DOI: 10.24996/ijs.2023.64.5.31

ISSN: 0067-2904

Entropy-Based Feature Selection using Extra Tree Classifier for IoT Security

Chinchu Krishna[1]*, Varghese Paul[2]

[1]Department of Information Technology, Rajagiri School of Engineering and Technology, A P J Abdul Kalam Technological University, Kerala, India,

[2]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, A P J Abdul Kalam Technological University, Kerala, India,

Received: 3/11/2021 Accepted: 21/8/2022 Published: 30/5/2023

Abstract

The Internet of Things (IoT) is a network of devices used for interconnection and data transfer. There is a dramatic increase in IoT attacks due to the lack of security mechanisms. The security mechanisms can be enhanced through the analysis and classification of these attacks. The multi-class classification of IoT botnet attacks (IBA) applied here uses a high-dimensional data set. The high-dimensional data set is a challenge in the classification process due to the requirements of a high number of computational resources. Dimensionality reduction (DR) discards irrelevant information while retaining the imperative bits from this high-dimensional data set. The DR technique proposed here is a classifier-based feature selection using an extra tree classifier (EXT). The entropy values of features are used for the construction of trees in EXT, which is to build a lower-dimensional space. Linear discriminant analysis (LDA), K-nearest neighbor classifier (KNN), decision tree classifier (DTC), and random forest classifier (RFC) empirically evaluate the proposed feature selection mechanism. EXT is compared with other DR techniques like RFC and principal component analysis (PCA). The performance metrics of the classifiers are used to evaluate the proposed work.

Keywords: Dimensionality reduction, extra tree classifier, IoT botnet attack, multiclass classification, entropy.

1. Introduction

Real-time data usually has high-dimensional feature vectors. The high-dimensional feature vectors are a challenge in the classification process due to the requirements of a high number of computational resources. The feature selection transforms the high-dimensional feature space to a low-dimensional feature space by separating the features of lesser pertinence. The DR [1] is the process of discarding irrelevant information while retaining the imperative bits [2]. The three classes of feature selection [1] algorithms are filters, wrappers, and embedded techniques [2] [3]. In the filter method, the selection of features does not depend on machine learning algorithms. In the wrapper method, classifiers are used to generate the best subset of features. The embedded method is a combination of the above two methods. The proposed method in this research work is entropy-based feature selection by EXT, which is an embedded method of feature selection. In this work, EXT is compared with a feature extraction as well as a feature selection mechanism, PCA [4] and RFC, respectively. The empirical evaluation is implemented using a high-dimensional IoT botnet attack dataset.

*Email: chinchuk@rajagiritech.edu.in

The IoT, widely known as the Internet of Everything, is a network of machines and devices competent at interconnecting with each other [5]. The total installed IoT devices was 20.35 billion in 2017 and is forecasted to be as much as 75.44 billion in 2025, a fivefold increase in ten years. There is an enormous change in the way humans utilize the internet. This results in seeking out new ways to connect and being reliant on services that allow us to remotely control, monitor, and manage the endpoints. The IoT application domain encompasses industrial control systems, medical and healthcare, home automation and smart home, smart city, autonomous vehicles, smart traffic, parking control, smart metering, smart grids, etc. [6]. An IoT botnet is a group of hacked computers and internet-connected devices that are co-opted for adulterous purposes. There is a dramatic increase in botnet attacks because of the lack of security mechanisms. In this paper, EXT uses the high-dimensional botnet attack dataset to implement the feature selection. There is an upsurge in the requirement for processing large data sets in real-world problems. These large datasets may comprise redundant and irrelevant information. A widely adopted solution to this problem is feature selection for DR. Feature selection is the technique of selecting a subset of features and thus reducing dimension. Computing areas like computer vision, pattern recognition, machine learning, etc. use this technique to process large data sets. The advantages of feature selection are reduced space complexity, time complexity, and computational cost.

Machine learning classifiers use this lower-dimensional data to validate the effect of DR. The classifiers use the lower-dimensional data from PCA and RFC to perform a comparative analysis. The contributions of this paper propose the study of EXT for entropy-based feature selection. The research study implements the EXT and computes the entropy of each feature vector. The higher-dimensional feature space maps the dataset to a lower dimension based on a threshold value of entropy. The machine learning classifiers assess the performance of the proposed method of entropy-based feature selection.

2. Background

The scope of machine learning techniques in IoT security [7] ranges from authentication, access control, IoT offloading, and malware detection. The authentication process uses different learning methods like supervised learning, unsupervised learning, and deep learning. These techniques induce a protection mechanism against spoofing and eavesdropping, which is a security attack caused by improper authentication mechanisms. The absence or improper use of access control mechanisms results in DoS, intrusion, and malware attacks. Machine learning techniques also formalize an access control mechanism, which helps to work with the IoT devices in different network connections with different sources of data. The absence of proper security mechanisms in IoT offloading may result in DoS attacks and jamming attacks. The algorithms for malware detection are Q-learning, Dyna-Q, post-decision state, and KNN. The proposed method is the multi-class classification of IoT attacks. Rather than malware detection, which is a binary classification, the proposed method classifies attacks into multiple classes.

The multi-class classification of malware focuses on the classification of different IoT attacks. On exploring the area of attacks in IoT, the different types are [8] [9] active attacks, passive attacks, physical layer attacks, datalink layer attacks, network layer attacks, privacy threats, software-based attacks, side-channel attacks, botnet attacks, and protocol-based attacks. The botnet attack results in infected devices are called zombies. The issues with zombies are that the device malfunctions because the infected device is used as a tool for executing another set of attacks, including DDOS (Distributed Denial-of-Service) since it is a compromised device. The papers [8] [9] [10] [11] also address machine learning and deep learning-based solutions to process IoT security attacks. The machine learning algorithms for the classification process are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Another detailed survey [8] [12] of the machine learning and deep learning methods for IoT security enumerates all the machine learning and deep learning techniques for IoT attack classification. The existing research work on these areas is explored in detail, which will help to identify the different learning algorithms for classification. The summary of the papers [8] [9] [10] [11] reveals four types of machine learning methods. Those are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The deep learning architectures explored are generative methods, discriminative methods, and hybrid methods

.

Further studies focus on the classification of IoT botnet attacks, feature selection, and feature extraction. A binary botnet attack classification that identifies the presence of attack implements in botnet attack detection at the IoT edge-based on sparse representation [13]. The sparse representation framework focuses on providing a solution using a limited number of training and testing datasets due to the limited computational power of IoT devices. Isolation of attacked IoT devices proposes here is to reduce the impact of attacks. Here too, the focus primarily is on detecting the attack. Hence a multiclass classification approach may be introduced such that further algorithmic solutions can be fine-tuned with multiple attacks. Deep autoencoders [14] explore botnet attacks. The work presents the classification of normal and abnormal traffic and is binary classification. An unsupervised intelligent system based on a one-class support vector machine and Grey Wolf optimization [15] concentrates on botnet attack detection. There is significant research work in the multi-class classification of IoT attacks, but the research work focused on the multi-class classification of IoT botnet attacks is limited.

The next phase is to study the work that is related to data preprocessing before the training phase of the classifier. The DR techniques before the training phase are feature selection or feature extraction [16]. The two different types of feature selection methods are filter methods and wrapper methods. A combination of these two is called embedded methods [3] [17], which forms a hybrid method. Each method has its advantages, and the combination enhances the contributions from the filter method and the wrapper method.

An induction algorithm generates a subset of the original feature. Cross-validation techniques analyze the process of feature elimination in several combinations. Filter methods are associated with some classifiers that analyze the process of feature elimination. The different types of entropy [18] for feature extraction are approximate, sample, Shannon, Rényi, Tsallis, and permutation. The pruned data produced by DR is used to build the classifier using the least-squares support vector [19]. Medical applications [18] [20] also use entropy-based feature selection for the classification [21] process. The above-mentioned methods of entropy-based DR can be implemented by the EXT-based approach.

The literature survey on existing methodologies explores IoT security issues and the applicability of machine learning classifiers to IoT security for attack classification. The machine learning algorithms recommend a selected/extracted feature set to save time and resources. How to test the validity of the IoT botnet classes generated? This is the area that addresses the evaluation metrics of the classifiers. The impact of feature selection/extraction is reflected in evaluation metrics. Accuracy [22] is a metric to evaluate the performance, which is the ratio between the rightly classified feature points and the total number of feature points. The evaluation metrics are accuracy, F1 score, Cohen's kappa coefficient, and time of execution. The research work proposed here introduces an entropy-based feature selection technique using EXT. The EXT is compared with other feature selection methods of RFC and the feature extraction method of PCA. The implementation of multi-class classification of IoT botnet attacks using machine learning classifiers (LDA, KNN, DTC, and RFC) uses the features extracted from EXT.

3. Data Set Description

The multi-class IoT botnet attack classification uses the real-time IoT traffic dataset instead of emulated or simulated datasets. The features are extracted from nine commercial IoT network traffic datasets called the N-BaIoT [14] and [23]. That is the dataset for experimental evaluation. The data is collected through port mirroring. The port mirroring is deployed on switches and the format of collected data is pcap **(**Packet Capture)**.** The benign dataset and malicious dataset are collected separately. The benign dataset is collected immediately after the installation of the network since it is the basis for identifying other types of attacks. The packet’s contextual information regarding protocols and hosts is captured. The number of features in the dataset is 115. The statistical features are extracted from five temporal windows: 100 ms, 500 ms, 1.5 sec, 10 sec, and 1 min. The attacks [14] in the malicious dataset are caused by two types of botnets, Gafgyt and Mirai. The Gafgyt attacks are scanning for vulnerabilities (g.scan**)**, transmission of spam data (g.spam), UDP flooding (g.udp), TCP flooding (g.tcp), and connection establishment to a specified IP address by transmitting spam data (g.combo). The types of attacks that are caused by the Mirai botnet are: scanning for vulnerabilities (m.scan), Ack flooding (m.ack), Syn flooding (m.syn), UDP flooding (m.udp), limited UDP flooding (m.UDPPlain). The feature headers in the stream aggregation are H (Statistics of traffic from this packet's host (IP), HH (Statistics of traffic going from this packet's host (IP) to the packet's destination host), HpHp (Statistics of summarizing the recent traffic going from packet's host+port (IP) to the packet's destination host+port), HH_jit: (Statistics of summarizing the jitter of the traffic going from this packet's host (IP) to the packet's destination host). The dataset is a balanced dataset. The training of the machine learning classifiers is performed with equal proportions from all the classes. Hence, the chance of identifying the different attacks is equal.

4. Proposed Method

The proposed method in Figure 1 gives an overview of the work. The dataset is real-time traffic data rather than simulated or emulated data. Features may be measured at different scales. The model fitting may result in bias in the predicted output if this is not normalized. The first step is the min-max normalization in data preprocessing. This makes the feature values between 0 and 1. The normalization technique reduces the challenges associated with measurement on different scales. The DR is the next step. The research work proposes EXT for entropy-based feature selection for DR. The EXT computes the entropy of each feature vector. The entropy value indicates the importance of the feature. The higher dimensional feature space maps to a lower dimension based on a threshold value of entropy. The evaluation of the research work is done with other DR techniques like RFC and PCA with machine learning classifiers. The number of principal components (PC) selected is the same as the number of features selected in EXT. This makes the dimension of the input dataset to the classifier the same. The comparison of EXT is also done with RFC. The threshold value of entropy in EXT and RFC is varied between a set of values to identify the optimal threshold value of entropy to study the change in performance of classifiers. The selected dataset is applied to four classifiers independently for a comparative study of the multi-class classification of botnet attacks.

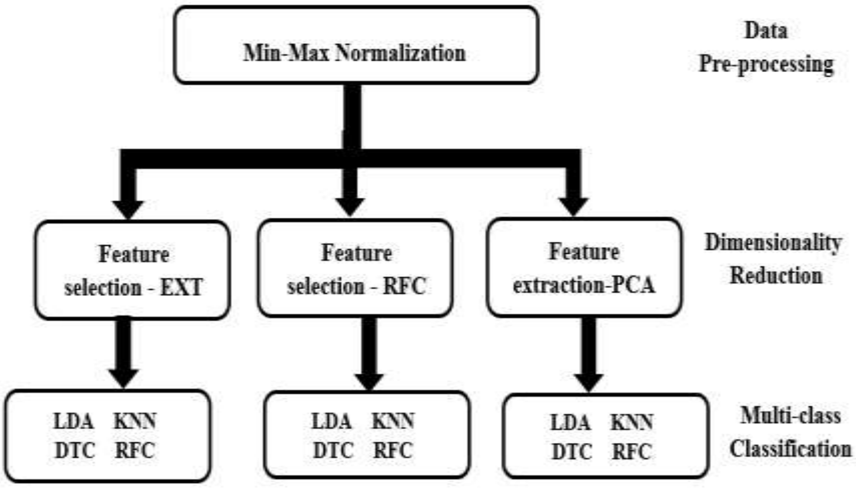

Figure 1: Proposed method

The classifiers use these selected/extracted features for training to evaluate the performance using various parameters. The classifiers implemented are LDA [24], KNN [25], DTC [26], and RFC [27]. The performance metrics for the evaluation of these classifiers [28] [29] are accuracy, time of execution, F1 score, and Cohen's kappa coefficient.

4.1.Data Pre-Processing: Min-Max Normalization

Features may be measured at different scales, and the contribution towards the model fitting may result in bias in the predicted output. Feature-wise normalization applies to the dataset before model fitting. The normalization applied in this paper is Min-Max normalization [30]. In this, all features transform into the range [0, 1]. The minimum and maximum values of a feature are mapped to 0 and 1 respectively. It performs a linear transformation of the original data. It maps a value ‘d’ of ‘p’ to ‘d'’ in the range [0, 1], written as Eq. (1).

$$\mathbf{d'} = \frac{\boldsymbol{d}-\mathbf{min}(\boldsymbol{p})}{\boldsymbol{max}(\boldsymbol{p})-\mathbf{min}(\boldsymbol{p})} \tag{1}$$

where d’ = new value; d = old value and p = feature vector

4.2. EXT and RFC: Entropy-Based Dimensionality Reduction

The EXT procedure generates an ensemble of the unpruned decision or regression trees following the top-down procedure. The splitting of nodes is by selecting cut-points fully at random, hence the name “Extremely Randomized Tree” [31]. In RFC [27], the sampling technique applied in the training set is random sampling with replacement. The principles for the best split are by searching in a subset of randomly selected features. EXT is less computationally expensive than RFC due to the random splits of features.

4.2.1. Entropy

P=(p_1,p_2, . . ,p_m) is a finite discrete probability distribution where p_k>0 (k=1,2, 3,…,m) and $\sum_{\boldsymbol{k=1}}^{\boldsymbol{m}} \mathrm{p}_{\boldsymbol{k}} = \mathbf{1}$. The amount of uncertainty of the probability distribution concerning the outcome of an experiment p_1, p_2, p_3, . . , p_m is called the entropy (H) of the distribution [18] [20] [21], written as Eq. (2).

$$\mathbf{H}(\mathrm{p}_{\mathbf{1}}, \mathrm{p}_{\mathbf{2}}, \dots, \mathrm{p}_{\boldsymbol{m}}) = \sum_{\boldsymbol{k=0}}^{\boldsymbol{m}} \mathrm{p}_{\boldsymbol{k}}\, \boldsymbol{log_2}\, \frac{\mathbf{1}}{\mathrm{p}_{\boldsymbol{k}}} \tag{2}$$

where H = Entropy and p_1,p_2, . . ,p_m is a finite discrete probability distribution

4.2.2. Feature selection

In EXT, the construction of a forest is determined by the mathematical criteria in the decision of feature selection. The entropy is the mathematical criteria calculated, and this indicates the importance of the features. The entropy is ordered in descending order. The feature selection is based on a predefined threshold value of entropy and thus implements DR. The information gained for a sample S, $Score_c(s, S)$ is written as Eq. (3).

$$Score_c(s, S) = \frac{2I_c^S(S)}{H_s(S) + H_c(S)} \tag{3}$$

where I_c^S (S) = mutual information; Hs(S) = split entropy and Hc(S) = entropy.
The example for entropy-based feature selection is given below. Consider the following data in Table 1.

Table 1: Data for decision tree construction

Day	Weather	Level of temperature	Level of humidity	Level of wind	Day outing
1	Cloudless day	High	High	Gentle breeze	No
2	Cloudless day	High	High	Strong breeze	No
3	Cloudy day	High	High	Gentle breeze	Yes
4	Rainy day	Medium	High	Gentle breeze	Yes
5	Rainy day	Low	Optimal	Gentle breeze	Yes
6	Rainy day	Low	Optimal	Strong breeze	No
7	Cloudy day	Low	Optimal	Strong breeze	Yes
8	Cloudless day	Medium	High	Gentle breeze	No
9	Cloudless day	Low	Optimal	Gentle breeze	Yes
10	Rainy day	Medium	Optimal	Gentle breeze	Yes
11	Cloudless day	Medium	Optimal	Strong breeze	Yes
12	Cloudy day	Medium	High	Strong breeze	Yes
13	Cloudy day	High	Optimal	Gentle breeze	Yes
14	Rainy day	Medium	High	Gentle breeze	Yes

EXT builds the extra tree forest by the following parameters
Number of decision tree=5
Number of features in a random sample of features=2
The entropy is calculated as
Entropy(S)=0.863
The equation to calculate information gain, G is written as Eq. (4).

$$G(S, X) = Entropy(S) - Entropy(S, X) \tag{4}$$

Table 2: Calculated values of information gain

Decision tree number	Feature 1	Feature 2	Gain(Š, Feature 1)	Gain(Š, Feature 2)
1	Weather	Level of temperature	0.259	0.067
2	Level of temperature	Level of wind	0.067	0.025
3	Weather	Level of humidity	0.259	0.075
4	Level of temperature	Level of humidity	0.067	0.075
5	Level of wind	Level of humidity	0.025	0.075

The information values calculated are shown in Table 2. The total gain for the features of weather, level of temperature, level of humidity, and level of wind is 0.518, 0.201, 0.225, and 0.05, respectively. The order of features based on importance is weather, level of humidity, level of temperature, and level of wind. The most important feature is the weather. The threshold value of the feature importance helps to reduce the number of features if there is high-dimensional input data.

4.2.3. PCA

PCA is a feature extraction method [32] [33] [34] of the DR technique. From a high-dimensional data set, a low-dimensional dataset is formed [35]. This low-dimensional dataset [36] will contain most of the information in the high-dimensional dataset. It recommends preprocessing steps like standardization. The feature variables correlate with each other so that there may be redundancy in the dataset. The generation of the covariance matrix approximates the correlations. The eigenvectors and eigenvalues from the covariance matrix calculate the PC of the data. PCs are combinations or mixtures of the initial variables that eliminate correlation. Then, the features are in compressed form.

4.3. Machine Learning Classifiers

The classifiers used to create the model are LDA, KNN, DTC, and RFC. The LDA [24] for 11 class classification form 10 non-zero eigenvalues. A Euclidean distance is used to classify data points; the smallest Euclidean distance is used to assign the feature vector to a particular class. The KNN [25] rule maps an unclassified feature vector to the classification of the nearest member of the set of previously classified feature vectors. The given n pairs of data points are $\{(x_1, \Theta_1), (x_2, \Theta_2), (x_n, \Theta_n)\}$, x_i takes the values in a metric space X upon which is defined as a metric d and Θ_i is the index of the category to which xi belongs. A new point (x, Θ) is to be classified to the nearest neighbor $\boldsymbol{x'_n}$ where $\boldsymbol{x'_n} \in \{x_1, x_2, x_3, \ldots, x_n\}$ if $\min d(x_i, x) = d(\boldsymbol{x'_n}, \boldsymbol{x})$, i=1, 2,….,n.

In DTC [26], the first phase is to perform the selection of splits. The second phase decides the terminal nodes, and the last is the assignment of each terminal node to a class label. Terminal nodes are assigned to the classes which have the highest probabilities to minimize the problem of misclassification rate in the class assignment problem. RFC [27] incorporates an amalgamation of tree classifiers where each classifier is generated using a random vector sampled separately from the input feature space, and each tree is assigned to classify an input vector. The total number of trees in the forest is formulated as 100. The criterion used to measure the quality of the test is the Gini index. The nodes are expanded until all leaves are pure or until all leaves contain fewer than two nodes. Here, RFC is used both for feature selection and feature extraction. The next step is to describe the input and output of the classifiers. The feature extraction/selection phase generates lower-dimensional data from the actual dataset. The training dataset consists of 11 classes, which include 10 attack classes and one benign class.

The algorithm is written below (4.3.1). This gives a brief overview of the proposed method. Preprocessing is implemented with min-max normalization. This generates a normalized dataset. The next step splits the datasets into a training dataset and a test dataset. The train test ratio is 70%-30%. Once the training and test data are generated, the next phase is entropy-based feature selection. The extra tree algorithm provides the feature importance, which is the measure of entropy. The threshold of feature importance is varied from 0.01 to 0.055 with an increment of 0.005 to attain the best feature set without degradation of model performance. The selected features have a feature importance above the specified threshold. The different models like LDA, KNN, DTC, and RFC use these selected features for training.

4.3.1. Algorithm

```
1. Apply Min_max_normalization on the dataset.
2. Split the dataset to train dataset and test dataset
3. Calculate the entropy of feature vector by EXT of train dataset
4. For threshold value from 0.01 to 0.055 of entropy, select features
    4.1.Train and test 4 models separately with LDA, KNN, DTC, RFC with train dataset
    4.2.Test the model with test dataset and calculate accuracy
5. Repeat steps 3 and 4 by RFC (Instead of EXT)
6. Extract features by PCA
    6.1.Train and test 4 models separately with LDA, KNN, DTC, RFC with train dataset
    6.2.Test the model with test dataset and calculate accuracy
```

5. Results and Discussion

5.1.Experiments Environment and Setup

The programming language is Python 3.7. The libraries used in Python are Pandas 0.23.4, Scikit-learn 0.19.2 and Scipy 1.6.2. All experiments are conducted on a Ubuntu 18.04 (x86_64 architecture), Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz. VGA compatible controller is the NVIDIA Corporation GP107M [GeForce GTX 1050 Mobile].

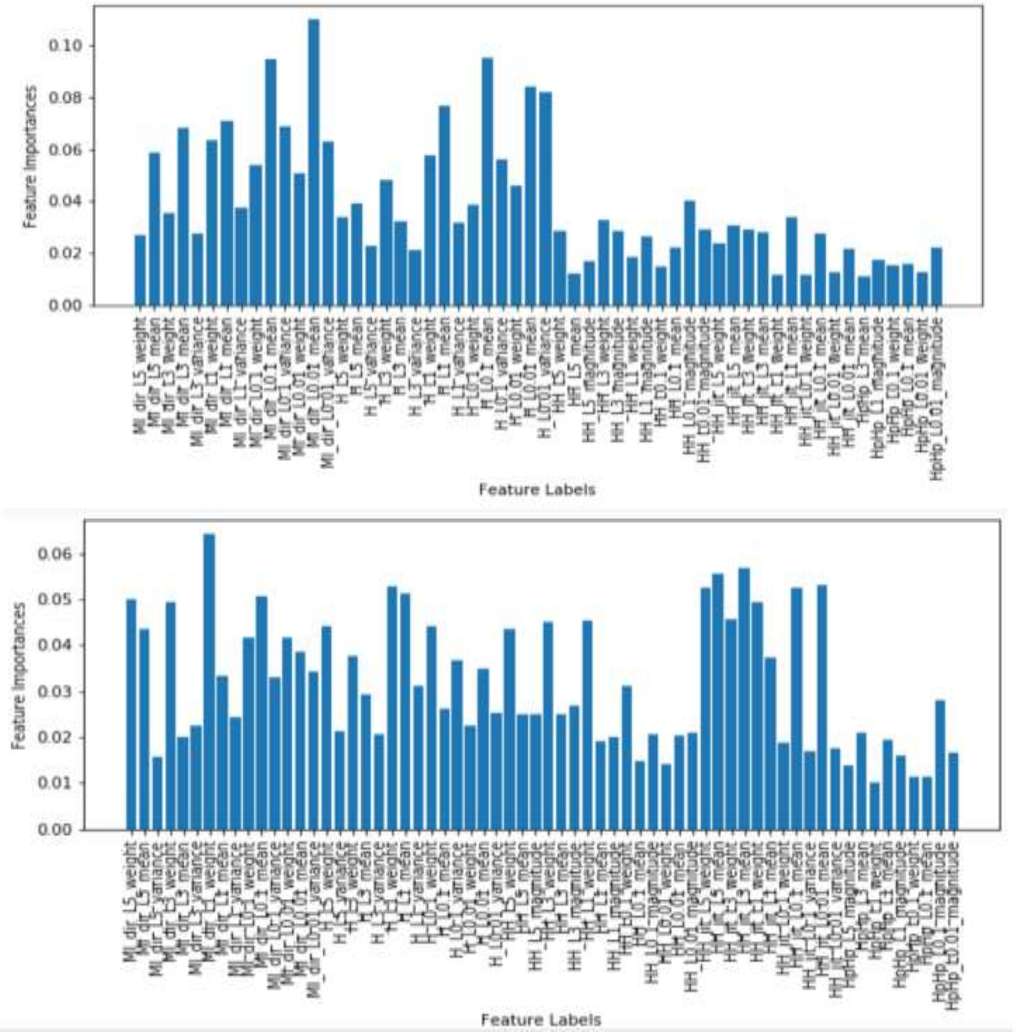

Figure 2: Selected features with entropy threshold values of 0.01 of EXT and RFC

5.2.Feature Selection

EXT maps the higher-dimensional data to the lower dimension by feature selection. The dimension of the dataset taken for evaluation is 89000*115. To study the effect of DR in classifiers, the threshold value for the entropy is changed in the interval from 0.01 to 0.055. The features that have a feature importance (entropy value) above the threshold value are selected. The process repeats for RFC. The selected feature sets of threshold value 0.01 of EXT and RFC are shown in Figure 2. There is a constant decrease in the number of selected features in EXT and RFC. The graph showing the relationship between the entropy threshold value and the number of features selected is shown in Figure 3. The next session will examine the effects of this selected dataset by training with different classifiers. The comparative study of EXT with PCA is done by generating an extracted dataset from PCA with the same dimension as EXT.

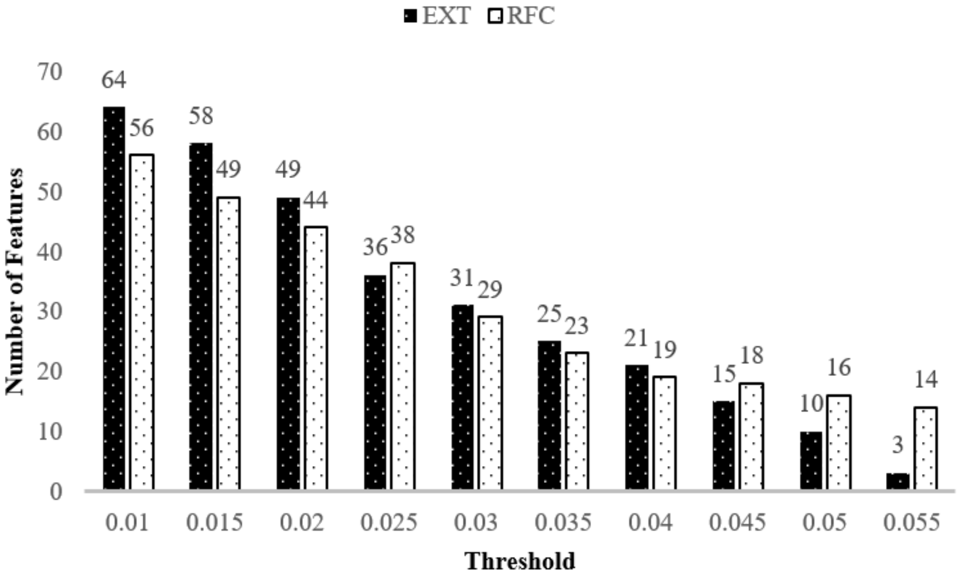

Figure 3: Number of features with different threshold values of EXT and RFC

5.3. Accuracy

The selected features are used to train the classification models with LDA, KNN, DTC, and RFC algorithms. The most commonly used measure of the classification algorithm is accuracy [27] [32]. It is a ratio between the rightly classified feature points and the total number of feature points, written as Eq. (5).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

where TP = true positive; TN = true negative; FP = false positive and FN = false negative. The number of samples that are correctly classified from class A, TP of class i=$C_{i,i}$

The false negative in the ith class is the sum of all class samples that were incorrectly classified as other classes, written as Eq. (6).

$$\text{FN of class i} = \left(\sum_{j=1}^{11} C_{i,j}\right) - C_{i,i} \quad (6)$$

The false positive for any class i is written as Eq. (7).

$$\text{FP of class i} = \left(\sum_{j=1}^{11} C_{j,i}\right) - C_{i,i} \quad (7)$$

The accuracy analysis is done from two perspectives. The first is to analyze the different DR techniques, and the second is to analyze the classifiers. The accuracy analysis of EXT, RFC, and PCA is done by training the datasets from these algorithms with classifiers. The best performing classifier with PCA is KNN. The number of PCs chosen is 64, 58, 49, 36, 31, 25, 21, 15, 10, 3. These numbers are equal to the number of features selected in EXT. The accuracy of KNN is above 99.5% for all the datasets with the PC from 64 to 10. The accuracy is 98.3% for the dataset with 3 PCs. The next performing classifiers are DTC and RFC with PCA. All are providing almost the same accuracy. The least performing classifier is LDA with PCA. It gives accuracy above 70% for all datasets with PC equal to or greater than 15.

The RFC feature selection technique provides a different number of selected features from EXT and is shown in Figure 3. The best performing classifier using RFC feature selection is also KNN. The accuracy is above 99.6% when the number of features is equal to or greater than 56. The accuracy corresponding to the number of features between 16 and 23 is above 89%. The number of features at threshold 0.005 is 14 and accuracy is 84.2% with RFC feature selection. DTC and RFC classifiers provide the same level of accuracy for all the selected features with RFC. The difference is too small, hence lines are getting overlapped in the graph. The least performing classifier is LDA. The number of features selected should be greater than or equal to 38 to get the accuracy at least above 71%. In lower dimensions, there is a decrement in the accuracy for KNN, DTC, and RFC.

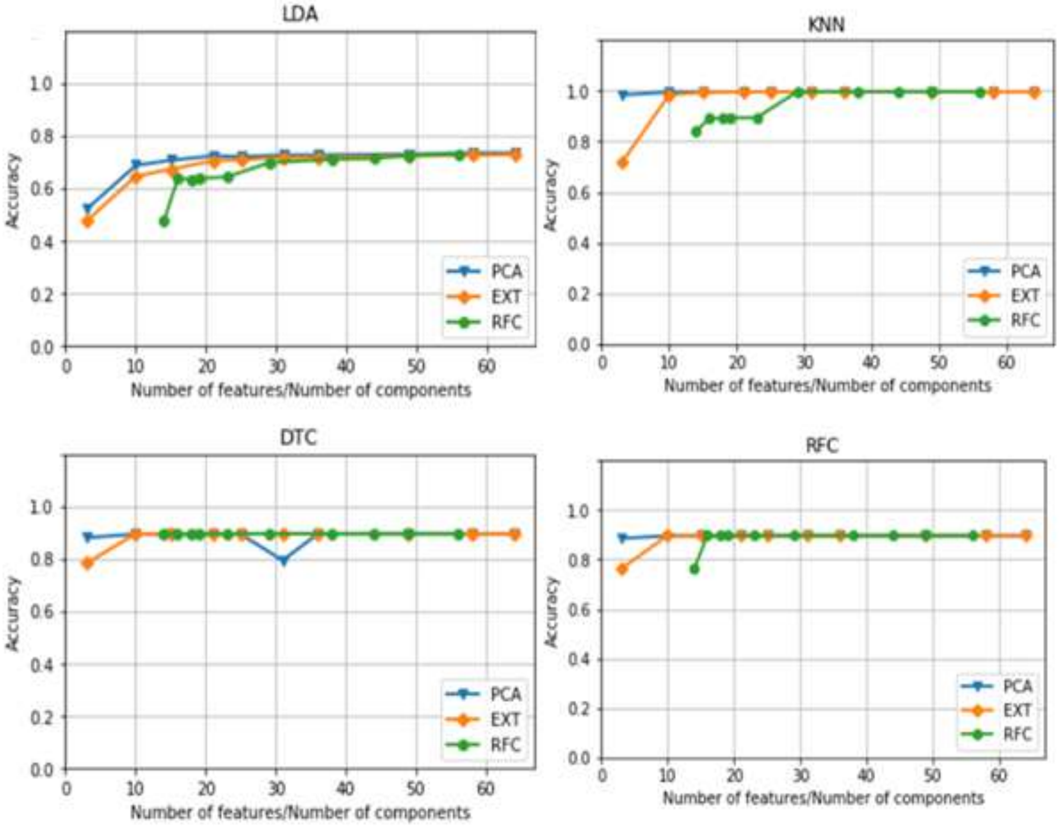

Figure 4: Accuracy analysis of classifiers

The best performing classifier with EXT is also KNN. The threshold of 0.05 selects 10 features and the KNN accuracy for this dataset is 98.4%. For all the thresholds except 0.05 and 0.055, the accuracy is between 99.4% and 99.6%. Among the three DR techniques, the best performing classifier is KNN. When the threshold is 0.055, the number of features selected is 3, and KNN accuracy is 71.7%. At this value of the threshold, DTC and RFC are better, and the accuracies are 78.4% and 76.2%. The next two performing classifiers with EXT are RFC and DTC. Since they are giving almost the same performance, the overlapping happens like PCA and RFC. DTC and RFC generate an accuracy of 89.7% for all the thresholds except 0.055. The selected 10 features out of 115 are sufficient for this performance. An accuracy analysis of classifiers for EXT, RFC, and PCA is shown in Figure 4. The accuracy is almost the same for KNN with these three techniques, hence overlapping happens in the graph. This kind of performance overlapping for a different set of feature vectors happens in DTC and RFC as well. Optimal performance is identified whenever the accuracy decrease/fall happens. Thus, the minimum feature set is the feature set corresponding to the optimal performance.

5.4. *Time of Execution*

The next parameter for evaluation is the time of execution. The classifiers are evaluated with PCA, RFC, and EXT. It is shown in Figure 5. The x-axis is the number of features for EXT and RFC and the number of components for PCA, as in Figure 4. The y-axis is the time of execution in seconds. The most important result is that EXT performs best in all the classifiers in terms of execution time. In LDA and DTC, RFC is the next performed one. In KNN, RFC has a different pattern of time of execution compared with EXT and PCA. The second commonly performed DR technique with KNN is PCA. In the RFC classifier, the feature reduction techniques of PCA and RFC have almost equal time of execution. The longer time of classification with PCA may be justified with feature selection time, but this work analyses the time of execution of classifiers.

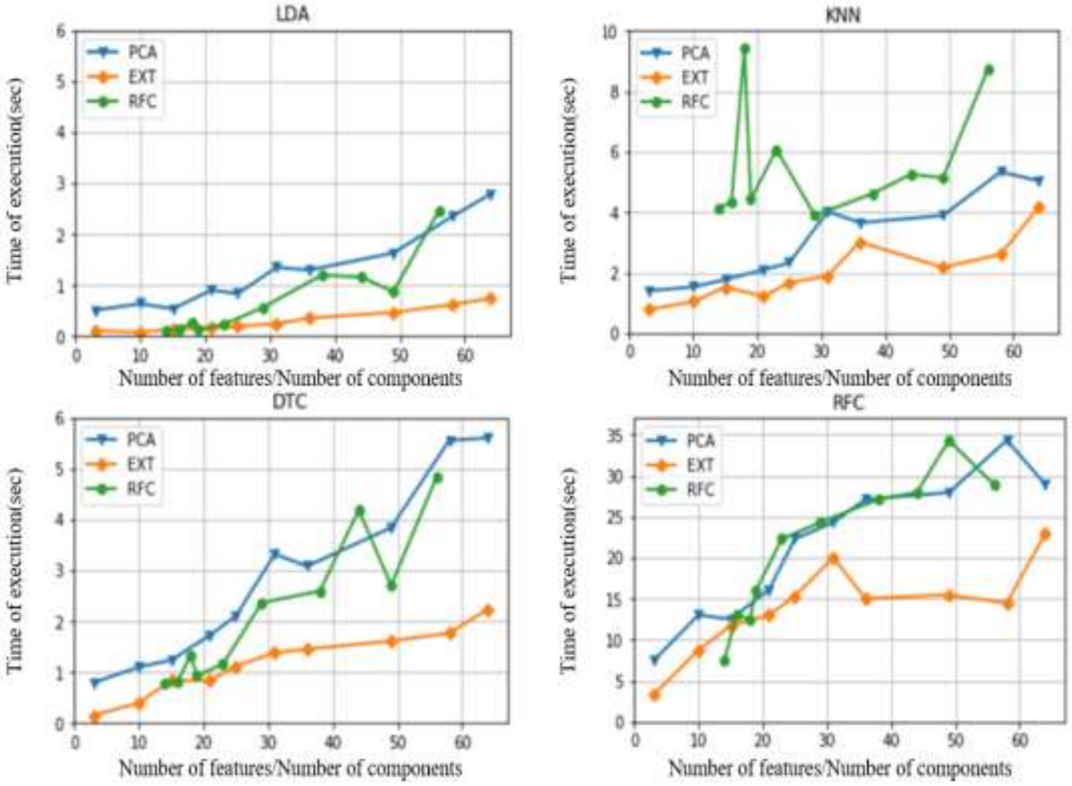

Figure 5: Time of Execution

5.5. *F_1 score and Cohen's kappa coefficient*

The F_1 score is another parameter to estimate the test's accuracy, which is written as Eq. (8). The F_1 [27] [32] computation uses both precision (PR) and recall (RC), which is the harmonic mean of PR and RC. Precision is the proportion of TP to the sum of TP and FP. The recall is the capacity to properly recognize positive to attain the true positive rate. To quantify a test's accuracy [27], the parameter used is the F_1 score, and it balances the use of precision and recall.

$$F_1 \text{ score} = 2 \cdot \frac{PR \cdot RC}{PR+RC} \tag{8}$$

where $PR = \frac{TP}{TP+FP}$ and $RC = \frac{TP}{TP+FN}$.

Cohen's kappa coefficient is a parameter used to compute inter-rater reliability [26]. Note that no distributional or random sampling assumptions are necessarily involved in kappa's calculation. It is written as Eq. (9).

$$\text{К} = \frac{p_o - p_e}{1 - p_e} \tag{9}$$

where p_o = accuracy calculated and $p_e = \frac{\sum_{i=1}^{11} C_{:i} * C_{i:}}{Total\ samples}$

where $C_{:i}$ and $C_{i:}$ = the sums of elements in the ith column and ith row of the confusion matrix, respectively. ; p_o = probability of agreement and p_e = sum of correct and incorrect probabilities.

The F_1 score and Cohen's kappa coefficient of classifiers that use the feature set from EXT are shown in Table 3.

Table 3: F_1 Score and Cohen's kappa coefficient of classifiers with EXT

Entropy threshold value	No. of features	F_1 Score				Cohen's kappa coefficient			
		LDA	KNN	DTC	RFC	LDA	KNN	DTC	RFC
0.01	64	0.729	0.996	0.897	0.897	0.702	0.996	0.887	0.887
0.015	58	0.727	0.996	0.897	0.897	0.699	0.996	0.886	0.887
0.02	49	0.725	0.997	0.897	0.897	0.697	0.996	0.887	0.887
0.025	36	0.719	0.997	0.897	0.897	0.69	0.996	0.886	0.887
0.03	31	0.719	0.996	0.897	0.897	0.69	0.996	0.886	0.887
0.035	25	0.709	0.996	0.897	0.897	0.679	0.996	0.886	0.887
0.04	21	0.705	0.996	0.897	0.897	0.675	0.996	0.886	0.887
0.045	15	0.673	0.994	0.896	0.897	0.64	0.993	0.886	0.886
0.05	10	0.647	0.984	0.896	0.897	0.611	0.983	0.886	0.887
0.055	3	0.477	0.717	0.785	0.763	0.422	0.688	0.763	0.739

The importance of the number of features can be analyzed from Table 1. The LDA gives an F1 score above 0.7 if the number of features is 21 or above. But KNN gives an F1 score above 98 if the number of features is 10 or above. DTC and RFC give an F1 score above 89 if the number of features is 10 or above. The best performing algorithm in terms of F1 score is KNN. Similarly, the best performing algorithm in terms of Cohen's kappa coefficient is KNN, the next two are RFC and DTC, and the least performing is LDA.

6. Conclusion

The research work proposed is the study of EXT for entropy-based feature selection. This implements the EXT and computes the entropy of each feature vector, which is the feature importance. The threshold value on this feature's importance provides a reduced set of features. The work focused on data analysis of security mechanisms of IoT applications that used high-dimensional real-time traffic data sets. The higher-dimensional feature space maps the dataset to a lower dimension based on a threshold value of entropy. EXT is compared with existing methods of RFC and PCA. A RFC is an entropy-based feature selection. While PCA is a feature extraction method, the number of PC chosen in PCA is the same as the number of features selected in EXT, which makes the dimension of the input dataset to the model the same. The machine learning classifiers assess the performance of the proposed method of entropy-based feature selection. The machine learning classifiers used are LDA, KNN, DTC, and RFC. The evaluation metrics are accuracy, time of execution, F_1 score, and Cohen's kappa coefficient. The classification process generates different classes of IoT botnet attacks like Gafgyt and Mirai. In the eleven output classes, five are of Gafgyt, five are of Mirai attacks, and one is the benign class. The experimental evaluation conducted herein emphasizes that the entropy-based DR using EXT provides feature selection with high classification accuracy in a short amount of time.

The proposed work focuses on the multi-class classification of botnet attacks. Exploring the broad area of IoT security attacks, an IoT botnet attack is one type of attack class. The work can be further extended to process all the existing IoT attacks: active attacks, passive attacks, physical layer attacks, datalink layer attacks, network layer attacks, privacy threats, software-based attacks, side-channel attacks, and protocol-based attacks. It provides a proper guideline for the development of IoT security solutions. Rather than using machine learning classifiers, deep learning can also be introduced for IoT attack classification and analysis, which requires extended research work.

Reference

[1] G. T. Reddy, M. P. Kumar Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava and T. Baker, "Analysis of Dimensionality Reduction Techniques on Big Data," *IEEE Access*, vol. 8, pp. 54776 - 54788, 2020.

[2] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists, O'Reilly Media, Inc.*, 2018.

[3] B. Venkatesh and J. Anuradha, "A Review of Feature Selection and Its Methods," *Cybernetics and Information Technologies*, vol. 19, pp. 3-26, 2019.

[4] Y. Aït-Sahalia and D. Xiu, "Principal Component Analysis of High Frequency Data," *Journal of the American Statistical Association*, vol. 114, no. 525, pp. 287-303, 2019.

[5] P. Asghari, A. M. Rahmani and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241-261, 2019.

[6] C. U. Om Kumar and P. R. K. Sathia Bhama, "Detecting and confronting fash attacks from IoT botnets," *The Journal of Supercomputing*, vol. 75, p. pages8312–8338, 2019.

[7] L. Xiao, X. Wan, X. Lu, Y. Zhang and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41 - 49, 2018.

[8] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 3, pp. 1646-1685, 2020.

[9] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh and W.-C. Hong, "Internet of things: Evolution, concerns and security challenges," *Sensors*, vol. 21, no. 5, p. 1809, 2021.

[10] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686 - 1721, 2020.

[11] A. Churcher, R. Ullah, J. Ahmad, S. u. Rehman, F. Masood, M. Gogate, F. Alqahtani, B. Nour and W. J. Buchanan, "An experimental analysis of attack classification using machine learning in iot networks," *Sensors*, vol. 21, no. 2, p. 446, 2021.

[12] M. A. Amanullah, R. A. Ariyaluran Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. Mohamed Nainar, N. M. Akim and M. Imran, "Deep learning and big data technologies for IoT security," *Computer Communications*, vol. 151, pp. 495-517, 2020.

[13] C. Tzagkarakis, N. Petroulakis and S. Ioannidis, "Botnet attack detection at the IoT edge based on sparse representation," in *Global IoT Summit* (GIoTS),IEEE, 2019.

[14] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-22, 2018.

[15] A. A. Shorman, H. Faris and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7, p. 2809–2825, 2020.

[16] S. Rathee and S. Ratnooa, "Feature selection using multi-objective CHC genetic algorithm," *Procedia Computer Science*, vol. 167, pp. 1656-1664, 2020.

[17] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[18] S. Nalband, A. Prince and A. Agrawal, "Entropy-based feature extraction and classification of vibroarthographic signal using complete ensemble empirical mode decomposition with adaptive noise," *IET Science, Measurement & Technology*, vol. 12, no. 3, pp. 350-359, 2018.

[19] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1258-1265, 2018.

[20] L. Sun, X. Zhang, Y. Qian, J. Xu and Z. Shiguang, "Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification," *Information Sciences*, vol. 502, pp. 18-41, 2019.

[21] J. S. Deshmukh and A. K. Tripathy, "Entropy based classifier for cross-domain opinion mining," *Applied computing and informatics*, vol. 14, no. 1, pp. 55-64, 2018.

[22] Y. Zhou, J. Kang, S. Kwong, X. Wang and Q. Zhang, "An evolutionary multi-objective optimization framework of discretization-based feature selection for classification," *Swarm and Evolutionary Computation*, vol. 60, p. 100770, 2021.

[23] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai and Y. Elovici, "detection_of_IoT_botnet_attacks_N_BaIoT Data Set," 19 March 2018. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT. [Accessed 12 June 2020].

[24] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen and Y. Xu, "Robust sparse linear discriminant analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 390-403, 2018.

[25] S. Zhang, X. Li, M. Zong, X. Zhu and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1774-1785, 2017.

[26] A. Suresh, R. Udendhran and M. Balamurgan, "Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers," *Soft Computing*, vol. 24, no. 11, pp. 7947-7953, 2020.

[27] M. Bassier, M. Vergauwen and B. V. Genechten, "Classification of sensor independent point cloud data of building objects using random forests," *Journal of Building Engineering*, vol. 21, pp. 468-477, 2019.

[28] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168-192, 2020.

[29] Z. Mousavi, T. Y. Rezaii, S. Sheykhivand, A. Farzamnia and S. N. Razavi, "Deep convolutional neural network for classification of sleep stages from single-channel EEG signals," *Journal of Neuroscience Methods*, vol. 324, p. 108312, 2019.

[30] S. Jain, S. Shukla and R. Wadhvani, "Dynamic selection of normalization techniques using data complexity measures," *Expert Systems with Applications*, vol. 106, pp. 252-262, 2018.

[31] U. Saeed, S. U. Jan, Y.-D. Lee and I. Koo, "Fault diagnosis based on extremely randomized trees in wireless sensor networks," *Reliability Engineering & System Safety*, vol. 205, p. 107284, 2021.

[32] S. Nofallah, S. Mehta, E. Mercan, S. Knezevich , C. J. May, D. Weaver, D. Witten , J. G. Elmore and L. Shapiro, "Machine learning techniques for mitoses classification," *Computerized Medical Imaging and Graphics*, vol. 87, p. 101832, 2021.

[33] M. P. Uddin, M. A. Mamun, M. I. Afjal and M. A. Hossain, "Information-theoretic feature selection with segmentation-based folded principal component analysis (PCA) for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 42, no. 1, pp. 286-321, 2021.

[34] D. Yang, W. Chen, H. Shi, F. Wan and Y. Zhou, "Raman spectrum feature extraction and diagnosis of oil–paper insulation ageing based on kernel principal component analysis," *High Voltage*, vol. 6, no. 1, pp. 51-60, 2021.

[35] M. R. Mahmoudi, M. H. Heydari, S. N. Qasem, A. Mosavi and S. S. Band, ""Principal component analysis to study the relations between the spread rates of COVID-19 in high risks countries," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 457-464, 2021.

[36] J. Zhang, W. Cui, X. Guo, B. Wang and Z. Wang, "Classification of digital pathological images of non-Hodgkin's lymphoma subtypes based on the fusion of transfer learning and principal component analysis," *Medical Physics*, vol. 47, no. 9, pp. 4241-4253, 2020.