Received July 29, 2019, accepted August 12, 2019, date of publication August 19, 2019, date of current version September 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936346

Mini-Batch Normalized Mutual Information: A Hybrid Feature Selection Method

G. S. THEJAS[1], (Member, IEEE), SAJAL RAJ JOSHI[2], S. S. IYENGAR[1], (Life Fellow, IEEE), N. R. SUNITHA[2], (Member, IEEE), AND PRAJWAL BADRINATH[1]
[1]School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA
[2]Department of Computer Science Engineering, Siddaganga Institute of Technology, Tumakuru 572103, India

Corresponding authors: G. S. Thejas (tgs001@fiu.edu) and Sajal Raj Joshi (sajrajjazz@gmail.com)

This work was supported by the Florida International University Graduate School Dissertation Year Fellowship Award received by the author G. S. Thejas.

ABSTRACT Feature Selection has been a significant preprocessing procedure for classification in the area of Supervised Machine Learning. It is mostly applied when the attribute set is very large. The large set of attributes often tend to misguide the classifier. Extensive research has been performed to increase the efficacy of the predictor by finding the optimal set of features. The feature subset should be such that it enhances the classification accuracy by the removal of redundant features. We propose a new feature selection mechanism, an amalgamation of the filter and the wrapper techniques by taking into consideration the benefits of both the methods. Our hybrid model is based on a two phase process where we rank the features and then choose the best subset of features based on the ranking. We validated our model with various datasets, using multiple evaluation metrics. Furthermore, we have also compared and analyzed our results with previous works. The proposed model outperformed many existent algorithms and has given us good results.

INDEX TERMS Feature selection, filter method, hybrid feature selection, normalized mutual information, mini batch K-means, random forest, wrapper method.

I. INTRODUCTION

One of the essential phases in classification is to determine the useful set of features for the classifier. In supervised as well as in unsupervised learning, the large volume of data has become a significant problem and is becoming more prominent with the increase in data samples and the number of features in each sample. The main intention of reducing the dimension by keeping a minimum number of features is to decrease the computation time, obtain greater accuracy, and reduce overfitting.

Dimensionality reduction is divided into 2 categories: Feature Extraction (FE) and Feature Selection(FS). In FE, we transform the existing features into new features with lesser dimensionality employing a linear or a nonlinear combination of features. In this method, the actual data is manipulated and hence not immune from distortion under transformation. In the FS process, we select the feature's subset based on some criteria. Many of the attributes in the dataset may be utterly irrelevant to the class or redundant when considered along with other features. The accuracy of the induced classifier is decreased by the presence of irrelevant or redundant features [1]. Identifying such features and removing them reduces the dimensionality which inturn reduces the computation time while improving the accuracy. In [2], they state that the overabundance of features rendered the nearest neighbor approach on Internet Advertisement dataset.

FS has many applications in various fields like image processing, natural language processing, bioinformatics, data mining, and machine learning(ML) [3]. The selection method is divided into two standard categories based on their working modules, classifier independent 'filter' technique, and classifier dependent 'wrapper' and 'embedded' technique.

The filter technique, a classifier independent process, performs the selection of the features based on statistical metrics like distance, correlation, consistency measure, and mutual information (MI). It either ranks the features or provides a relevant subset of features associated with the class label. It improves the computational efficiency and scales down the data dimensionality by entirely being independent of the classifier [4]. The drawback of this process is the lack

The associate editor coordinating the review of this manuscript and approving it for publication was Nizam Uddin Ahamed.

of knowledge regarding the relationship between feature attributes and target class.

The classifier dependent systems rely upon the classifier for the selection process. The wrapper method uses the outcome of the classifier to obtain the subset of features, making it biased to the classifier. Also, it is vulnerable to overfitting, mostly when the quantity of data is very few [5]. The embedded method makes use of the classifier in the training phase and selects the optimal features like the learning procedure. When compared to the wrapper method, embedded methods are less vulnerable to overfitting and computation is much faster [6].

We propose a combination of filter and wrapper method, which has the advantage of both the techniques. It is fast and general like the filter method. At the same time, it accounts to learning algorithm obtaining the best set of features without the need for the user to input the feature number unlike most of other established algorithms like Recursive Feature Elimination (RFE).

In this paper, we cluster the data using mini-batch Kmeans clustering and rank them using normalized mutual information(NMI), a measure to calculate the relevance and the redundancy between the candidate attribute and the class. We apply a greedy search method by using Random Forest to get the optimal set of features. However, our method is flexible in terms of the learning algorithm that can be used in our process.

Organization of the paper: Section II discusses the related work regarding various standard techniques as well as different hybrid approaches. In section III, we discuss the preliminary concepts behind this work and propose our techniques. In section IV, we elaborate each component of our work in detail. In section V, we show experimental results and compare them with the previous works, and in Section VI, we conclude our work and give light to the future work.

A. ABBREVIATIONS AND ACRONYMS

All Features(AF), Feature Selection (FS), Feature Extraction (FE), Mini Batch K-Means Normalized Mutual Information Feature Inclusion(KNFI), Mini batch K-Means Normalized Mutual Information Feature Elimination (KNFE), Normalized Mutual Information (NMI), Random Forest (RF), Recursive Feature Elimination (RFE).

II. RELATED WORK

A. FILTER METHOD

Guyon and Elisseeff [7] give information about all the developments to improve the performance of the model using statistical analysis. They came up with a simple approach where less computation is required. Their process did not consider the dependency between the features but considered each feature as an independent one. Saeys *et al.* [4] show that the various FS methods have given impressive results in the field of bioinformatics. Using Weka tool, Pushpalatha and Karegowda [8] perform CFS based filter approach to rank with five search techniques. Dash *et al.* [9] choose the best feature subset for clustering by evaluating the various subsets of features. It considers the effect of the underlying clusters with no unanimous agreement in evaluating the clusters.

B. WRAPPER METHOD

In [10], the authors introduced a variant of particle swarm optimization (PSO) to determine the least number of features which results in finer classification. It is a wrapper based technique named as competitive swarm optimizer (CSO). Also, they proposed an archive technique to reduce computational cost. Jiang *et al.* [11] optimized the multi objective function of a pre-existing wrapper method to a single objective function to reduce the computation cost by adding a new evaluation function. Mafarja and Mirjalili [12] introduced a new wrapper method which was mainly based upon Whale Optimization Algorithm (WOA), with slight changes to make the model work even for binary datasets. Xue *et al.* [13] performed feature selection with genetic algorithm and extreme machine learning, which is computationally efficient in comparison with other wrapper methods.

C. HYBRID METHOD

Venkatesh and Anuradha [14] came up with a hybrid approach of filter and wrapper method by considering MI and RFE. Sharmin *et al.* [15] used MI as a metric for creating a framework for selecting features and discretization at the same time based on x^2 test. Hoque *et al.* [3] considered the information between the attributes and the classes. They considered the MI of the candidate attribute with all the attributes in the selected set of feature attributes. Genetic algorithm was used to select attributes that increased the MI with the label class and decreased the MI with other feature attributes. Battiti [16] introduced Mutual Information Feature Selection (MIFS), an incremental greedy search method to select the most likely 'k' features among n features. Instead of calculating MI between the attributes and the classes, they calculated MI between the attributes i.e., the previously selected attributes and the set of candidate attributes. The performance tends to degrade if there are significant errors in estimating MI. Kwak and Choi [17] proposed an improvised method of MIFS to improve the estimations of MI between the class labels and the input attributes called MIFS-U. Peng *et al.* [18] proposed a method mRMR which avoids expansion of subset where the redundancy divides over the cardinality $\|C\|$ of the selected subset. Brown *et al.* [5] justified that this alteration allows mRMR to outperform the established MIFS & MIFS-U techniques. Estevez *et al.* [19] normalized the value of MI to curb down the value between zero and unity, which removed the bias towards multivalued features. The proposed approach of normalizing the value of mutual information in the FS process, namely NIMFS, which is as an upgraded model of mRMR, MIFS-U, and MIFS to find the irrelevant and redundant features. They also proposed genetic algorithm based feature selection process. Haury *et al.* [20] give the

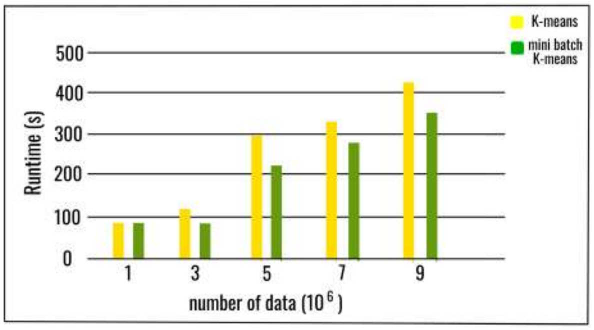

FIGURE 1. Runtime analysis of k-means and minibatch k-means. [25].

comparative analysis of FS method based upon stability and interpretability of the classes. This comparative analysis suggested that a simple filter method outperformed more complex embedded and wrapper method. Faker and Dogdu [21] considered homogeneity metric as a measure to rank and remove the least ranked features. Zhang *et al.* [22], proposed a hybrid filter and wrapper method where they created a subset of features with bootstrapping strategy. For each subset, classification accuracy is calculated to find the optimized subset.

III. PROPOSED APPROACH

A. PRELIMINARIES

1) MINI BATCH K-MEANS METHOD

K-means is one of the popular clustering algorithms. With the increase in dataset size, the computation time increases as the entire data needs to be present in the main memory [23]. Because of this, we prefer Mini Batch K-means for large datasets. We intend to apply a fixed size of small random batches of data for easy storage in the memory. In every iteration, the cluster is updated taking new random samples from the dataset. For a given dataset $D = x_1, x_2, x_3, \ldots.., x_p$, $x_i \in \mathbb{R}^{m*n}$, x_i represents the records in an n-dimensional real vector. The number of records in dataset D is 'm'. We obtain a set S of cluster center $s \in \mathbb{R}^n$ to decrease over the dataset D of records $s \in \mathbb{R}^{m*n}$ as shown in the following function.

$$Min \sum_{x \in T} \|f(S, x) - x\|^2 \tag{1}$$

where, f(S,x) yields the nearest cluster center $s \in S$ to record x. If K is the number of clusters, it is given by $k = |S|$. We randomly select K records by using Kmeans++ to initialize the centers and we set the cluster centers S to be equal to the values of these. In our case, we have considered the number of clusters equal to the number of class. When the data is huge, the convergence rate of the original Kmeans significantly drops. In this case, an improved K-means called Mini Batch Kmeans is introduced [24].

2) NORMALIZED MUTUAL INFORMATION (NMI)

NMI is one of the ways for measuring the criteria of cluster quality, which is information-theoretic interpretation. This measure calculates the cluster quality with cluster number. Mathematically:

$$NMI(\Omega, S) = \frac{MI(\Omega; S)}{[G(\Omega) + G(S))]/2} \tag{2}$$

where Ω is the set of clusters and S is the set of classes. Here MI is given by the formula:

$$MI(\Omega; S) = \sum_k \sum_j P(d_k \cap s_j) log \frac{P(d_k \cap s_j)}{P(d_k)P(s_j)} \tag{3}$$

where $P(d_k)$ =probability of document in cluster d_k,*

$P(s_j)$ = probability of document in cluster s_j, *

$P(d_k \cap s_j)$ =probability of document being in the convergence of d_k and s_j. *

NMI increases the knowledge of the class by evaluating the amount of information obtained from the clusters. The value is 0 when the clustering is random concerning the class and gives no knowledge about the class. MI reaches maximum value if it perfectly recreates the classes. G is the entropy. Mathematically:

$$G(\Omega) = -\sum_k P(d_k) log P(d_k) \tag{4}$$

This gives the entropy of cluster levels. The normalization in Eqn. 2 by the denominator solves the problem of purity. It also formalizes that fewer clusters are better since the entropy usually increases with the increase in cluster number. [26] The value of NMI is always between 0 and 1.

B. OUR APPROACH

Here, we present our proposed hybrid filter-wrapper approach for the FS. There are two objective functions in our FS. First, the feature ranking function based on the filter approach and second, the selection of optimal features based upon the rankings. This optimal selection is a wrapper based method that depends upon the outcome of the learning algorithm. Our approach is independent of any number of a class labels and is suitable to use with any classifier. In our experiments, we have considered Random Forest as the classifier. However, we can use any classifier. The implemented code can be accessed from the project repository [1]. Our approach has 2 phases;

1) FEATURE RANKING

In the first phase, the main idea is to separately cluster the features one by one based upon the total classes in the dataset. Our objective is to have a selection algorithm which takes less computation time in comparison to the existing algorithms. Since the data are large these days, we have considered mini-batch K-means, which takes into account a batch of data and performs clustering. The computation time, in this case, is much lesser than the normal K-means clustering. The cluster's quality is the metric to find the relation of that feature with the class. As the cluster quality increases,

[1]https://github.com/thejasgs/HFS

the feature tends to be more relevant and is considered to be more important. The use of NMI gives a cluster score between 0 to 1. The high ranking score indicates better classification using the candidate feature. The cluster score for all the features is evaluated separately. Comparing the score of each feature, we obtain the ranking list. This ranking obtained is based upon the individual relationship between the candidate attribute and the class label.

2) FEATURE SELECTION

In the FS problem, a feature variable may have a dependency on other variables. The dependent features tend to produce imbalanced results when acted upon together and hence, is considered a redundant feature. The redundant feature tends to deteriorate the classification process so we remove those features. We considered the ranking obtained from the first phase as the base for the selection of features. We consider this to have a linear approach of selecting the features to get the optimal features in minimum time. When the feature size in the dataset increases, comparison with all the possible subsets is an impractical approach and seems to be computationally very expensive. We present two approaches for the selection of features:

1) Feature Inclusion: This is almost a linear selection approach where the ranked features from phase one are added one by one into the subset. If the addition of the features enhances the classification accuracy, we consider the feature or else we discard the feature. Here, the highest ranked feature is initially included in the list as shown in step one of algorithm 1. We add the next ranked feature and obtain its performance. If the performance increases, we add the feature into the list or else discard the feature. The feature is removed if it does not perform well with the selected subset, considering that it is redundant as it degrades the classification model. This process loops for all the features, as shown in algorithm 1. This process is named MiniBatch K-Means Normalized Mutual Information Feature Inclusion (**KNFI**)
2) Least Ranked Feature Exclusion: This is a linear elimination approach where the least ranked features are eliminated one by one from the entire set of features. Initially, the list consists of all the features and the classification accuracy is calculated for the entire list. Then, in every loop, one least ranked feature is removed from the list. This process is carried out until the list becomes empty. The highest performance among all the iterations is considered as the outcome of our approach, as shown in algorithm 2. This process is named Mini-Batch K-Means Normalized Mutual Information least ranked Feature Exclusion (**KNFE**)

Algorithm 1 Ranking Based Feature Inclusion for Optimal Feature Subset(KNFI)

```
Input: Set of ranked features $S = \{f_0, f_1, f_2, ......f_m\}$, where
    m = total number of features, obtained from the feature
    ranking phase, $f_0$ is the highest ranked feature and $f_m$ is
    the least ranked feature.
Output: prints the selected set of features
    Initialisation :
 1: Lst = S[0] prev=0
    LOOP Process
 2: for k = 0 to m-1 do
 3:   x_tst = x_tst [ Lst ]
 4:   x_tr =x_tr [ Lst ]
 5:   train the model based on any classifier and store the
      accuracy on acc
 6:   if acc > prev then
 7:     if $(k \neq m - 1)$ then
 8:       Add S[ k + 1 ] into the Lst
 9:       prev=acc
10:     else
11:       Print Lst
12:     end if
13:   else
14:     Remove S [ k ] object from the Lst
15:     if $(k \neq m - 1)$ then
16:       Add S[ k + 1 ] to the Lst
17:     else
18:       Print Lst
19:     end if
20:   end if
21: end for
22: return Lst
```

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

The conduction of all the experiments is performed in Python Language using the python libraries. Florida International University provided us the required hardware. We used an Intel i7 4 core CPU with 16GB RAM. Also for large datasets, we used the Flounder Server (AMD Opteron Processor 6380 with 64 cores and 504GB RAM.

B. DATASETS

Here we have considered nine datasets from the ML Repository of UCI [27], three-click fraud datasets, one Intrusion Detection dataset, and Sonar dataset. We have tested upon two versons of TalkingData datset. The information of these datasets is given below. We selected fifteen datasets having different number of features, instances, and classes. Also, we have considered both binary as well as multiclass datasets, which are shown in Table 1 and Table 2 respectively.

C. PREPROCESSING

UNSW_NB15 Dataset It is an intrusion detection dataset that takes into consideration the instances of both the normal activities and the attack activities. To avoid overfitting due to a large number of normal activities, we have removed the normal activity instances. Initially, the data was in 4 different CSV files. We merged all the CSV files into a

TABLE 1. Binary datasets used in experiment.

Dataset	Features	Instances
UNSW_NB15 [28]	47	2,540,047
TalkingData(version 1) [29]	9	1,000,000
TalkingData(version 2) [29]	9	913,692
Criteo [30]	39	756,554
Avazu [31]	16	1,000,000
Ionosphere	34	351
Breast_Cancer [27]	10	699
Spambase [27]	57	4,601
Sonar [32]	60	208

TABLE 2. MultiClass datasets used in experiment.

Dataset	Features	# Classes	Instances
UNSW_NB15 [28]	47	9	2,540,047
Lung_Cancer [27]	56	3	32
Lymphographic [27]	18	4	148
Iris [27]	4	3	150
Heart Disease [27]	13	5	303
Abalone [27]	8	28	4,177

Algorithm 2 Ranking Based Feature Elimination(KNFE)

```
Input: Set of ranked features $S = \{f_0, f_1, f_2, ......f_m\}$, where
    m = total number of features,,$f_0$ is the least ranked feature
    and $f_m$ is the highest ranked feature.
Output: prints the result for every eliminated feature from
    the feature list
    Initialization :
 1: Lst = S prev=0
    LOOP Process
 2: for k = 0 to m-1 do
 3:     x_tst = x_tst [ Lst ]
 4:     x_tr =x_tr [ Lst ]
 5:     //train the model based on any classifier and store the
        accuracy on acc
 6:     //print the result along with the evaluation metrics
 7:     if acc > prev then
 8:         prev=acc // to store the greatest accuracy
 9:         fet=i // to store the no. of feature eliminated
10:     end if
11:     delete Lst[0] //deleting the least ranked feature
12: end for
13: return
```

single dataset and performed the experiments. We removed the socket information(i.e., source ip address, source port number, destination ip address and destination port number) such that model becomes independent of them. We removed the white spaces present in some of the multiclass labels. All the categorical values were converted to the numerical values as the classifier can only learn numerical values. The different ranges of numerical data in the features become a challenge for the classifier to train the model [21]. To compensate this, we performed normalization on the entire data.

1) TalkingData dataset

It is an AdTracking Fraud Dataset [33] which has records of 200 million clicks over four days. It has features like app ID, os, IP address, click time, device type, channel, attributed time, and target label as is_attributed. In the preprocessing stage, we dropped the attributed time. We separated Click time into separate columns, i.e., day, hour, minute, and second. Two variants of the above mentioned dataset were used. In the first version, we considered one million rows of data in which the ratio of classes match the ratio at 200 million rows (Talkingdata Version 1) is taken. 913692 data samples were used for the second variant, where the rows were equally categorized into two classes (Talkingdata Version 2) [29].

2) Avazu

This dataset is a Click fraud dataset consisting of clicks recorded over ten days and has features like id, click (Target Label), device_id, device_ip, an hour of click, and so on. We do the preprocessing i.e., separation of the 'hour of click' column into separate columns. We consider 1 million rows of data in which the ratio of classes match the ratio at 200 million rows to reduce the data size

3) Criteo

It is a Click fraud dataset that consists of 40 features. To clean the data, we have removed instances with 'NaN' values.

4) Ionosphere Dataset

In the Ionosphere dataset provided UCI repository, we converted the class labels (*good*, *bad*) into numerical values.

5) Breast Cancer, Lung Cance, Heart Disease datasets

In this dataset, there are some missing values represented by a question mark(?). We removed the instances containing ? as a cleaning process.

6) Lymphography Dataset, Iris Dataset

These datasets were clean, and no preprocessing step had to be applied. However, we performed resampling as the instances with the same classes were together in the actual dataset.

7) Abalone Dataset

In this dataset, the first feature consists of categorical string values that we converted into numerical values.

8) Spambase Dataset, Sonar dataset

These datasets are considered to compare our model with other research approaches. The spambase dataset is taken from UCI repository, and Sonar dataset is taken from Kaggle dataset. The datasets were clean with no NaN values, and no preprocessing was needed.

We normalized the entire data by using MinMaxScalar function for all the datasets.

V. RESULTS AND DISCUSSION

A. BASE CLASSIFIER: RANDOM FOREST

RF is a prevalent supervised ML technique that is flexible and very easy to use [34]. As the name implies, RF has a large number of individual decision trees. Each decision tree acts as an individual classifier. We get a class prediction from each tree in the RF, and the class that gets the most votes becomes the model prediction of RF. With the increase in the number of trees, the classifier has a greater ability to resist noise and obtain greater accuracy. The RF, being a simple classifier built on decision trees, can easily adapt to large changes in the data size, having the benefit of scalability [35].

B. EVALUATION METRICS

The accuracy of the algorithm needs to be evaluated by certain standard metrics. For binary classification, we have considered the standard metric, Area Under Curve (AUC) and also the F1 Score, which is computed based upon the Precision and Recall score. For the multiclass dataset, we have considered the F1 Score as the evaluation criteria. The F1 Score can also be obtained from the confusion matrix. This metric can only be used for the test data whose true values are already known such that we get a confusion matrix.

We can obtain the following information from the confusion matrix:

- True Positive (TrPos): model correctly predicting Positive cases as Positive.
- False Positive (FlPos): model incorrectly predicting the Negative cases as Positive.
- False Negative (FlNeg): model incorrectly predicting positive cases as Negative.
- True Negative (TrNeg): model correctly predicting negative cases as positive.

Precision score(Pr): It measures accuracy based upon correctly predicted cases.

$$Pr = \frac{TrPos}{TrPos + FlPos} \tag{5}$$

Recall score(RC): It is the TrPos rate to predict the ofteness of predicting positive.

$$RC = \frac{TrPos}{TrPos + FlNeg} \tag{6}$$

F1 Score(F1): F1 is the weighted average of recall and precision of each class.

$$F1 = 2\left(\frac{Pr * RC}{Pr + RC}\right) \tag{7}$$

ROC-AUC curve is a standard metric to measure the performance of the classification model. The probability curve between the true positive rates against false positive rates is referred to as ROC. AUC represents the degree of separability. The higher the AUC, the more the efficiency of the model.

C. ANALYSIS METHOD

To empirically test the advantages and disadvantages of our method, we performed several experiments on real-world datasets with four different approaches. They are:

1) Considering all the features present in the dataset for classification and calculation of its accuracy, AUC (for binary datasets), precision, recall, F-1 Score. We represent this as **AF**.
2) Our approach (**KNFI**), where we perform classification based on the ranked features and determine its evaluation metrics. Without the need of the user to specify the number of optimal features, our approach automatically calculates it. This number has been considered as the base number for performing RFE, where we explicitly have to provide the required number of optimal features.
3) Using RFE (Recursive Feature Elimination), a standard process, provided by Scikit_learn [36], selects features by recursively considering the small set of features. The user explicitly has to give the desired subset number (k), and then it returns the best accuracy from the best subset with k features. In our experiment, we have considered the value of K, referring to our KNFI approach.
4) Our second approach **KNFE**, where we remove the least ranked features one after another, performing the classification and calculating its evaluation metrics. The best accuracy obtained after removing 'k' features is considered as the comparing value with other methods.

A comparative analysis is performed for the results obtained from the four methods in terms of various evaluation metrics, as mentioned above. We can observe that our approach takes less computation time compared to the existing methods, and in many datasets, it produced better results.

D. DISCUSSIONS

1) FOR BINARY DATASETS

In the UNSW_NB15 dataset, both our KNFI and KNFE methods improvised the learning algorithm to obtain greater accuracy, AUC, and F1-Score, as shown in Table 3. KNFI selected 17 features and stood superior in terms of all the evaluation metrics. Also, the evaluation metrics greatly increased in the Ionosphere dataset as in Table 4 for our 6 selected features among the 34 features. Most of the redundant features were removed, giving us better results.

We have a slight increase in accuracy for the Avazu dataset as in Table 5 for both of our approaches. However, the AUC is slightly decreased in both the methods. The decrease in AUC could be due to the presence of imbalanced data. The F1-Score is a much better metric of measurement [37]. The F1-Score remained constant with an increase in accuracy, giving us a better-trained model with the selected features.

[2]Ftr is the number of selected features.

[3]Acc is the accuracy of the model.

TABLE 3. Experimental results of UNSW_NB15 binary datasets.

Method	Ftr	Acc	AUC	F1
AF	43	99.93	99.46	99.93
KNFI	17	**99.963**	99.614	99.96
RFE	17	99.960	99.612	99.96
KNFE	-6	**99.944**	99.96	99.94

TABLE 4. Experimental results of ionosphere datasets.

Method	Ftr	Acc	AUC	F1
AF	34	92.96	90.91	92.84
KNFI	6	**97.18**	95.23	97.14
RFE	6	91.54	7.92	91.55
KNFE	-7	**95.77**	94.238	95.74

TABLE 5. Experimental results of avazu dataset.

Method	Ftr.	Acc	AUC	F1
AF	25	83.029	54.235	77.89
KNFI	7	**83.4375**	53.283	77.89
RFE	7	83.075	53.013	77.63
KNFE	-17	**83.381**	52.456	77.36

TABLE 6. Experimental results of talking dataset version 2.

Method	Ftr.	Acc	AUC	F1
AF	9	99.9179	99.9179	99.92
KNFI	4	**99.919**	99.919	99.92
RFE	4	99.919	99.919	99.92
KNFE	0	**99.9179**	99.917	99.92

TABLE 7. Experimental results of spambase dataset.

Method	Ftr.	Acc	AUC	F1
AF	57	98.04	97.69	98.04
KNFI	15	**97.82**	97.52	97.82
RFE	15	97.285	96.69	97.27
KNFE	-3	**98.58**	98.301	98.93

This is shown in Table 5. Also, in the TalkingData dataset (version 2), the accuracy increased slightly for KNFI. However, for KNFE, it showed zero elimination of features for the best classification accuracy meaning all the features are independent and contributing for the classification model.

In the Spambase dataset, our KNFE approach enhanced the classification accuracy along with all the evaluation metrics by removing three redundant features. From KNFI approach, the accuracy slightly reduced, taking least prediction time and performed well in comparision to RFE. This is shown in Table 7. Also, in the Sonar dataset, KNFE method outperformed all other approaches by removing nine redundant features. Our KNFI approach also gave better results compared to the AF and the RFE methods, as shown in Table 8. The relevance of the features in Sonar Dataset is shown in fig. 2. Some features tend to have very high importance in accordance to the class label and some features tend to have no importance or very low importance in accordance to the class label. We obtain the ranking of the features and then

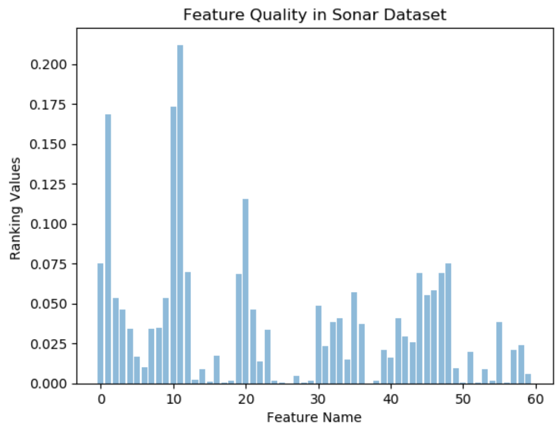

FIGURE 2. Feature ranking for the sonar dataset.

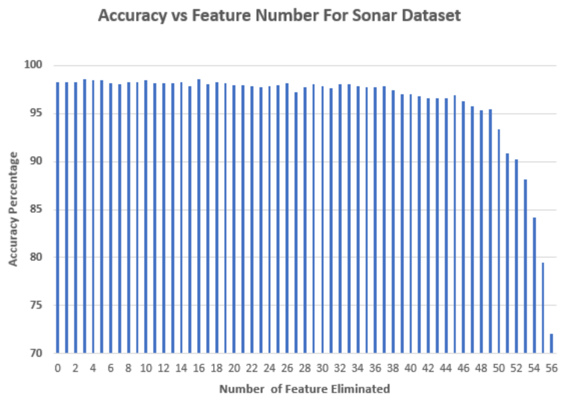

FIGURE 3. Change in accuracy in the KNFE method.

TABLE 8. Experimental results of sonar dataset.

Method	Ftr.	Acc	AUC	F1
AF	60	92.86	93.05	92.88
KNFI	3	**95.24**	95.138	95.24
RFE	3	88.09	88.88	88.16
KNFE	-9	**97.62**	97.91	97.63

preform KNFI and KNFE. In fig. 3, we show the change in the accuracy as we eliminate the least ranked features one at a time. There is a drastic decrease in accuracy as we eliminate large number of features. For a particular number of features eliminated, we observed the highest accuracy.

However, in the TalkingData (Version 1), Criteo and Breast Cancer datasets shown in Table 9, 10 and 11 respectively, the performance seems to drop when performing KNFI process. However, KNFE gave either better results or the same results. This case appears when all the features tend to contribute to fitting the model. In such a scenario, either few features are removed or zero features are removed as in case the of TalkingData dataset (Table 9). The difference in prediction for AF contribution and zero feature elimination in KNFE is due to the change in the pattern of features provided during the training of data. The performance decreased in KNFI

TABLE 9. Experimental results of talking dataset version 1.

Method	Ftr.	Acc	AUC	F1
AF	8	95.127	91.672	95.08
KNFI	6	**94.252**	90.434	94.14
RFE	6	94.784	91.059	94.67
KNFE	0	**95.20**	91.72	95.11

TABLE 10. Experimental results of criteo dataset.

Method	Ftr.	Acc	AUC	F1
AF	39	73.545	62.386	70.29
KNFI	3	**70.205**	57.725	65.85
RFE	3	70.268	55.902	63.85
KNFE	-5	**73.545**	62.45	70.33

TABLE 11. Experimental results of breast cancer dataset.

Method	Ftr.	Acc	AUC	F1
AF	10	98.540	98.113	98.53
KNFI	4	**97.810**	97.517	97.81
RFE	4	94.890	93.744	94.84
KNFE	-3	**98.540**	98.113	98.53

TABLE 12. Experimental results of UNSW_NB15 dataset.

Method	Ftr.	Acc	F1
AF	43	89.326	88.87
KNFI	16	**90.107**	88.88
RFE	16	89.356	89.02
KNFE	-18	**89.591**	89.02

model. Whenever proper information is not extracted from the FS process, the classification accuracy may be negatively affected. The corealtion of the features also affect the FS process. Furthermore, when the sample size is big, the classifier predicts values well with the entire attributes. Also some datasets tend to perform well with other classifers [38].

2) MULTICLASS DATASETS

In most of the MultiClass datasets, we can observe the positive impact of our KNFI as well as KNFE techniques. In UNSW_NB15 dataset (Table 12), the accuracy increased by 0.781 percent along with the increase in F1 Score. Our model selected 16 out of 43 features to get the most efficient results. Our KNFI method enhanced the accuracy and outperformed all other methods giving us good results.

For the Lung_cancer dataset (Table 13), both our methods doubled the accuracy as well as the F1 Score and took the least prediction time. Similarly, for the Lymphographic dataset (Table 14), our KNFE method gave better results when compared to all the methods.

The Iris Dataset (Table 15) performed well when selecting two of the best features from all the four features. The heart disease dataset (Table 17) had a massive fifteen percent increase in accuracy along with a considerable increase in F1 Score using KNFI. Even KNFE increased the accuracy.

TABLE 13. Experimental results of lung_cancer dataset dataset.

Method	Ftr.	Acc	F1
AF	56	33.333	37.78
KNFI	3	**66.666**	68.25
RFE	3	50.00	52.78
KNFE	-14	**66.666**	68.25

TABLE 14. Experimental results of lymphography dataset dataset.

Method	Ftr.	Acc	F1
AF	18	86.66	85.19
KNFI	2	**90.00**	89.78
RFE	2	76.66	80.00
KNFE	-2	**86.66**	75.17

TABLE 15. Experimental results of iris dataset dataset.

Method	Ftr.	Acc	F1
AF	4	96.666	96.67
KNFI	2	**99.9999**	99.99
RFE	2	99.999	99.999
KNFE	-4	**99.999**	99.999

TABLE 16. Experimental results of heart disease dataset.

Method	Ftr.	Acc	F1
AF	13	41.667	34.60
KNFI	4	**56.667**	51.53
RFE	4	43.333	36.71
KNFE	-11	**51.667**	40.90

TABLE 17. Experimental results of abalone dataset.

Method	Ftr.	Acc	F1
AF	8	24.521	22.86
KNFI	1	**21.650**	20.27
RFE	1	17.344	17.14
KNFE	0	**25.239**	23.61

For the Abalone dataset, our KNFI did not produce improved the performance. However, our KNFE increased the preformance. The dataset contains less number of features and many classes. This makes the prediction of classification much tricky. Also, if additional knowlegde is not obtained from the FS method, it may not increase the performance. [38].

3) OTHER COMPARED WORKS

Other than RFE, we also compared our work with other previous works. In comparison with the previous studies of the UNSW_NB15 dataset, our approach of KNFI produced improved results for binary as well as multiclass datasets. As a preprocessing step, we remove all the instances that have *NaN* values, which decreases the number of instances. This has enhanced the performance of the classifier. When our model is run on this dataset, the efficacy of the predictor increased significantly. These results can be seen in Tables (18 & 19).

TABLE 18. Comparision of accuracy for binary UNSW_NB15 with previous studies.

Study	Method	Accuracy
Zewairi, *et al.*	Deep Learning	98.99
Primartha and Tama	Random Forest	95.5
	Multilayer Perceptron	83.50
Nour, *et al.*	Naive Bayes	79.50
	Linear Regression	83.00
	Expectation-Maximization	77.20
Belouch, *et al.*	Random Tree	86.59
	Naive Bayes	80.40
	RepTree	87.80
	Artificial Neural Network	86.31
	Decision Tree	86.13
Faker, *et al.*	Gradient Boosted Tree	97.92
	Random Forest	98.86
	Deep Neural Network	99.19
Our Work	Random Forest(AF)	99.93
	KNFI	**99.963**
	KNFE	**99.944**

TABLE 19. Comparision of accuracy for UNSW_NB15 multiClass with previous studies.

Study	Method	Accuracy
Belouch, *et al.*	Random Tree	76.21
	Naive Bayes	73.86
	RepTree	79.20
	Artificial Neural Network	78.14
Our Work	Random Forest(AF)	89.326
	KNFI	**90.107**
	KNFE	**89.591**

TABLE 20. Comparision of ionosphere data with previous studies.

Method	# Ftr.	F1	RC	Pr	Acc
Venkatesh *et al.* [14]	15	95.09	94.65	95.70	95.28
HGEFS [13]	n.a.	n.a.	n.a.	n.a	91.33
FSFOA [39]	n.a.	n.a.	n.a.	n.a	95.12
KNFI	6	97.14	97.18	97.29	**97.18**
KNFE	-7	95.74	95.77	95.76	**95.77**

We compared the Ionosphere dataset with the previously existing hybrid feature selection methods. We can observe in Table 20 that both KNFI and KNFE methods produced much better results with greater classification accuracy.

We also compare the Spambase dataset and Sonar dataset with the previous works performed in [16]–[19] in terms of classification accuracy since other evaluation metrics have not been provided. They have calculated the rate of classification for the different number of selected features. As a comparison metric, we have taken the instances with the highest accuracy as presented in their papers. To give comparative analysis, we have also calculated the accuracy using KNFE for the same number of features as provided in the previous papers. Also, we have evaluated using KNFI and KNFE. They are shown in Table 21 and Table 22. Our method outperformed other methods giving us good results. The KNFE(MAX) represents our method without any constraint of number of required features.

TABLE 21. Comparision of Accuracy for Spambase dataset with previous studies.

Ftr. selection method	# features	accuracy
GAMIFS [19]	3	83.50
NMIFS [19]	3	75.8
MIFS [16]	3	78.4
MIFS-U [17]	3	81.2
OFS-MI	3	78.4
KNFE	3	**84.15**
KNFI	15	**97.82**
KNFE(MAX)	54	**98.59**

TABLE 22. Comparision of Accuracy for Sonar dataset with previous studies.

Ftr. selection method	# features	accuracy
NMIFS [19]	15	86.73
MIFS(β=0.5) [16]	15	85.96
MIFS-U($\beta = 0.5$) [17]	15	84.04
HGEFS [13]	N.A.	83.00
FSFOA [39]	N.A	86.98
KNFE	15	**92.85**
KNFI	3	**95.24**
KNFE(MAX)	51	**97.62**

VI. CONCLUSION

This paper presented a new hybrid method taking into consideration the advantages of both filter and wrapper method with no constraint for the user to input the number of features required. In our approach, we used the NMI as a metric to rank the features after clustering by Mini-Batch K Means. Once we obtained the ranked features, we came up with two methods to select the features; the feature inclusion method (KNFI) and feature exclusion method (KNFE). We came up with an algorithm for the feature inclusion method, and in the feature removal method, we removed the least important features to get the best performance accuracy. In most of the datasets, our KNFI method performed well taking least number of features whereas, in datasets with least relationship among the features, our KNFE method produced superior results. For future work, optimizing the time taken to get the selected features would help to reduce time complexity. Also, we can come up with better metrics to get the actual relationships among the features such that the redundant features are eliminated.

REFERENCES

[1] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. Mach. Learn.* Amsterdam, The Netherlands: Elsevier, 1994, pp. 121–129.

[2] N. Kushmerick, "Learning to remove Internet advertisements," in *Proc. 3rd Annu. Conf. Auton. Agents (AGENTS)*. New York, NY, USA: ACM, 1999, pp. 175–181. doi: 10.1145/301136.301186z.

[3] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6371–6385, 2014.

[4] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[5] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13 pp. 27–66, Jan. 2012.

[6] S. Lee, L. T. Vinh, Y.-T. Park, and B. J. d'Auriol, "A novel feature selection method based on normalized mutual information," *Appl. Intell.*, vol. 37, no. 1, pp. 100–120, Jul. 2012.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[8] K. Pushpalatha and A. G. Karegowda, "CFS based feature subset selection for enhancing classification of similar looking food grains- a filter approach," in *Proc. 2nd Int. Conf. Emerg. Comput. Inf. Technol.*, Dec. 2017, pp. 1–6.

[9] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - a filter solution," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2002, pp. 115–122.

[10] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.

[11] L. Jiang, G. Kong, and C. Li, "Wrapper framework for test-cost-sensitive feature selection," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[12] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.

[13] X. Xue, M. Yao, and Z. Wu, "A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm," *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 389–412, Nov. 2018.

[14] B. Venkatesh and J. Anuradha, "A hybrid feature selection approach for handling a high-dimensional data," in *Innovations in Computer Science and Engineering*, H. S. Saini, R. Sayal, A. Govardhan, and R. Buyya, Eds. Singapore: Springer, 2019, pp. 365–373.

[15] S. Sharmin, M. Shoyaib, A. A. Ali, M. A. H. Khan, and O. Chae, "Simultaneous feature selection and discretization based on mutual information," *Pattern Recognit.*, vol. 91, pp. 162–174, Jul. 2019.

[16] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[17] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 143–159, Jan. 2002.

[18] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[19] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[20] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS ONE*, vol. 6, no. 12, 2011, Art. no. e28210.

[21] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proc. ACM Southeast Conf.*, Apr. 2019, pp. 86–93.

[22] J. Zhang, Y. Xiong, and S. Min, "A new hybrid filter/wrapper algorithm for feature selection in classification," *Analytica Chim. Acta*, vol. 1080, pp. 43–54, Nov. 2019.

[23] D. Arthur and S. Vassilvitskii, "How slow is the κ-means method?" in *Proc. Symp. Comput. Geometry*, vol. 6, no. 32, 2006, pp. 1–10.

[24] D. Sculley, "Web-scale κ-means clustering," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 1177–1178.

[25] (May 2019). *Ml: Mini Batch K-Means Clustering Algorithm*. [Online]. Available: https://www.geeksforgeeks.org/ml-mini-batch-k-means-clustering-algorithm/

[26] C. D. Manning, P. Raghavan, and H. Schütze. *Evaluation of Clustering-Introduction to Information Retrieval*. Accessed: May 2, 2019. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

[27] A. Frank. (2010). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[28] *The UNSW-NB15 Dataset Description*. Accessed: May 10, 2019. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[29] G. S. Thejas, K. G. Boroojeni, K. Chandna, I. Bhatia, S. S. Iyengar, and N. R. Sunitha, "Deep learning-based model to fight against ad click fraud," in *Proc. ACM Southeast Conf.*, Apr. 2019, pp. 176–181.

[30] *Display Advertising Challenge*. Accessed: Apr. 15, 2019. [Online]. Available: https://www.kaggle.com/c/criteo-display-ad-challenge/data

[31] *Click-Through Rate Prediction*. Accessed: Apr. 15, 2019. [Online]. Available: https://www.kaggle.com/c/avazu-ctr-prediction/data

[32] D. Anh. (Feb. 2018). *Sonar Data Set*. [Online]. Available: https://www.kaggle.com/adx891/sonar-data-set

[33] *TalkingData AdTracking Fraud Detection Challenge*. Accessed: Jun. 15, 2018. [Online]. Available: https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection

[34] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[35] Y. Liu, "Random forest algorithm in big data environment," *Comput. Model. New Technol.*, vol. 18, no. 12A, pp. 147–151, 2014.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[37] Unknown. (Jan. 1970). *On the Dangers of AUC*. [Online]. Available: http://sandeeptata.blogspot.com/2015/04/on-dangers-of-auc.html

[38] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectronics*, May 2015, pp. 1200–1205.

[39] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature selection using forest optimization algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, Dec. 2016.

G. S. THEJAS received the bachelor's and master's degrees (M.Tech.) in computer science and engineering from the Sri Siddhartha Institute of Technology (SSIT), in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree with the School of Computing and Information Sciences (SCIS), Florida International University (FIU), Miami, FL, USA, under the supervision of Dr. S. S. Iyengar and external guidance of Dr. N. R. Sunitha. He was a Trainee with Defense Research and Development Organization/Electronic and RADAR Development Establishment (DRDO/LRDE). He was an Assistant Professor with the Siddaganga Institute of Technology (SIT) for five years. His areas of research include machine learning, deep learning, cybersecurity, human computer interaction (HCI), and performance optimization using parallel computing. He possesses memberships in ACM and Institution of Engineers (IEI). He is a recipient of the Dissertation Year Fellowship Award at FIU. His research has successfully produced several publications in top journals and conferences, such as ACM, the IEEE, Springer, and MDPI, and has also secured an Indian patent.

SAJAL RAJ JOSHI completed the high school from St. Xavier's College, Maitighar, Kathmandu, Nepal. He held a research internship on Machine Learning at Florida International University under Science Without Borders summer program. He is currently pursuing the bachelor's degree with the Siddaganga Institute of Technology (SIT), Karnataka, India. His areas of interests include machine learning, security, and cloud computing.

S. S. IYENGAR is currently a Distinguished University Professor, a Ryder Professor, and the Director of the School of Computing and Information Sciences, Florida International University, Miami, Florida, USA. He is also a Leading Researcher in the fields of distributed sensor networks, computational robotics, and oceanographic applications, and is perhaps best known for introducing novel data structures and algorithmic techniques for large scale computations in sensor technologies and image processing applications. He has published over 500 research papers and has authored or coauthored 12 textbooks and edited ten others.

He is a member of the European Academy of Sciences and a Fellow of the National Academy of Inventors, the Association of Computing Machinery, the American Association for the Advancement of Science, and the Society for Design and Process Science. He has received the Distinguished Alumnus Award of the Indian Institute of Science. In 1998, he received the IEEE Computer Society's Technical Achievement Award. He is an IEEE Golden Core Member, an IEEE Distinguished Visitor, a SIAM Distinguished Lecturer, and an ACM National Lecturer. In 2006, his paper entitled "A Fast Parallel Thinning Algorithm for the Binary Image Skeletonization" was the most frequently read article in the month of January in the *International Journal of High Performance Computing Applications*. His innovative work called the Brooks-Iyengar Algorithm along with the Prof. R. Brooks from Clemson University is applied in industries and some real-world applications which have led to get IEEE Test of Time Award, in 2019.

N. R. SUNITHA received the B.E. degree from Gulbarga University, the M.S. degree from the Birla Institute of Technology and Science, and the Ph.D. degree from Visveswaraiah Technological University, Belguam. She is currently a Professor with the Department of CS&E, Siddaganga Institute of Technology, India. She has published more than 65 peer-reviewed research articles in leading conferences and journals, such as ACM, Springer, the IEEE, and Elsevier. Her research interests include cryptography and network security, storage area networks, big data processing, industrial automation, and computer security and reliability. She was funded for her research projects from ABB GISL, AICTE, DRDO, IISc, and ICT Skill Development Society in India. She has acquired totally six patents in her research field. Dr. Sunitha possesses membership of personal bodies in the Association of Computing Machinery, USA (ACM), the Indian Society for Technical Education, India (ISTE)—Life Member, the Computer Society of India (CSI), the International Association of Engineers (IAENG), and the Institution of Engineers (FIE). She received the IBM Mentor Award, in 2014. She was a Chairperson in the international conferences, such as Conference on Information Science and Technology Management (CISTM 2007), Conference on Network Security and Applications (CSNA 2010), National Conference on Advances in Computer Applications (NCACA), and International Conference on Advances in Computing (ICAdC 2012). Her biodata included in Marquis Who's Who in Science & Engineering 2010. She is a Reviewer of the journals, such as Elsevier's *Computers and Security* and *International Journal of Network Security* (IJNS).

PRAJWAL BADRINATH received the B.S. degree from REVA University, Bangalore, India, in 2017. He is currently pursuing the M.S. degree with Florida International University, Miami, FL, USA. His research interests include machine learning and time series analysis.