Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7143054, 11 pages
https://doi.org/10.1155/2022/7143054

Research Article

Analyzing IoT Attack Feature Association with Threat Actors

Muhammad Shafiq,[1] **Zhaoquan Gu**,[1] **Shah Nazir**,[2] **and Rahul Yadav**[3]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China 510006*
[2]*Department of Computer Science University of Swabi, Pakistan*
[3]*Peng Cheng Laboratory, Shenzhen, China*

Correspondence should be addressed to Muhammad Shafiq; srsshafiq@gmail.com

Received 14 January 2022; Revised 16 April 2022; Accepted 22 April 2022; Published 5 May 2022

Academic Editor: Barbara Guidi

Internet of Things (IoT) refers to the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data. These devices can be controlled remotely, which makes them susceptible to exploitation or even takeover by an attacker. The lack of security features on many IoT devices makes them easy to access confidential information, issue commands from a distance, or even use the compromised device as part of a DDoS attack against another network. Feature selection is an important part of problem formulation in machine learning. To overcome the above problems, this paper proposes a novel feature selection framework RFS for IoT attack detection using machine learning (ML) techniques. The RFS is based on the concept of effective feature selection and consists of three main stages: feature selection, modeling, and attacks detection. For feature selection, three different models are proposed. Based on these approaches, three different algorithms are proposed. A set of 40 features was included in the model, derived from combinatorial optimization and statistical analysis methods. Our experimental study shows that the proposed frame work significantly improves over state-of-the-art cyberattacks techniques for time series data with outliers.

1. Introduction

Internet of Things is one of the most important trends in modern information technology. We expect that everything around us will soon be smart and connected, with different sensors generating vast amounts of data every second. Usually, data analysis is performed to extract useful information from it. Data can provide us with intelligence to make the right decisions or confirm our assumptions about the current situation. However, sometimes, malicious users use IoT devices as an attack vector because it allows them to turn smart objects into vulnerable targets, which can be used for malicious purposes. That is why detecting possible threats embedded in data generated by smart objects is crucial before any attack occurs. Hence, feature selection plays a vital role in improving the security level of monitored devices. It helps minimize false-positive rates while retaining valuable features included in the model that can predict possible danger scenarios based on common patterns associated with existing threats. Feature selection enables us to improve the detector efficiency, even if we decide not to use the implemented algorithm completely.

However, only certain parts (features) are needed; it provides flexibility and robustness to build detectors that can adapt behavior depending on environmental conditions. However, there is no one-size-fits-all method of detection that works equally well for all problems. Proper methods of feature evaluation play a significant role during the modeling process.

The process of identifying whether a network attack has taken place is to identify the underlying anomaly behavior. The anomaly detection algorithms are used to identify the abnormal activity in the network. However, it is not an easy task to find the right features for detecting anomalies in the datasets.

In machine learning, data can be noisy, and so we need to do some kind of feature selection to remove noise. If your data has a lot of noise, then you can have a decrease in

training performance. Different methods used for feature selection include manual selection, correlation coefficients (manual), principal component analysis (PCA), and mutual information (MINS criterion) to name a few. An example of feature space using automated methods is discussed below.

Therefore, there are different features selection algorithms that have been proposed for network attack detection problems. Cohen et al. [1] proposed a feature selection algorithm based on the mutual information metric (MIM). MIM computes the mutual information between each pair of attributes in the collection and measures their importance in terms of their contribution towards anomaly detection in networks. The main drawback in this approach is that it does not consider redundancy in the dataset and thus does not select essential attributes in the dataset. Similarly, Safaei et al. [2] proposed a feature selection algorithm based on feature ranking for detecting anomalies in wireless sensor networks (WSN). This algorithm considers both redundancy and correlation attributed to datasets. However, it only considers multivariate data; it does not consider univariate data. Parvez et al. [3] proposed an improved approach for detecting attacks at the sensor level using support vector machines (SVM). SVM works based on statistical learning theory, which can be used to detect attacks. Feature engineering [4] is used to transform data into a format suitable for machine learning and artificial intelligence algorithms. Before evaluating machine learning models, it is essential first to understand how each feature impacts model performance. In some cases, a feature that has a positive impact [5] on a model might actually increase an end user's propensity to click on malicious content, for example, big data features. Feature selection provides a way to measure an individual feature's contribution as it relates to overall performance. The resulting features can be further analyzed individually or aggregated with other similar features. The most common methods for feature evaluation are information gain (IG) [6] and reduction in multicollinearity (VIF). Both measures involve identifying variables that maximize their respective objectives and then eliminating any variables that fail to meet a certain threshold. The basics of feature selection are to filter the input data and select those features that are most relevant to the outcomes. The selection is based on experimental results, and it can be used with almost any learning algorithm. Well-chosen features greatly improve the performance of machine learning, especially for complex learning algorithms. Feature selection is the process of determining which input variables to consider when carrying out a machine learning task. It helps improve overall performance, and it reduces overfitting by reducing dimensionality.

This study presents a novel framework RFS for feature selection and compares it with existing methods. For feature selection, three different models are proposed.

Based on these approaches, three different algorithms are proposed. The framework is implemented in a network attack detection system using the MLA method. The model is designed to detect both insider and external attacks on the IoT network. The proposed framework is evaluated using the Bot-IoT data from an operational distributed computer network. In addition, the proposed approach is compared with other methods on a set of attack detection data obtained from a commercial intrusion detection system. A series of experiments have been conducted to evaluate the performance of the proposed model. The results demonstrate that our method significantly outperforms existing methods in terms of accuracy and computational time while maintaining low false-positive ratios. However, the main contributions of this work are summarized as follows:

(i) In order to establish effective framework for IoT network attack detection, this paper firstly introduced the RFS framework definition, explained the MLA concepts, and proposed a novel integrated framework for IoT network attack detection. Also, the framework is capable of identifying any malicious threats in real time and classifying them into appropriate attack categories according to the prediction results. And at last, some test results show that this framework works correctly

(ii) For feature selection, three different models are proposed which are based on three different proposed algorithms

(iii) Based on the proposed approaches, OOF Algorithm is proposed to effectively filter and select the best feature set for detecting IoT attacks

(iv) The method has been evaluated on real-world datasets collected from the Internet, and the results show that our method can efficiently detect various kinds of IoT network attacks without false alarms

(v) The remainder of this article is organized as follows. In Section 2, the related work is presented. In Section 3, we describe the methodology of our proposed work. In Section 4, we present and discuss the main results of our study. In Section 5, we discussed analysis and observations, and in Section 6, we conclude the study

2. Related Works

With our rapidly evolving world and always-on connectivity, it should come as no surprise that the Internet of Things (IoT) has become an integral part of today's world. The number of connected devices continues to rise and shows no signs of slowing down. IoT security is a constantly growing problem because more and more devices are added every day, and finding effective ways to protect these devices can be challenging. A major part of protection is making sure that only authorized people can access these devices; however, since not everything gets tested before deployment, unauthorized individuals will eventually get access anyway through various mechanisms such as exploiting vulnerabilities or using default credentials to gain administrative rights over these things. These privileged users have pretty much free reign to attack whatever they want in terms of data storage, identity theft, DDoS attacks, spamming, etc. Due to DDoS attacks, this inevitably puts enterprises in danger by

losing confidential information or providing services under extortionate demands. That is why there is so much emphasis on looking at how these things are protected when being accessed remotely by paying close attention to what each thing does during operation within their respective networks. Historically, what was important with servers/PCs/smartphones/tablets could now prove just as valuable for monitoring IoT devices in real time when put into practice correctly.

In 2020, Alqahtani et al. in [7] studied and proposed an IoT botnet attack detection approach based on optimized extreme gradient boosting and feature selection. In this approach, each node collects all network traffic from its ports and transfers it to the core for botnet detection. Some optimization techniques are applied in the core to reduce core computational load for detecting botnet attacks in port devices using real-time kernel function. This optimization helps core to detect botnet attacks in a short period of time without transferring a large amount of data from port devices to core. For modeling, boosting is used in core to detect outlier-type attacks from the normal network traffic. An effective feature selection method is proposed to reduce core computational load by removing features that do not affect prediction model performance. In the experiment result, they showed that their presented approach is efficient for detecting Bot-IoT attack traffics.

Recently, Awotunde et al. in [8] proposed a novel approach to support intrusion detection in industrial IoT networks, built on a deep learning model with rule-based feature selection. In this research, the impact of IoT on security is highlighted, and a review of state-of-the-art intrusion detection systems is made. Then, they present a neural network model for detecting intrusions through analysis of IoT data flow characteristics. It builds a system to examine IoT data flow characteristics using the framework of flow graphs matrices over computer networks to detect anomalies in the time delay between entries and exits from the nodes and their cycles.

Similarly, the author in [9] proposes an efficient and scalable feature selection approach that selects features based on user-defined interests to perform traffic classification. The proposed approach was aimed at achieving high accuracy in traffic classification by selecting only relevant features. It is much faster than other algorithms, which require multiple iterations before the optimal solution is achieved while achieving better results. Similarly, in our previous work [10–12], we develop a feature selection mechanism for IoT malicious traffic identification and then empirically evaluate the proposed approach in terms of its performance when classifying positive and negative samples. Then, we discuss some of the challenges that need to be overcome in operating and managing such a system. Similarly, the work in [13] provides an effective approach to daily activity feature selection by understanding the correlation between daily activities; the estimated correlations are used to compute the importance of selected features. Thus, it can effectively reduce the computational complexity to study the performance of representative subnetworks in smart homes. More in-depth in [14] put forward an extended feature set that can be used to detect IoT device attacks. This technique leverages the fact that the majority of whale-attacking IoT devices are built on Arduino platforms. The key to the approach described in this paper is the replacement of inner cores with Arduino boards.

However, studying the above literature review, it is concluded that the feature selection method still needs to be studied in depth to identify cyberattack detection in IoT networks. For this purpose, in the next section, we proposed our RFS framework, which is basically included in several phases. However, the detailed proposed framework is discussed in the next section.

3. Proposed Framework Methodology

The Internet of Things (IoT) is growing rapidly and is predicted to be one of the fastest growing technologies in the upcoming years. With a continually growing number of connected devices, IoT makes it possible to streamline our daily lives in a whole new way. However, Internet of Things can leave its users vulnerable to attacks. As such, it is important that security must not be overlooked during development when creating an IoT project or device. The number of connected devices is increasing, enabling us to make more convenient and efficient use of sensor data. However, with greater connectivity comes new security risks. There are ways in which hackers can exploit connected devices. Due to their presence on networks, IoT devices can be used as a stepping stone or pivot point into other connected systems (e.g., corporate networks). It is necessary to use effective methods for detecting these kinds of targeted cyberattacks on organizations that possess large-scale deployments of sensors or smart objects that would be difficult or time-consuming to secure using conventional approaches properly. Similarly, a feature selection algorithm is one of the main tuning parameters in machine learning models such as support vector machines, artificial neural networks, and regression models. The criteria used to select the best subset of features depend on the type of model used. This paper proposes different feature selection algorithms to choose traits that are valued for a single or multiobjective optimization for feature selection. It is also shown that the best subset of features can vary with machine learning models according to the use case.

However, this paper presents a novel feature selection method for IoT attack detection. Our proposed RFS (robust feature selection) framework consists of three phases: feature engineering, identification, and results analysis. However, the feature engineering phase consists of three different methods, in which we proposed three different updated models and algorithms for the robust feature selection. The proposed framework RFS and their submodels are illustrated in Figure 1. However, it is important to discuss the challenges before going through in depth; the various challenges of implementing the proposed framework are significant. For example, the privacy and security dimensions raise issues not only for communication between the devices but also for requiring that the cloud is sufficiently secure to handle the data produced by IoT devices. The challenge of

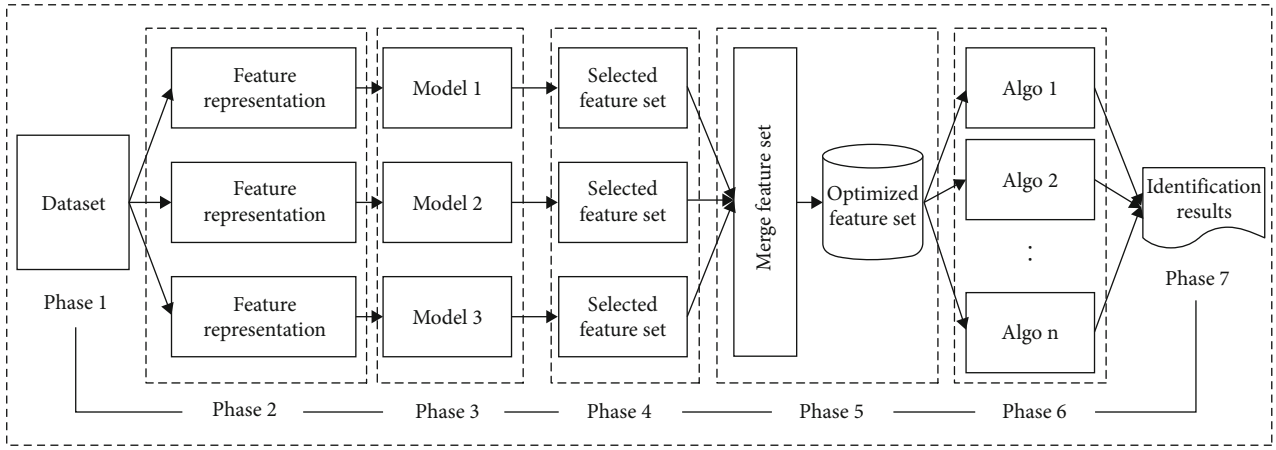

Figure 1: Proposed framework for IoT attack detection.

connecting devices to either Wi-Fi or mobile broadband (that can be more expensive than wired connections) is a further issue that needs consideration. Overall, IoT demonstrates the incredibly difficult task of developing sufficient standards in a relatively short period considering that it is still evolving. However, for this aim, Bot-IoT dataset is used in this research work.

3.1. Proposed Method. As discussed in the previous section, IoT cyberattack detection is a recent hot topic, and several researchers in the research community try to overcome the problem. To tickle the problem in this study work, we proposed robust feature selection (RFS) framework as shown in Figure 1. Our proposed framework consists of seven phases, but in phase 3, we proposed three more updated models and algorithms for the feature selection discussed in the following subsection. In the first phase, a dataset is collected and prepared, and then in phase 2, features are extracted and normalized. Then in phase 3, which is the main phase of our proposed framework, three different updated models with the algorithm are proposed for the effective feature selection. Then in phase 4, selected features are presented separately, and then in phase 5, the selected features are merged and optimized. Similarly, in phase 6, the different effective ML classifiers are applied with the effective feature set, and phase 7 presents the identification and verification of the proposed framework. As discussed, the primary phase is phase 3, which is our main proposed method in our proposed framework and is discussed in detail in the following subsection.

3.1.1. Updated Delphi Method. The Delphi method [15, 16] is a structured process used to reach consensus and produce forecasts. Proponents claim that it can increase the accuracy and reliability of forecasting and aid in planning and decision making. It has been applied to areas such as predicting the outcomes of elections, estimating future demand for products, determining the likelihood of conflict between political parties, forecasting evolutionary changes in various fields, developing projective techniques in psychotherapy, appraising environmental risks, investigating the possible impacts of job automation on employment, and helping communities to deal with decisions involving high dependence on an oil economy better than traditional forecasting methods. The Delphi technique is a structured communication technique that seeks input from a panel of experts to respond to a question. It is a structured communication technique in which a panel of experts is asked to provide anonymous feedback on questions asked by an organization. The term "Delphi method" often encompasses the broader concept of any type of structured communication technique in which respondents provide multiple-choice or true-false feedback, generally in the form of their opinions, attitudes, or predictions about some topic, generally after some preparation time during which they may study or reflect on the issue. The five-step Delphi method involves (1) identifying the problem, (2) gathering experts to get their opinion, (3) discussing and prioritizing ideas, (4) developing a demographic questionnaire emailed to these experts, and (5) using the results of the survey to develop a solution. However, in this study, the Delphi method is adopted with our proposed algorithms to effectively select feature set to identify IoT attack detection, as shown in Figure 2. In this proposed method, the control initially starts from the feature. The Delphi method is applied to the feature set, and then, the control passes to the proposed algorithm in which the effective feature is selected and normalized. Finally, effective feature set contains the feature that passes by the Delphi method and proposed algorithm. More in depth, for the Delphi method process, three different published papers are initially selected [10, 12, 17]. The published paper feature is selected for the updated Delphi method with the proposed algorithm model. Based on the model feature selected for the model and then after applying the model, effective features are generated and passed to the main RFS framework.

3.1.2. Proposed OOF Algorithm. The aim of the research is to develop a method for feature selection. Studies show that algorithms can be useful not only for reducing the number of features used in the predictive model and improve the accuracy and time of processing but also for increasing the

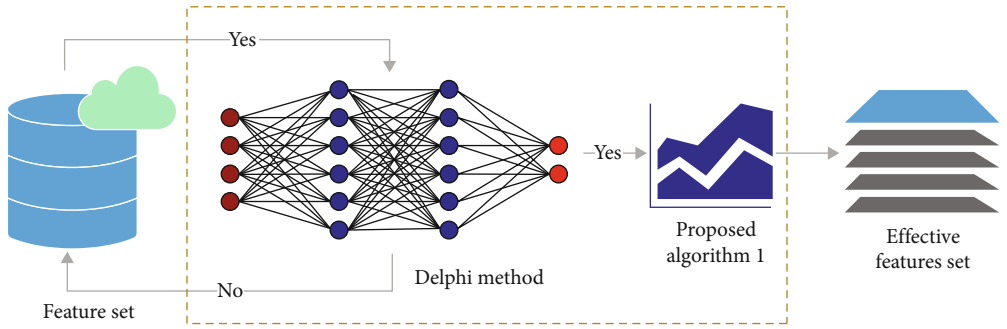

Figure 2: Proposed updated Delphi method with the proposed algorithm.

1. begin
2. F_p = getFirstFeatures (*list*);
3. End until (Fp==Null);
4. Last_*M123*←*classify X;*
5. Insert the *feature* into $R_{wrapper}$;
6. *Feature* = getNextFeatures;
7. **for** feature is not **Null**
8. insert the feature into $R_{wrapper}$;
9. X is a dataset of sample values for $R_{wrapper}$;
10. **if**(δ< = *last_FeatureValue*)
11. Remove features from $R_{wrapper}$;
12. **else**
13. *Feature* = getNextFeatures(*list*, *feature*);
14. **end if**
15. **end for**

Reture $R_{wrapper}$;

Algorithm 1: Optimization of features based on Delphi method.

performance and quality of data mining and other applications. Our proposed OOF algorithm is based on the feature set selected by the Delphi method. The algorithms consist of fifteen steps. Initially, the algorithm takes the feature as input and classifies and filters it with the Delphi method and then passes the control to the next step to make a list of features with an effective feature set. Then using the wrapper technique, the algorithm filters the feature with the threshold value to filter all the feature and remove the redundant feature from the list. Afterward, the algorithm put forward an effective feature set and return the algorithm as shown in Algorithm 1.

3.1.3. Information Gain Technique. Feature selection is an important part of data mining. Feature selection or feature extraction is concerned with identifying a subset of relevant features to use in predictive models rather than attempting to model all features present in training data. Feature engineering has also been defined as the process of constructing new variables which are derived from one or more existing ones by means of various methods (regression, rule induction, neural networks, etc.). Although there are many ways to accomplish feature selection, it is usually an essential part of almost every type of machine learning algorithm and model since it helps choose which attributes are relevant for each specific problem under study.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i), \tag{1}$$

$$H(Y \mid X) = \sum_{x \in X} p(x) H(Y \mid X = x), \tag{2}$$

$$IG(Y \mid X) = H(Y) - H(Y \mid X). \tag{3}$$

Information gain [18] is one of the most commonly used techniques in feature selection. Information gain uses reduction in entropy to determine the relevancy of features in a dataset. It relies on two basic assumptions: (1) the amount of information contained in a feature with respect to a given target function increases with relevance. (2) The relative importance of features between each other is symmetric with respect to their contribution towards identifying relevant patterns in data. Information gain decomposes an objective function into two components—information content and

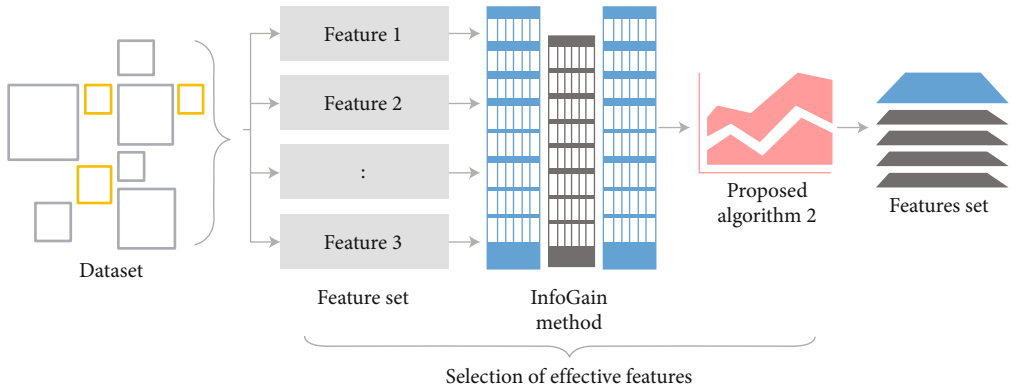

Figure 3: Proposed updated InfoGain model.

redundancy. Information content measures how much information is contained within a feature, while redundancy measures the degree to which different features are providing different values collectively. The goal of information gain-based feature selection is to find out all redundant features and select only those that offer maximum information per bit (for example, chi-square score). A measure called "information gain" along with chi-square X^2 statistic helps in achieving that. This is done by minimizing chi-square statistics or maximizing information gain. This step must be performed prior to further classification/clustering/pattern detection methods as it reduces dimensionality problems found in high dimensional spaces by removing irrelevant predictors/features that contribute little or no additional value through increased statistical significance or predictive power, thus reducing the dimensionality of data. All information-theoretic feature selection techniques work with frequency distributions; i.e., discrete or continuous-valued variables can be represented using histograms where there are multiple bins corresponding to various possible values of each variable. Any histogram corresponds to a set of underlying probability distribution parameters, namely, mean U and standard deviation δ representing center or centre-of-mass and spread, respectively.

Formally, let $P(x\,u, \delta)$ denote a discrete probability distribution whose cumulative density function is given by $P(x\,u, \delta)$, where x denotes state vector consisting of n numeric values assumed uniformly distributed over the range $[0..n]$. In real-world situations, having uncertainty about u leads to concerns about accuracy as well as the robustness of any classifier developed. Thus, for the best feature selection in this study, we applied information gain feature selection technique with proposed new algorithm for the IoT attack detection. Information gain technique is based on entropy, and to calculate the information gain initially, entropy is calculated as illustrated in Equation (1). Then based on entropy, conditional entropy is calculated as in Equation (2), similar to Equation (3) in information gain.

3.1.4. Proposed InfoGainPlus. Algorithm: as discussed in the previous section, feature selection is an important part of the ML algorithm for effective model creation. For this purpose, in this research study, different updated models are presented for the effective feature. Thus in this section, InfoGainPlus algorithm is proposed based on the InfoGain algorithm. Our proposed algorithm is based on the features set that selected by the ReliefF algorithm. The algorithms consist of several different phases. However, the process of this proposed algorithm is the same as shown in Algorithm 1. But in the initial step, the Delphi method was replaced into InfoGain algorithm. The rest of the algorithm steps are the same steps that we followed for the filtration and optimization for the feature selection.

3.1.5. Decision Tree Method. C4.5 is a "classification tree" algorithm [19], which is a type of machine learning algorithm that constructs a classification or regression tree used to predict the value of a target variable. The C4.5 algorithm uses information gain to select attributes that are most helpful for predicting the target variable and recursive partitioning to recursively split off subtrees until each subtree has only one instance. C4.5 is an old algorithm, roughly equivalent to C5.0, although faster on some datasets (and slower on others). C4.5 can be used to create rules for classification (or regression) by selecting only the attributes that are helpful for prediction and discarding all others; it can also be used to create classifiers that do not use any attribute selection (but cannot be run as efficiently) [20]. The C4.5 algorithm was invented by J Ross Quinlan in 1986 and was first described in his Ph.D. thesis in 1987 at the University of Waikato in New Zealand. It has been widely used since then; Quinlan estimated in 1995 that 50% of all production systems use some form of decision tree learning, with C4.5 being the most popular implementation. The detail implementation of C4.5 algorithm is shown in Equations (1)–(3), where $p\,(x\_i \mid (y\_j$ and $p\,(x\_i)$ and $p\,(x\_j)$ are the joint probabilities and marginal probabilities, respectively. In this research study, we applied the C4.5 decision tree algorithm as a model for the selection of effective feature selection, as shown in Figure 3. The proposed model is very easy to apply. Initially, the feature set is supplied, and based on the C4.5 decision tree algorithm, effective features are selected, and then based on the C4.5 decision tree, new algorithm is proposed. The proposed algorithms initially filter all

the supplied features and then base the C4.5 decision tree effective set of feature returns as a result.

$$GR(X \mid Y) = \frac{(H(X) - H(X \mid Y))}{(H(X))}, \quad (4)$$

whereas

$$H(X \mid Y) = -\Sigma_j p\left(y_j\right) \Sigma_i p\left(x_i \mid \left(y_j \log_2 p\left(\left(x_i \mid \left(y_j\right)\right)\right)\right)\right). \quad (5)$$

Similarly,

$$H(X) = -\sum_{x_i} p(x_i) \log_2 p((x_i)). \quad (6)$$

3.1.6. Proposed EFS Algorithm. When building a machine learning model, one of the most important steps is to choose features that are relevant to the problem. If there are limited data, feature selection acts as a filter to select the most relevant features. The goal of feature selection is to identify a subset of explanatory variables that are most predictive of the outcome variable. By doing so, we can build an accurate machine learning model with less effort by avoiding irrelevant or redundant features. For this purpose, we proposed a new algorithm named EFS, which filters the entire supplied feature based on C4.5 ML algorithms and then uses threshold value to filter and remove redundant features that are not effective or do not give enough information for the identification of attacks in IoT network environment. The proposed algorithm acts the same as shown in Algorithm 1. Initially, the algorithm followed the C4.5 ML algorithm to filter and rank the feature supplied. Then, in the second phase, the algorithm ranks those features that give enough information for identification.

4. Evaluation Methodology

To evaluate our proposed framework RFS, we performed experiments on a real-world testbed dataset. The IoT platforms used in experiments are popularly available and have been widely used by other researchers. However, the dataset and result analysis are explained in the following.

4.1. Dataset Selection. In this research study, the Bot-IoT dataset is used to evaluate our proposed framework, which is publicly available online. The Bot-IoT dataset provides a large, representative collection of real-world attack traces, labelled to enable effective training and evaluation of IoT attack detection algorithms. The Bot-IoT dataset covers network layer DDoS attacks on DVRs, IP, cameras, security cameras, smart refrigerators, televisions, and other embedded devices. Their bots are positioned in various geographical regions across multiple network providers. These bots generate traffic to remote devices with randomly generated user agent strings so that different bots generate traffic from different sources. Some bots also spoof their source IP addresses by randomly choosing IP addresses belonging to other service providers or bot-generated addresses, making it difficult to identify them based on source addresses alone. The data contains 100 K observations from each device type as well as an activity log for each device. To ensure that the dataset is comprehensive, they instrumented ten popular IoT devices listed below: we also added a wormhole router as part of every experiment run to ensure all data coming from our bots reaches all our honeypots.

4.2. Performance Measurements. For the performance measurements of our proposed framework, accuracy, recall, and precision will be used to measure how well a model is able to predict. Accuracy of the classifier is a widely accepted way of measuring performance in many ML problems. High classification accuracy is achieved when most predictions made by a classifier are correct. While creating ML models, we should focus on achieving a high accuracy level without creating many false-positive/false-negative predictions. In other words, it measures from what fraction of data points incorrectly classified as positive does it correctly identify those that belong to positive classes. It is just another name for accuracy but only deals with positive classes. To determine how well a model does in identifying an attack, use accuracy, recall, and precision metrics. The usual metric to evaluate classifier performance is accuracy which is defined as number of correctly predicted instances divided by total number of instances. If we have a binary classification problem positive value to a positive instance, while precision measures how often a classifier assigns a positive value to a negative instance.

However, below are the measurement metrics that we used in this research work.

4.2.1. Accuracy. Accuracy is a statistical measure of how close results are to being correct. This term is used in classification problems, where instances have been labeled correctly or incorrectly by some criterion. When determining which algorithms are best to use to solve a problem, if the goal is to maximize accuracy, then select algorithms that maximize accuracy. If the goal is a high level of class purity meaning only one label, then choose an algorithm with a low level of impurity meaning it rarely classifies something as something else rather than one with high Accuracy. However, in this study, Equation (1) is used for the accuracy metric.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}. \quad (7)$$

4.2.2. Precision. Precision is a metric that measures how good a model's predictions are at identifying relevant objects. The higher precision score, then, is a measure of how well it can distinguish between relevant and irrelevant data points. In other words, precision gives an indication of a model's ability to provide accurate results. Precision ranges from 0% to 100%, with higher scores indicating better performance. A precision score of 100% means that every element in our dataset has been correctly identified by our algorithm. When building an algorithm, it should be designed in such a way that it produces no false positives,

i.e., identifying nonrelevant data points as relevant or vice versa, which would result in inaccurate outputs and very low precision scores. For precision calculation, Equation (8) is used.

$$\text{Precision} = \frac{TP}{(TP + FP)}. \tag{8}$$

4.2.3. Sensitivity. A machine learning algorithm's sensitivity is its ability to correctly identify positive examples in comparison to negative examples. For example, consider a simple binary classification problem that is trying to distinguish dogs from cats. The most sensitive classifier would return positive results for both dog and cat images with high confidence, whereas a least sensitive classifier would only give high confidence results for dogs and not cats (or vice versa). Similarly, we used Equation (9). for the sensitivity metric evaluation.

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}. \tag{9}$$

Then, recall and precision metrics are also used instead of accuracy. In binary classification, if we have two classes, then recall or sensitivity measures how often a classifier assigns *a*.

4.2.4. Specificity. In computer science, specificity is a measure of how unique a pattern is in a given set of data. A highly specific pattern will have few matches in a dataset, while a less specific pattern will have many. Given a machine learning problem, finding an optimal set of parameters for training algorithms can be viewed as finding a local maximum in some function where specificity measures how far away that maximum is from other maxima. Thus to calculate the specificity metric, Equation (10) is used.

$$\text{Specificity} = \frac{TN}{(FP + TN)}. \tag{10}$$

5. Results and Analysis

This paper focuses on attack identification. It describes an analysis method that can use many IoT data sources to identify an attack. The proposed algorithms and framework have been applied to the Bot-IoT dataset [21], which consist of both benign and anomalous time series data examples. As a result, the report compares the performance of four applied algorithms with respect to feature selection and accuracy, precision, specificity analysis, and sensitivity analysis. The main goal of the proposed work is to propose different algorithms and framework with different feature selection techniques for IoT attack identification at large scale. Some feature selection method should be better than the rest in a specific problem; hence, it is crucial to find out what the best feature selection procedure is for a particular classification problem. We have to pay special attention to test many and compare results of these procedures in order to know what feature selection is the best one for our problem. Two highly similar procedures can give very different results with respect to accuracy, precision, sensitivity, and specificity.

However, for this purpose, in this research study, a new framework RFS is proposed for the effective feature selection and cyberattack detection in Internet of Things network environment. For the evaluation, Bot-IoT dataset is used with four different ML classifiers with dour different measurement metrics as shown in Figure 4. The chosen feature subset follows the methods used in the best practices to filter features from the initial attributes set. Each feature that joined in the feature subset is necessary and sufficient for its position in the ranking average accuracy score. It also helps us to predict with high confidence that it passes all the statistical test (accuracy, precision, sensitivity, and specificity). Those characteristics hold according to our dataset of datasets. In addition, since these are machine learning models, we are trying to improve performance on each iterative algorithm process.

However, all the applied ML classifiers achieve very promising performance result for the identification cyberattacks in IoT network. However, Naïve Bayes ML algorithm for the SSR attacks identification is little low as compared to other applied ML classifiers for the SSR attacks identification. Similarly, the highest accuracy performance result achieved C4.5 decision tree algorithm for the detection of TCPDoS attacks. Figure 4 shows the accuracy of identifying different categories of cyberattacks in IoT networks using different machine learning algorithms. The best performance in identifying cyberattacks in IoT network is made by decision tree classifier (99%) which can identify all kinds of attacks with accuracy rate 99%. The best performance in identifying DDoS attacks is 99%, and it has been achieved by using decision tree classifier with 9% accuracy rate, respectively. Similarly, precision metric results are also very effective, and all the applied ML algorithm performances are very promising with respective precision metric. However, the overall performance of SVM algorithm for precision metric is little low as compared to other applied ML classifiers.

The precision performance of C4.5 and random forest ML algorithms for IoT cyberattack detection is higher compared to other ML algorithm. On the other hand, the result of sensitivity and specificity metrics are very promising for identification of IoT cyberattack detection using ML algorithms. However, for both metrics, SVM and ML classifiers achieve little low performance result compared to other ML classifiers. Similarly, Naïve Bayes algorithm performance is low in sensitivity metrics as compared to SVM as shown in Figure 5. However, it is evident that the specificity and sensitivity results of Naïve Bayes and SVM are low as compared to C4.5 and random forest algorithms. We also found that the accuracy of the C4.5 algorithm is higher than other algorithms. Thus, overall result of C4.5 algorithms is promising as compared to ML algorithm with metrics precision, sensitivity, and specificity. A support vector machine (SVM) and a Naïve Bayes classifier along with some other machine learning algorithms are used for detecting IoT attacks. We have investigated the performance of these

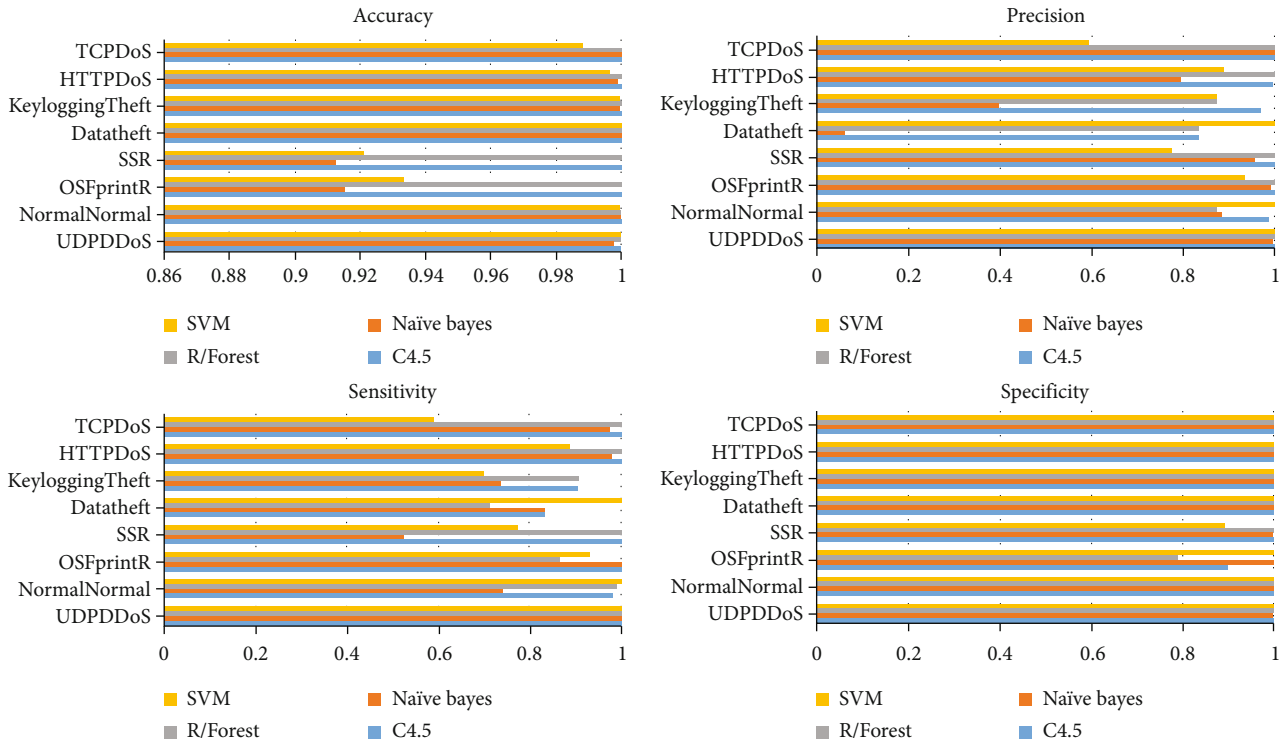

Figure 4: Proposed model results.

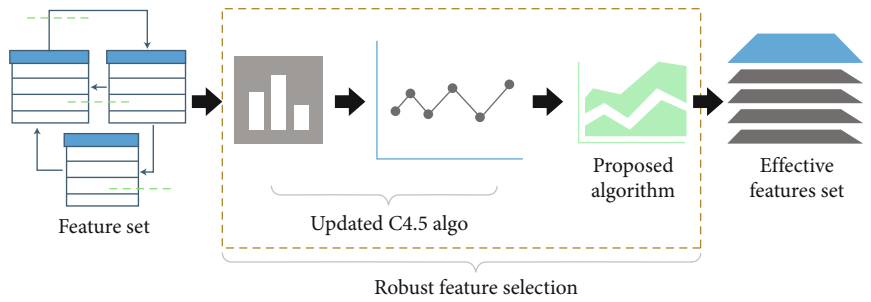

Figure 5: Robust feature selection model.

methods based on some sample datasets and have found that both of them are good enough to be used for detecting the attacks. The main advantage of using these algorithms is that they can be easily integrated with an existing system. However, there are some drawbacks also associated with these two algorithms. The SVM algorithm has been reported to have poor performance in case of imbalanced classes, while the Naïve Bayes' algorithm is not suitable for handling numeric attributes. Both of these algorithms require a large amount of data for training and testing phases. The result of Naïve Bayes and SVM algorithms is a little low as compared to C4.5 and random forest with respective sensitivity and specificity metrics for the identification of IoT attack detection. The effectiveness of the proposed methodology for IoT cyberattack detection is shown in the results presented in Figure 5. It clearly shows that when using the proposed framework, the total number of attacks detected is larger than when using traditional security tools. Also, it can be seen that when using our proposed framework, all three security breaches are detected, but when using only traditional security tools, two out of three security breaches are detected.

The results of this research show that the proposed framework is effective in detecting the IoT cyberattacks. The proposed framework can effectively detect attacks at all layers of IoT networks. From the performance evaluation, we can see that the detection accuracy is above 99%. Moreover, the execution time of our proposed method is not very high. Based on these results, we can conclude that the proposed framework for IoT cyberattack detection is effective and efficient.

We can conclude that the proposed model is effective for the detection of IoT attacks. The model is strong in identifying state changes and anomaly patterns, making it suitable for detecting unknown attacks. In order to leverage more information for authentication and authorization, we plan to extend the proposed detection model by adding more features based on IoT topology information and other valuable

information from the event log database. Our proposed model is effective for the detection of IoT attacks. A single model with anomaly score and confidence metric can provide an effective solution for identifying anomalies in IoTs. Anomaly detection is promising in preventing IoT attacks as most attacks cause spike of normal traffic pattern. For example, a worm infection will cause a very large number of devices to contact at the same time to one specific IP address. Meanwhile, malicious control commands will usually be issued periodically, which makes it possible to build a detection system based on crafted. With anomaly detection, the model scores and confidence metrics can help in analyzing and identifying suspicious objects/events by providing an empirical assessment of the model's correctness and trustworthiness.

6. Conclusion

The main goal of feature selection is to improve dataset efficiency, reducing computational time and memory requirements. This is achieved by discarding irrelevant or redundant features. Feature selection methods are mostly applied after completing some preliminary data analysis, that is, after identifying potential relevant variables based on domain knowledge, hypotheses, etc. A feature-screening step may be included as part of preliminary analysis, which ensures removal of features with noise effects (as opposed to real information) or those not contributing significantly to prediction performance.

Thus in this paper, we propose a novel feature selection framework RFS for IoT attack detection. For feature selection, three different models are proposed. Based on these approaches, three different algorithms are proposed. Our experimental study shows that the proposed framework significantly improves over state-of-the-art IoT attack detection techniques for time series data with outliers.

The proposed framework not only detects outliers and attacks but also predicts the IoT system's likely future performance, making it a unique tool for helping to prevent these attacks. The real-time attack detection function of our framework can help to reduce the impact of attack, which makes it an invaluable tool in protecting IoT systems. It can present high-impact findings to power users so that they know when there is a clear and present danger to their IoT platform.

Data Availability

The data that support the findings of this study are unavailable in any public repositories.

Conflicts of Interest

The authors declare that there are no conflict of interest.

Acknowledgments

This work is supported in part by the Guangdong Province Key Research and Development Plan (Grant No. 2019B010136003), the National Natural Science Foundation of China (61902082), the Guangdong Higher Education Innovation Group (2020KCXTD007), and the Guangzhou Higher Education Innovation Group (202032854).

References

[1] S. Cohen, E. Ruppin, and G. Dror, "Feature selection based on the Shapley value," *other words*, vol. 1, p. 98Eqr, 2005.

[2] M. Safaei, A. S. Ismail, H. Chizari et al., "Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive Bayesian-network-based approach," *Software: Practice and Experience*, vol. 50, no. 4, pp. 428–446, 2020.

[3] I. Parvez, M. Aghili, A. I. Sarwat, S. Rahman, and F. Alam, "Online power quality disturbance detection by support vector machine in smart meter," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 5, pp. 1328–1339, 2019.

[4] R. M. Swarna Priya, P. K. R. Maddikunta, M. Parimala et al., "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.

[5] A. Ali and C. Gravino, "Evaluating the impact of feature selection consistency in software prediction," *Science of Computer Programming*, vol. 213, article 102715, 2022.

[6] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning- based text categorization," *Information Processing & Management*, vol. 42, no. 1, pp. 155–165, 2006.

[7] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors*, vol. 20, no. 21, p. 6336, 2020.

[8] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion detection in industrial Internet of Things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, 17 pages, 2021.

[9] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri, "Toward an efficient and scalable feature selection approach for Internet traffic classification," *Computer Networks*, vol. 57, no. 9, pp. 2040–2057, 2013.

[10] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[11] M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, "Data mining and machine learning methods for sustainable smart cities traffic classification: A survey," *Sustainable Cities and Society*, vol. 60, article 102177, 2020.

[12] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Computers & Security*, vol. 94, article 101863, 2020.

[13] Y. Liu, Y. Mu, K. Chen, Y. Li, and J. Guo, "Daily activity feature selection in smart homes based on Pearson correlation coefficient," *Neural Processing Letters*, vol. 51, no. 2, pp. 1771–1787, 2020.

[14] M. Mafarja, A. A. Heidari, M. Habib, H. Faris, T. Thaher, and I. Aljarah, "Augmented whale feature selection for IoT attacks: structure, analysis and applications," *Future Generation Computer Systems*, vol. 112, pp. 18–40, 2020.

[15] A. T. Gumus, "valuation of hazardous waste transportation firms by using a two step fuzzy-AHP and TOPSIS methodology," *Expert Systems with Applications*, vol. 36, no. 2, pp. 4067–4074, 2009.

[16] H. A. Linstone and M. Turoff, *The Delphi Method*, Addison-Wesley, Reading, MA, 1975.

[17] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for Internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[18] J. T. Kent, "Information gain and a general measure of correlation," *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.

[19] H. Chauhan and A. Chauhan, "Implementation of decision tree algorithm c4. 5," *International Journal of Scientific and Research Publications*, vol. 3, no. 10, pp. 1–3, 2013.

[20] S.-J. Lee, Z. Xu, T. Li, and Y. Yang, "A novel bagging C4.5 algorithm based on wrapper feature selection for supporting wise clinical decision making," *Journal of Biomedical Informatics*, vol. 78, pp. 144–155, 2018.

[21] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "To- wards the development of realistic botnet dataset in the Internet things for network forensic analytics: Bot-IoT dataset," 2018, http://arxiv.org/abs/1811.00701.