.

Manuscript revised September 19, 2022; accepted September 20, 2022. Date of publication September 21, 2022.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

These days, IoT devices are ubiquitous. According to Cisco, there will be more than 50 billion IoT-based devices connected to the Internet by 2020 [1].

Smart environments, such as smart cities, smart grids, smart houses, etc., are being developed using the IoT paradigm. Smart settings are designed to make human life better by making it more comfortable. One of the examples is the smart city of Padova in Italy [2]. Almost every industry, including healthcare, education, smart cities, energy distribution, and transportation, uses IoT. Because IoT-based devices constantly collect and transmit our personal data over the Internet, an attacker or intrusive party has the ability to remotely access these applications, disrupt their normal operation, and even result in fatalities. Many IoT devices deployed in smart homes were impacted by the Mirai virus assault on Dyn, a Domain Name System (DNS) service, in 2016 [3]. IoT-based smart environments' available services are impacted by attacks like DoS and malicious control. In terms of battery life, computing power, network bandwidth, and memory capacity, IoT devices have various restrictions. IDSs that are built specifically for IoT networks must be reliable and safe because typical IDSs cannot be used in IoT-based networks. IDS, which can be hardware or software, guards against intrusions that jeopardize the integrity, confidentiality, and availability of an information system by keeping track of traffic data. IDSs can generally be divided into two types: SIDSs (Signature-based intrusion detection systems) and AIDSs (Anomaly-based intrusion detection systems) (AIDS). In SIDS, a database is kept with the signatures of well-known attacks, and incoming traffic is compared to this database. The administrator gets informed if there is a match. On the other hand, AIDS models typical user system behavior. The fundamental premise is that harmful operations depart from customary user behavior. AIDS is able to recognize unknown or zero-day assaults since it does not rely on established patterns. This restricts the use of SIDS in IoT networks since IoT devices are heterogeneous and vary in nature, making it practically unfeasible to rely on pre-defined attack signatures [4].

The majority of service providers use a centralized cloud-based security system for IoT-based IDS. Communication restrictions, such as those related to data transfer capacity requirements, power usage, memory utilization, and latency, apply when centralizing the enormous data generated by a large number of IoT devices. Due to its decentralized architecture, IoT systems based on opportunistic networks are more susceptible to security issues. As a result, the distributed nature of the Internet of Things (IoT) necessitates a dispersed security mechanism that supports interoperability, scalability, and flexibility with a single security mechanism across its heterogeneous devices [5]. Fog computing can therefore be employed in the context of a distributed architecture because it offers dispersed services for computational offloading.

To examine the data produced by IoT sensors, the concept of machine learning is employed as an analytical tool and is coupled with fog nodes. The suggested approach preprocesses data using Label-One-Hot-Encoding, which has limited the amount of characteristics and prevented the incoming traffic from becoming sparse. The Correlation Coefficient approach is used for feature selection, and it has removed strongly linked data from network traffic. There are a number of free datasets, including KDDCUP99, NSL-KDD, and CICIDS-2017, that can be used to assess the performance of IDS, however they either lack IoT-based attacks or are out of date [6]. We used an actual IoT-based dataset that was developed using the Distributed Smart Space Orchestration System (DS2OS) in an IoT-based environment and incorporates a variety of IoT-based threats in order to assess the performance of our model [7].

We have contrasted our approach with various categorization techniques in order to assess the performance of the suggested model. Our model exhibits the following benefits:

- To analyze the massive amounts of data produced by IoT sensors, machine learning analytical tools are merged with fog computing.
- The suggested model makes use of the actual NSL-KDD dataset, which includes several recent IoT attacks.
- It has been well studied how feature selection affects assault detection.
- extensive research using the 10-fold cross validation resampling technique and several performance metrics including F1 and recall.

We have talked about relevant work in the section after this. Section 3 describes the suggested model; Section 4 discusses the paper's evaluation, analysis, and comparison. We finished the paper with future scopes in Section 5.

## II. Related Work

The literature has recently presented a large number of IDSs that are used to monitor IoT-based networks against various assaults. The authors of [7] were able to simulate the communication behavior of IoT sites by observing the traffic of IoT services. Using Distributed Smart Space Orchestration System (DS2OS) in an IoT environment, they produced an IoT-based dataset. Their approach makes use of the BIRCH algorithm and K-means clustering to find anomalies (balanced iterative reducing and clustering using hierarchies). The accuracy of their suggested model was 96.3 percent. To get rid of linked features, this model did not employ any feature selection methods.

Another IDS for anomaly detection for IoT sensors and IoT locations employing various machine learning algorithms is proposed in [8]. Random Forest (RF) has outperformed and achieved an accuracy of 99.4% in the suggested model. The suggested model is tested using a DS2OS-generated IoT-based dataset. However, this model has not employed any feature selection techniques to eliminate correlated and duplicate features. They use the entire dataset for their experiment.

The authors of [9] have suggested a model that makes use of a two-layer dimension reduction technique that combines PCA and LDA. The effectiveness of feature reduction methods in anomaly detection has been demonstrated by this model. The

Certainty Factor version of K-Nearest Neighbor and the Naive Bayes classifier are used in this model to detect anomalies in IoT backbone networks. Using the NSLKDD dataset, the performance of this model has been assessed. The model attempted to recognize assaults like Remote to Local (R2L) User to Root (U2R).

The authors of [10] have suggested a unified IDS for an IoT ecosystem. K-means clustering is utilized during the preprocessing phase to determine the similarity between the features. They chose features using the information gain technique, and the UNSW-NB15 dataset was utilized to evaluate their proposed model. Their model has improved the network traffic's accuracy and attack detection rate. Their suggested model has a FAR of 3.80 percent and an accuracy of 88.92 percent.

It is suggested to use transfer learning as the foundation of a mobile IDS for IoT systems. PCA is employed in [11] for dimension reduction, and experiments are run on the KDDCUP99 dataset. K-means clustering technique is used during the preprocessing step to group data with similar components into one class. Their model, which has characteristics reduced from 41 to 8-16, has a high detection rate of 96.8% and a FAR of 1.6%. Fog-based IDS, where heavy computation is offloaded through fog nodes, has been proposed by many researchers.

For IDS in an IoT network, authors of [12] proposed cognitive fog computing. Instead of using a centralized cloud-based infrastructure, the suggested model may identify malicious activity in local fog nodes, and a summary of all fog nodes is saved on the cloud for further analysis. The suggested model is evaluated on NSL-KDD dataset and the online sequential extreme learning machine (OSELM) technique is employed for detection. Their model has a 97.36 percent accuracy rate and a 0.37 percent FAR.

A detector utilizing a trust joint light probe based defense (TLPD) mechanism has been suggested by the authors in [13]. Their suggested model aims to recognize an assault known as a "on-off" that can harm wireless sensor networks in industrial communication systems. An IoT network may be attacked by a defective node when it is in the on or active state in an on-off attack, yet the IoT network functions correctly when the specific defective node is in the off or inactive state. The authors of [14] have suggested a deep learning-based IDS. This model showed that for IoT-based applications, distributed fog threat detection is more scalable than centralized cloud. Their deep model has a 99.20 percent accuracy rate for binary classification and a 98.27 percent accuracy rate for multi-class classification. The NSL-KDD dataset is used to test this model. [15] makes a suggestion for an adaptive IDS for IoT that can identify DoS threats. Wireshark is used in this study to gather a noel dataset over the course of four days on an IoT testbed. The Naive Bayes classifier performs worse than their proposed model. In [16], a locust swarm optimization-based IDS employing neural networks is proposed. The accuracy and FAR for this experiment, which uses the NSL-KDD and UNSW-NB15 datasets, are 94.04 and 2.21 percent, respectively. Authors in [17] provide yet another IDS for multi-agent environments. The technique of naive Bayes classification is used to find DDoS attacks. This paradigm is implemented as a set of multi-agents that are placed throughout the network to look for malicious node activity.

## III. Technique

### 3.1. PSO feature selection method based on wrappers

A swarm of potential solutions is initially seeded randomly by the PSO algorithm. The NSL-KKD dataset's characteristics and attributes determine whether a bit, character, or integer is used to represent the particles [21] [22]. Following that, particles are assessed using their fitness function. The current fitness value is then set as the new personal best and the particle with the best fitness value is selected as the best particle out of all particles if it is better than the personal best objective fitness value in history. Finally, current particle positions and velocities help to improve the solutions. The process develops until a maximum number of iterations is reached, as demonstrated in Algorithm 1. At the conclusion of the dimension reduction, the NSL-features dataset was reduced to 8.

### 3.2. Bayesian algorithm

The wrapper strategy, as shown in Fig. 1, requires a classification algorithm to evaluate how well the feature subset performs. This is achieved by utilizing a Bayesian classifier in this work, which is also utilized to classify the attacks into normal, Probe, DoS, R2L, and U2R using the selected features. The Bayesian classifier is built on the Bayes theorem, which is based on the assumption that qualities are distinct [18].

#### *3.2.1. Using mathematics as a basis*

in light of a feature vector $F = \{f_1, f_2, \dots, f_n\}$ and a class variable $L_i$, Bayes theorem states that:

$$P(L_i \mid F) = \frac{P(F \mid L_i) * P(L_i)}{P(F)}, P(F) \neq 0 \qquad (1)$$

$P(F \mid L_i)$ is the posterior probability where $i = 1,2,3, \dots, k$; $P(F/L_i)$ is the probability; $P(L_i)$ represents the class's previous probability; $P(F)$ represents the predictor's prior probability.

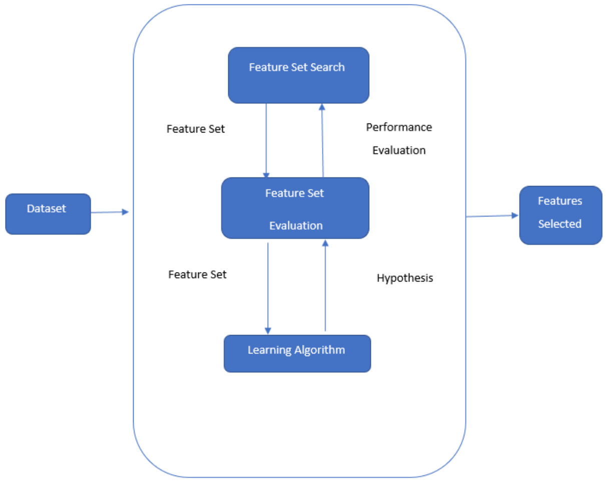

Fig.1. Feature selection using a wrapper approach.

**Algorithm 1: Wrapper based PSO Algorithm for Feature Selection**

```
Input: Dataset; Initialize parameter; Maximum number of iterations: $T_{max}$
Output: Relevant subset features
Initialize each particle and its velocity $V$ and location $X$;
repeat
    for each particle $i$ do
        calculate the fitness value $f_i$ of particle $i$;
        if $f_i < f_{pbest_i}$ Then
            $pbest_i = X_i$;
        end
        if $f_i < f_{gbest}$ Then
            $gbest = X_i$;
        end
        update $X_i$ and $V_i$;
    end
until $T_{max} = max$ or $f_{gbest} = 0$;
Return $gbest$
```

Consequently, by applying the chain rule, the chance $P(F/L_i)$ may be broken down into:

$P(F \mid L_i)$ is the posterior probability where $i = 1,2,3, \dots, k$;

$$P(F/L_i) = P(f_1, f_2, f_3 \dots f_n \mid L_i)$$
$$= P(f_1 \mid f_2, f_3 \dots f_n \mid L_i)$$
$$\times P(f_2 \mid f_3, f_4 \dots f_n \mid L_i) \dots P(f_{n-1} \mid f_n, L_i) P(f_n \mid L_i) \quad (2)$$

Although calculating Eq. (2) can be time-consuming and expensive, it is based on the naive independence assumption, which claims that:

$$P(f_j \mid f_{j+1}, \dots, f_n \mid L_i) = P(f_j \mid L_i) \quad (3)$$

We can get:

$$P(F/L_i) = P(f_1, f_2, f_3 \dots f_n \mid L_i) = \prod_{j=1}^{n} * P(F_j \mid L_i) \quad (4)$$

Posterior probability is expressed as follows:

$$P(F/L_i) = P(f_1 . f_2 . f_3 \dots f_n \mid L_i)$$
$$= \frac{P(L_i \mid F) \prod_{j=1}^{n} * P(F_j \mid L_i)}{P(F)} . P(F)$$
$$\neq 0 \quad (5)$$

Since the priority probability of predicator $P(F)$ is constant given the input, we have:

$$P(F/L_i) \alpha \, P(L_i) * \prod_{j=1}^{n} * P(F_j \mid L_i) \quad (6)$$

Now, different class of values of $L_i$ is obtained by finding the maximum of:

$$P(L_i) * \prod_{j=1}^{n} * P(F_j/L_i) \quad (7)$$

As

$$L_m = argmaxP(L_i) * \prod_{j=1}^{n} * P(F_j/L_i) \quad (8)$$

To determine the a priori probability of class $P(L_i)$, the relative frequency of class $L_i$ in the training data could be employed. The Bayesian classifier used to implement the model is shown in equation (8).

### 3.3. The method suggested

It was used as a feature search algorithm in the first stage of the wrapper feature selection approach during the data pre-processing and training phase of suggested method, as mentioned in Algorithm 1. In feature selection, PSO's rapid discovery of large search areas makes it an indispensable tool. Furthermore, PSO does a global search as opposed to a local, or greedy search, unlike many other search algorithms. The particles are assessed and tested for algorithm termination at the conclusion of each iteration. The position and velocity of particles are current if the termination criterion is not met. Until the halting requirement is satisfied, this procedure is repeated. PSO is used in conjunction with the Naive Bayes classifier as a random search strategy. The proposed model's operation is shown in Fig. 2.

### *3.4. Preparation of the dataset and its description*

The proposed model uses the benchmark dataset from NSL-KDD as evaluation data. A part of the data that contains 25192 samples is used for simulation in this work. Each sample contains 41 continuous and discrete features, and column 42 indicates the label of one of the types of attacks or normal mode. The total number of normal and attack cases in each fold of testing and training data are shown in tables 1 and 2, respectively. In order to reduce the complexity and misclassification error of the proposed model, we went through three stages of pre-processing. Stage 1: String digitization; Stage 2: Data normalization; Stage 3: Class imbalance management.

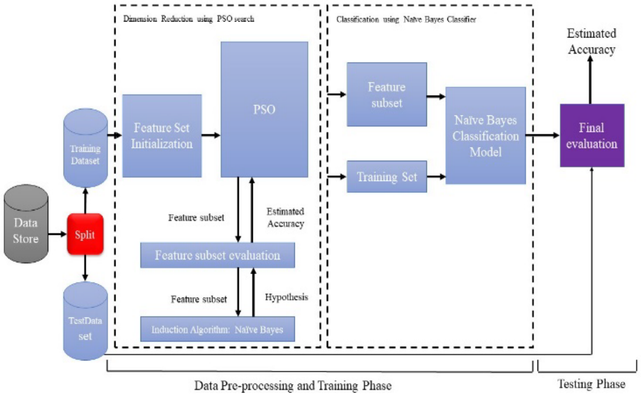
Dimension Reduction using PSO search
Classification using Naïve Bayes Classifier
Estimated Accuracy
Data Store
Split
Training Dataset
TestData set
Feature Set Initialization
PSO
Feature subset
Estimated Accuracy
Feature subset evaluation
Feature subset
Hypothesis
Induction Algorithm: Naïve Bayes
Feature subset
Training Set
Naïve Bayes Classification Model
Final evaluation
Data Pre-processing and Training Phase
Testing Phase

Fig.2. Proposed method frame work.

a second stage (Data normalization). As a result, the values for the dataset's properties vary widely. While qualities with large values may conceal the importance of lower values, it is possible that some attributes with smaller values are critical to classification. Prior to executing the indicated method, we normalized the dataset. It is a goal of normalization to lower the size of the underlying information (0-1). A zero-mean, one-standard-deviation normalization approach was used in this instance.

Phase 3 (Class Imbalance). Table 1 shows that the NSL-KDD datasets have an unequal distribution of classes. Additionally, the model may misclassify R2L and U2R classes. Synthetic Minority Oversampling (SMOTE) was used to correct the imbalance in the dataset, allowing the dataset to be successfully categorised (see Tables 2 and 3).

Unknown sorts of hazards are discovered in the training data, whilst recognized hazardous ones are found in the tests. In both training and testing data, the most common assaults are Probe, DOS, R2L, and U2R attacks.

Table.1. Number of each attack types in KDD-train database.

| Attack Types | Number of samples |
|---|---|
| **Normal** | 67343 |
| **DoS** | 45927 |
| **Probe** | 11656 |
| **R2L** | 995 |
| **U2R** | 52 |
| **Total** | **125973** |

Table.2. Number of each attack types in KDD-test database.

| Attack Types | Number of Sampels |
|---|---|
| **Normal** | 9711 |
| **DoS** | 7452 |
| **Probe** | 2421 |
| **R2L** | 2756 |
| **U2R** | 200 |
| **Total** | **22544** |

Table.3. detail of Attack cases in the dataset

| Attacks kind | Attack Type |
|---|---|
| DoS | Back, Smurf, Neptune, Land, Teardrop, Pod, Mailbomb, Udpstorm, Apache2, Processtable, Worm. |
| Probe | Satan, Saint, Portsweep, IPsweep, Mscan, Nmap, |
| R2L | Ftp_write, Multihop, Guess_password, Xlock, Imap, Xsnoop, Snmpguess, Phf Httptunnel, Snmpgetattack, Sendmail, Warezmaster, Named |
| U2R | Buffer_overflow, Xterm, Rootkit, Perl, Loadmodule Sqlattack Ps |

### *3.5. Evaluation metrics*

Performance indicators in this study include accuracy, precision, F1 score, and execution time. These metrics are defined as follows:

(1) F1 Score: There are four categories of categorization Consider what happens if we divide a sample dataset into two categories: normal and abnormal. The True Positive, False Positive, False Negative, and True Negative are shown in Table 4 respectively. If the categorization is True, it is accurate, and if it is False, it is incorrect. Positive samples indicate that the classifier is divided into the normal category, while negative samples indicate that the category is separated into the abnormal category.

(1) True Positive: The correct normal instance is recognized.

(2) False Positive: An abnormal occurrence is mislabeled as normal.

(3) False Negative: Misclassification of a normal instance as abnormal.

(4) True Negative: The correct anomalous case is recognized.

The number of pertinent instances among the discovered instances is represented by precision (P). P can be calculated using the formula below:

$$P = \frac{TP}{TP + FP} \qquad (9)$$

The percentage of relevant occurrences that have been found over all relevant instances is known as recall (R). R can be calculated using the formula below:

$$R = \frac{TP}{TP + FN} \qquad (10)$$

The common evaluation indicator is the F1 score because the indicators of P and R can occasionally be inconsistent. The weighted average of P and R, which is the F1 score, can be calculated using the following formula: 2PR

$$F1\ score = \frac{2PR}{(P + R)} \qquad (11)$$

(2) Time for calculation. The IDS detection algorithm's first computation time. The proposed method's detection time is shown in Figure 1.

Table.4. Recognition metrics

| | Actual: Yes | Actual: No |
|---|---|---|
| **Predicted: Yes** | True positive (TP) | False positive (FP) |
| **Predicted: No** | False Negative (FN) | True Negative (TN) |

### *3.6. Experimental setup*

The experiment was conducted using the MATLAB program 2021 version and the MATLAB Library tools for feature selection and classification.

## IV. Result And Discussion

The dataset includes 25,192 instances of 41 characteristics, as well as four additional attack types, including Probing, DoS, R2L, and U2R, in addition to the basic type of class label, in this experiment. A well-known strategy for building any classification system, 10-fold cross validation was used to carry out all of the tests since it reduces the likelihood of creating an over-fitted classification model.

### *4.1. All-around viability of the proposed method*

The performance of the proposed model was evaluated in two stages of the experiment. Testing of model performance was carried out on both the unbalanced and balanced datasets; the model's performance was judged on both occasions. The performance of the suggested model is shown in Table 5. The model had a 1.6 percent false positive rate overall and a true positive rate of 98.4 percent. With a true positive rate of 98.5 percent against U2R and 95.6 percent against R2L, respectively, the model performs best against U2R. The results are shown as a graph in Fig. 3.

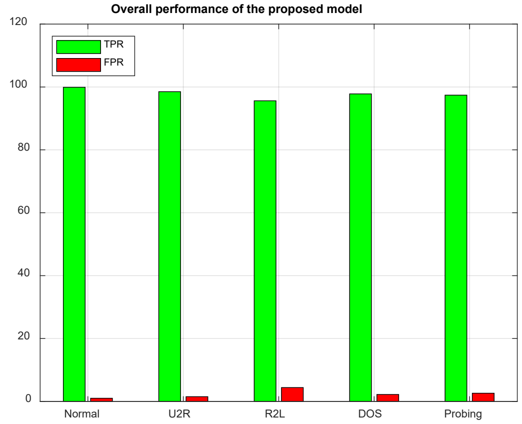

Fig.3. General performance of the presented method.

Table.5. Evaluate performance the proposed algorithm.

| Class | True Positive Rate (TPR) (%) | False Positive Rate (FPR) (%) |
|---|---|---|
| **Normal** | 99.9 | 0.1 |

| | | |
|---|---|---|
| **U2R** | 98.5 | 1.5 |
| **R2L** | 95.6 | 4.4 |
| **DoS** | 97.8 | 2.2 |
| **Probing** | 97.4 | 2.6 |
| **Average Weight** | 98.4 | 1.6 |

### *4.2. Examining how this algorithm compares to others*

Table 6 of the analysis shows the comparison of the proposed model to other models. With a 98.27 percent accuracy, the recommended model outperformed the other algorithms, including the Bayesian Network (95.99 percent), J48 (86.76 percent), and SMO (96.43 percent). Table 6's data is shown in Fig. 4 in a visual representation. Table 6 of the analysis shows the comparison of the proposed model to other models. With a 98.27 percent accuracy, the recommended model outperformed the other algorithms, including the Bayesian Network (95.99 percent), J48 (86.76 percent), and SMO (96.43 percent). Table 6's data is shown in Fig. 4 in a visual representation.

### *4.3. Wrapper method and other techniques for selecting features proposed*

A wrapper strategy is compared to other feature selection methods in Table 7 of the paper. CFS and consistency method strategies are both outperformed by the wrapper strategy, which selects 8 of the 41 available attributes and has an accuracy rating of 98.27 percent. With the help of a rank search, the consistency attribute selection approach achieved a 93% accuracy rate. CFS type filter's 91.13 percent accuracy was much less accurate than the proposed wrapper strategy's 99.97 percent accuracy (see Tables 7 and 8).

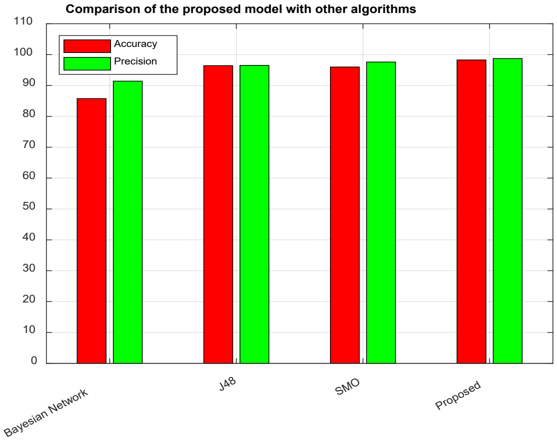

Fig.4. Comparison between the proposed method and other algorithms.

Table.6. Comparison between the proposed and other methods

| Algorithms | Bayesian network | J48 | SMO | Presented Method |
|---|---|---|---|---|
| **Acc** | 0.8576 | 0.9643 | 0.9599 | 0.9827 |
| **Precision** | 0.9140 | 0.965 | 0.976 | 0.9873 |

### *4.4. The comparison of the dataset's total features with the dataset's selected features*

Using both the entire dataset and simply the 8 features chosen by the suggested strategy, the performance accuracy of the suggested model. The dataset with the eight features from the proposed method has a greater performance accuracy of 98.27% when compared to the entire performance, as shown visually in Figs. 5 and 6.

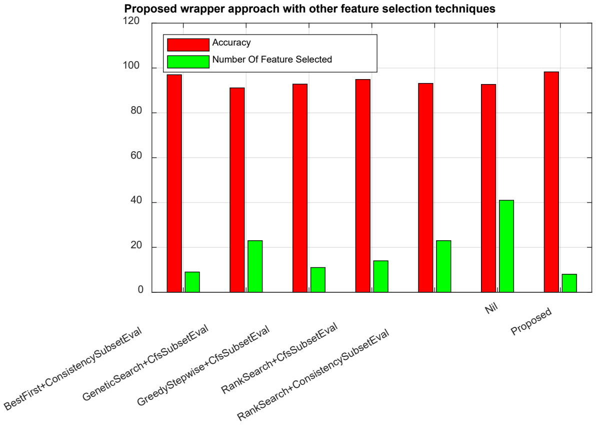

Fig.5. Accuracy comparison between proposed method and other feature selection methods.

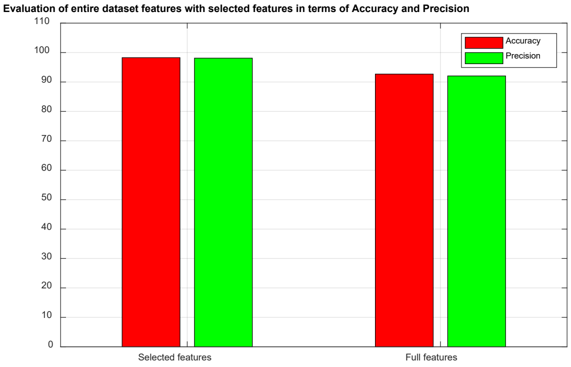

Fig.6. Comparison between of whole dataset features and selected features based on Accuracy and Precision.

Table.7. The proposed wrapper approach is compared to previous feature selection methods.

| Feature Selection Methods | Number of Selected Feature | Acc |
|---|---|---|
| **BestFirst+ ConsistencySubsetEval** | 9 | 0.9699 |
| **GeneticSearch+ CfsSubsetEval** | 23 | 0.9113 |
| **GreedyStepwise+ CfsSubsetEval** | 11 | 0.9281 |
| **RankSearch+ CfsSubsetEval** | 14 | 0.9488 |
| **RankSearch+ ConsistencySubsetEval** | 26 | 0.9313 |
| **Nil** | 41 | 0.9268 |
| **Presented Method** | 8 | 0.9827 |

### *4.5. The comparison of the suggested model's results with those of similar studies*

When compared to the accuracy findings from other relevant studies, as shown in Table 9, this study project performs better. The proposed method outperforms other methods in terms of the F-score parameter, according to the findings shown in Table 10.

Table.8. Comparison between of whole dataset features and selected features based on Accuracy and Precision

| NSL-KDD Dataset | Acc | Precision |
|---|---|---|
| **All Features** | 0.9268 | 0.9205 |
| **8 Selected Features** | 0.9827 | 0.9810 |

Table.9. Evaluation of the suggested method using information of relevant studies

| Papers | Acc | FPR |
|---|---|---|
| **Xingshuo et al. [19]** | 0.9807 | - |
| **Farhoud et al. [20]** | 0.9623 | 0.351 |
| **Proposed Method** | 0.9827 | 0.018 |

## V. Conclusion

PSO and Naive Bayesian methods are used to provide a unique technique to network intrusion detection in a fog computing environment. The proposed solution is based on the Nave Bayes Classifier and a wrapper approach to feature selection. This dataset is originally introduced with eight of the original 41 attributes being replaced by new ones. The Nave Bayes classifier was then used to sort the data. False positive rate (FPR) was 1.6% for the suggested model, with 98.27% accuracy. Compared to existing classifiers, the proposed model's output appears to be more accurate and efficient. In contrast to SVM, the F-scores of the Random Forest and Decision Tree algorithms are significantly higher. The wrapper technique, when used with the right attributes, is excellent at detecting anomalous intrusions.

Table.10. Comparison between presented and other methods based on F-score criteria

| Attacks | SVM | Random forest | Decision Tree | Proposed model |
|---|---|---|---|---|
| **Normal** | 0.93 | 0.99 | 0.99 | 0.99 |
| **DOS** | 0.96 | 0.99 | 0.99 | 0.99 |
| **R2L** | 0.40 | 0.94 | 0.94 | 0.98 |
| **probe** | 0.88 | 0.99 | 0.99 | 0.96 |
| **U2R** | 0.61 | 0.85 | 0.79 | 0.79 |

## REFERENCES

[1]. Evans, D., The Internet of Things: How the Next Evolution of the Internet Is Changing Everything, Cisco White Paper, 2011.

[2]. Kalnoor G, Gowrishankar S. IoT-based smart environment using intelligent intrusion detection system. Soft Computing. 2021 Sep;25(17):11573-88.

[3]. Etherington, D. and Conger, K., Large DDoS attacks cause outages at Twitter, Spotify, and other sites, TechCrunch, 2016.

[4]. Liang C, Shanmugam B, Azam S, Karim A, Islam A, Zamani M, Kavianpour S, Idris NB. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics. 2020 Jul 10;9(7):1120.

[5]. Stojmenovic, I., Fog computing: A cloud to the ground support for smart things and machine-to-machine networks, 2014 Australas. Telecommun. Networks Appl. Conf. ATNAC 2014, 2015, pp. 117–122.

[6]. Verma A, Ranga V. ELNIDS: Ensemble learning based network intrusion detection system for RPL based Internet of Things. In2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU) 2019 Apr 18 (pp. 1-6). IEEE.

[7]. Pahl, M.O. and Aubet, F.X., All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection, 14th Int. Conf. Netw. Serv. Manag. CNSM 2018 Work. 1st Int. Work. High-Precision Networks Oper. Control. HiPNet 2018 1st Work. Segm. Routing Serv. Funct. Chain. SR+SFC 2, 2018, pp. 72–80.

[8]. Tyagi H, Kumar R. Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches. Rev. d'Intelligence Artif.. 2021 Feb;35(1):11-21.

[9]. Pajouh, H.H., Javidan, R., Khayami, R., Dehghantanha, A., and Choo, K.K.R., A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE Trans. Emerg. Top. Comput., 2019, vol. 7, no. 2, pp. 314–323.

[10]. Tharewal S, Ashfaque MW, Banu SS, Uma P, Hassen SM, Shabaz M. Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. Wireless Communications and Mobile Computing. 2022 Mar 7;2022.

[11]. Deng, L., Li, D., Yao, X., Cox, D., and Wang, H., Mobile network intrusion detection for IoT system based on transfer learning algorithm, Cluster Comput., 2019, vol. 22, pp. 9889–9904.

[12]. Kumar P, Gupta GP, Tripathi R. Design of anomaly-based intrusion detection system using fog computing for IoT network. Automatic Control and Computer Sciences. 2021 Mar;55(2):137-47.

[13]. Liu, X., Liu, Y., Liu, A., and Yang, L.T., Defending ON-OFF attacks using light probing messages in smart sensors for industrial communication systems, IEEE Trans. Ind. Inf., 2018, vol. 14, no. 9, pp. 3801–3811.

[14]. Masood Z, Samar R, Raja MA. Design of fractional order epidemic model for future generation tiny hardware implants. Future Generation Computer Systems. 2020 May 1;106:43-54.

[15]. Anthi, E., Williams, L., and Burnap, P., Pulse: An adaptive intrusion detection for the internet of things, IET Conf. Publ., 2018, vol. 2018, no. CP740.

[16]. Sarvari S, Sani NF, Hanapi ZM, Abdullah MT. An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. IEEE Access. 2020 Apr 7;8:70651-63.

[17]. Mehmood, A., Mukherjee, M., Ahmed, S.H., Song, H., and Malik, K.M., NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks, J. Supercomput., 2018, vol. 74, no. 10, pp. 5156–5170.

[18]. Wibowo D, Novia M, Rumaksi RN, Gunawan SI. High Accuracy Detection of Covid-19 Based on Naive Bayes Classifier (NBC). InProceeding of The Symposium on Data Science (SDS) 2021 (Vol. 1, pp. 54-61).

[19]. Xingshuo, A., Xianwei, Z., Xing, L., Fuhong, L., & Lei, Y. (2018). Sample selected extreme learning machine based intrusion detection in fog computing. Wireless Communications and Mobile Computing, 1–10.

[20]. Farhoud, H., Payam, V. A., Juha, P., Timo, H., & Hannu, T. (2016). An intrusion detection system for fog computing and iot based logistic systems using a smart data approach. International Journal of Digital Content Technology and Its Applications

[21]. Asgarnezhad, R., Monadjemi, S. A., & Soltanaghaei, M. (2021). An application of MOGW optimization for feature selection in text classification. The Journal of Supercomputing, 77(6), 5806-5839.

[22]. Asgarnezhad, R., Monadjemi, S. A., & Aghaei, M. S. (2022). A new hierarchy framework for feature engineering through multi-objective evolutionary algorithm in text classification. Concurrency and Computation: Practice and Experience, 34(3), e6594.