IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 29, 2019, accepted July 11, 2019, date of publication July 17, 2019, date of current version August 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929487

Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach

ADEL BINBUSAYYIS, (Member, IEEE), AND THAVAVEL VAIYAPURI, (Member, IEEE)
College of Computer Science and Engineering, Prince Sattam Bin Abdulaziz University, Al Kharj 11942, Saudi Arabia

Corresponding author: Adel Binbusayyis (a.binbusayyis@psau.edu.sa)

ABSTRACT In today's interconnected era, intrusion detection system (IDS) has the potential to be the frontier of defense against cyberattacks and plays an essential role in achieving security of networking resources and infrastructures. The performance of IDS depends highly on data features. Selecting the most informative features eliminating the redundant and irrelevant features from network traffic data for IDS is still an open research issue. The key impetus of this paper is to identify and benchmark the potential set of features that can characterize network traffic for intrusion detection. In this correspondence, an ensemble approach is proposed. As a first step, the approach applies four different feature evaluation measures, such as correlation, consistency, information, and distance, to select the more crucial features for intrusion detection. Second, it applies the subset combination strategy to merge the output of the four measures and achieve the potential feature set. Along with this, a new framework that adopts the data analytic lifecycle practices is explored to employ the proposed ensemble for building an effective IDS. The effectiveness of the proposed approach is demonstrated by conducting several experiments on four intrusion detection evaluation datasets, namely KDDCup'99, NSL-KDD, UNSW-NB15, and CICIDS2017. The obtained results prove that the proposed approach contributes more potential features compared to the state-of-the-art approaches, leading to achieve a promising performance gain in the detection rate of 3.2%, the false alarm rate of 38%, and the detection time of 12%. Furthermore, ROC and statistical significance are analyzed for the identified feature subset to strongly conform its acceptability as a future benchmark for building an effective IDS.

INDEX TERMS Anomaly intrusion detection, correlation, consistency, data analytic lifecycle, diversity measure, ensemble learning, feature selection, information gain, ReliefF, stability measure.

I. INTRODUCTION

Rapid advances in Internet technologies has reshaped today's global market place enabling even small business to reach out internationally [1]. With this rapid evolution of Internet trends for business, the boundary of corporate networks from the public internet has been blurred opening doors for cyber-attacks [2]. In past 10 years, cyber threat landscape has grown to high scale and have placed all types of business at high risk of cyberattacks [3]. According to cybersecurity prediction 2019 and beyond, it is anticipated that, in coming years artificial intelligence will be used as weapon for cyberattacks [4]. The explosive growth of cyberattacks has spearheaded the organizations to invest more in security solutions to stay up-to-date against these cybersecurity incidents and secure their IT infrastructures.

Security solutions such as authentication, encryption and firewalls play crucial role in securing business data and represent an important first line of defense. But it is reported that the intruders can easily by-pass these techniques and are not effective to thwart all kinds of malicious attacks [5], [6]. Also, many organizations consider effective antivirus software as crucial second line of defense. But they are capable to detect only those attacks whose signature are in the database. A promising alternative of strong nature to circumvent these issues is the use of IDS [7]. Recent reports on cybersecurity implies that IDS is the most important and strong second line of defense to protect network resources and infrastructures from malicious attacks by analyzing the network traffic and user behaviors [8]–[11]. Accordingly, research in this field is gaining momentum.

The associate editor coordinating the review of this manuscript and approving it for publication was Gaurav Somani.

The development of artificial intelligence has made significant impact on the emergence of many IDS based on machine learning techniques. The success of these IDS depends on the quality of data employed in designing machine learning models [12]–[14]. But unfortunately, the network traffic data that are analyzed for attacks are noisy and high dimensional in nature [15]. Hence, Feature selection (FS) method plays a key role not only in pruning off redundant and irrelevant information from network traffic data but also in improving the detection accuracy of an IDS by reducing the chances of the learning model from overfitting. Whilst many works are being carried out in this context, a study to benchmark the most crucial feature set exploiting the advantage of ensemble strategy for IDS is not yet been explored. Our study is unique in this respect. Also, to the best of our knowledge, this study is unique in providing useful insights for both experts and incomers on how to use data analytic life cycle for building an effective IDS.

II. OUR CONTRIBUTIONS

The key contributions of our work are summarized in what follows,

- An ensemble for feature selection process is proposed. Although, ensemble learning is not new, utilizing four different evaluation measures such as consistency, correlation, information and distance within ensemble framework to identify the most informative features is novel and has shown its efficiency.
- This study defines a new framework that adopts the data analytic lifecycle practices to employ the proposed ensemble for building an effective IDS.
- The application of the proposed ensemble on four different benchmark datasets presents compact potential feature subset compared to other state-of-the-art approaches and demonstrates promising performance gain on all metrics. This makes the proposed method more efficient for real-time detection.
- For the first time, this study benchmarks key feature subset for two old and two recent cybersecurity datasets. This can help the researchers who are not working in the field of FS to build an efficient IDS utilizing the benchmark key features to contribute significantly for intrusion detection.

III. RECENT RELATED WORKS

The IDS is a well-explored research area. IDS found in literature follow two different approaches, signature-based and anomaly-based detection [16], [17]. Signature-based detection are also called misuse detection. This approach uses signature of an attack pattern to detect an intrusion and are efficient in detecting known attacks with low false alarm rate (FAR). But they fail or show inefficient results with unknown or new attacks [16]. Alternatively, Anomaly-based detection uses machine learning methods and network traffic features to model the normal network activities and detects attack from the activity that deviates from normal pattern [17]; thus, this approach is efficient in detecting unknown and new attacks. But they are flawed with high FAR due to its inability to define the boundary between normal and anomaly activities.

The ability of anomaly-based IDS to detect new attacks has attracted increasing interest in research and has been widely studied in literature [14], [18]. Recently, many computational and artificial intelligence algorithms with optimization techniques are investigated to improve the efficiency of anomaly-based IDS [6], [13], [19]. But the effectiveness of these intelligence algorithms is highly dependent on network traffic feature set that can accurately characterize network traffic for intrusion detection. Several previous studies have shown that FS can greatly improve the accuracy of intelligence algorithms [20]. In the sequel, extensive studies are conducted in the past decades on FS for IDS. The main goal of this section is to present a review of the most recent related works on FS as follows,

FS approaches for IDS are divided into three categories: filter, wrapper and embedded [20]. The filter methods select features based on intrinsic characteristics of the training data without interaction with the learning algorithm. Very few methods in this category have been recently introduced for IDS. For example, Ambusaidi *et al.* [21] introduced a new mutual information (MI) based filter FS method to analytically select optimal features for intrusion detection. Zhao *et al.* [22] came up with a new MI-based FS method considering three factors such as relationship between features and classes, the impact between features and classes, and redundancy between features. Though this category of FS methods are computationally faster and can be easily scaled up with the growth of network traffic data, it has not been explored in the past decade. The key reason is that the success of these methods depend on the evaluation criterion used to determine the importance of features and is of great challenge to select one such best evaluation criterion for a given problem.

Wrappers methods on other hand uses search techniques to generate the feature subsets and evaluates the fitness of generated feature subset using learning algorithms. These methods perform better in enhancing the performance of learning algorithm accuracy. Several recent studies have proposed wrapper-based FS methods for IDS. Most of these studies have employed optimization techniques such as genetic algorithm [23], [24],cuttlefish optimization [25], particle swarm optimization [26], multi-objective optimization [27] methods to handle the various combinations of the features in the network traffic and select the most informative features for network traffic characterization. The major drawbacks with this category of FS methods are that they are prone to overfitting, requires high computation resources to reach convergence and are intractable with large datasets.

Embedded methods perform FS as a part of the training process and uses the property of learning algorithm to guide feature evaluation. Recently, Hamed *et al.* [15] proposed a new embedded based FS method called recursive feature

addition. This method proposes bigram technique for feature selection with new evaluation measure combining FAR, accuracy and detection rate. More interest is not shown towards this category of FS as they are learning algorithm specific and they are based on greedy mechanism considering only the top-ranked features for classification.

Consequently, Hybrid methods emerged in the development of FS for IDS to combat the above mentioned drawbacks. These methods combine filter and wrapper methods to inherit the advantage of both methods and improve the prediction efficiency with better computational requirement. For instance, in [28], Jianglong Song et al attempts to integrate chi-square with RF to design hybrid FS method for Intrusion detection. Similarly, in [29], authors have attempted to define two different schemes, first by integrating information gain(IG) filter with Naïve Bayes classifier and second with decision tree. Recently, a hybrid method cascading the linear correlation coefficient algorithm with cuttlefish algorithm was defined using decision tree as a classifier [29]. The major setback with this category of methods is that the performance of the wrapper method depends on the performance of filter method as they cascaded in the hybrid method. In other words, the wrapper method are constrained to work only on those features that are provided by filter method. Here there is chance for informative features to be screened out and not be considered for wrapper evaluation.

With the development of FS algorithms, the researchers in the field of IDS face challenge in making right choice of FS method from available algorithms. Ensemble strategy is latest development in FS methods and is recommended as a solution to address this problem as it combines the output of several FS methods to improve the performance of the underlying problem [14]. Doing so the researchers are relieved from the task of choosing one best method for their problem. Also, this strategy provides more stable and robust FS performance with high dimensional and large dataset. Ensembles of FS methods for IDS have also been studied in recent years. Akashdeep et al in [30] have attempted to ensemble two filter methods namely IG and correlation filter to identify useful features and have claimed that results with the application of ensemble methods were promising. The same way, the study in [31] has attempted to ensemble four filter FS methods such as chi-square, gain ratio, IG and ReliefF to obtain an optimal feature selection and the results have shown very promising performance in detecting DDOS attacks. The major drawback with these two studies is that the authors have attempted to ensemble only univariate filter methods which does not consider the interaction amongst features.

Recently, an ensemble method in [32] have attempted to ensemble one MI-based filter method and two MI-based wrapper methods designed using decision tree and Naïve Bayes respectively. The key concern of this approach is twofold, first high computational requirements due to two wrappers in the ensemble and second, all the three FS methods in the ensemble are based on information measure. Driven by motivation to propose efficient FS methods in this correspondence, the present work aims to ensemble filter FS methods based on four different evaluation measures namely, correlation, consistency, information, and distance. Though the success of the ensemble approach relies on the stability and diversity of the base selectors, it has been overlooked by all the previous studies on intrusion detection. This work has taken efforts to conduct preliminary experiments and determine these measures for choosing the most appropriate FS methods for the proposed ensemble.

IV. PROPOSED WORK

The core focus of our work is twofold: First, leverage data analytics lifecycle practices stated in [33] and define a framework for building a successful IDS in the era of Big Data. Second, to exploit the advantage of ensemble strategy and identify the most informative features for intrusion detection. The subsection below describes our work in detail.

A. DATA ANALYTIC FRAMEWORK FOR BUILDING IDS

The proposed framework comprises five key components to guide and implement the key phases of data analytic lifecycle [33], namely data discovery, data preparation, model planning, model building, and model evaluation for successful completion of IDS. Each of these components play crucial role and adds valuable impact on IDS model performance. Fig.1 demonstrates our framework for building IDS. For better understanding of the reader, we follow top-down presentation approach to introduce first the key components of the framework and treating the related ensemble-based FS approach as a black-box.

1) DATA DISCOVERY

Data selection is extremely significant task because the choice of correct dataset decides the credibility of the model evaluation. Greater is the reliability of the model if its evaluated with more accurate data. Unfortunately, real network traffic data is unavailable due to the organization' privacy and security issues. Different methods are adopted for dataset collection such as simulated dataset, sanitized dataset, testbed dataset and standard dataset. But the application of first three methods in the context of intrusion detection is very difficult and introduces complication. For example, simulation method for generating traffic data is very hard job, sanitized method is very risky, testbed is very costly and time consuming. Therefore, the publicly available standard datasets are used by many researchers for benchmarking real evaluation of the performance of newly developed IDS [34], [35]. Nevertheless, the key issue with standard datasets is the lack of traffic diversity, sufficient number and types of recent traffic attack styles. Therefore, to achieve fair and rational performance evaluation, this work uses two old datasets namely, KDDCup [36] and NSL-KDD [37], and two recently released benchmark datasets, UNSW-NB15 [38] and CICIDS2017 [39]. A brief overview of these benchmark datasets is discussed in Section-V(A) and statistics of the utilized datasets is reported in Table-1.

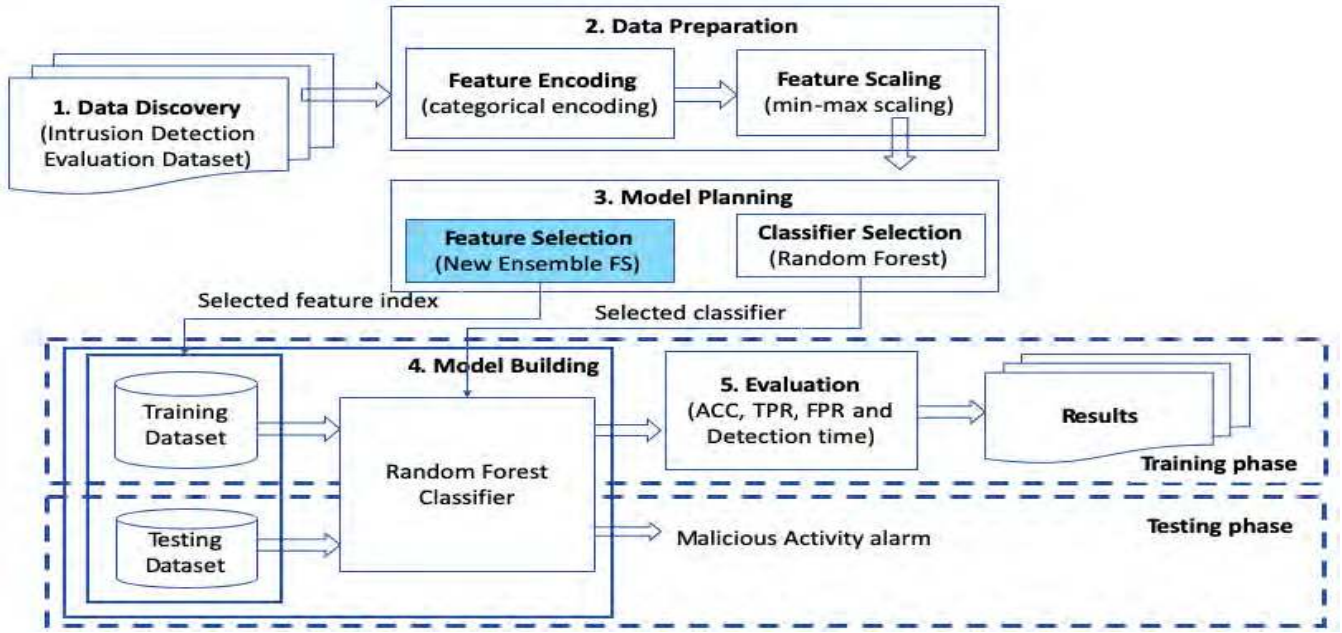


FIGURE 1. Proposed data analytic framework for building IDS.

TABLE 1. Statistics of the utilized benchmark datasets.

Dataset	Number of Attributes	Number of Instances	Number of Attack Types
KDDCup' 99 [36]	41	805049	4
NSL-KDD [37]	41	148517	4
UNSW-NB15 [38]	43	257673	9
CICIDS2017 [39]	78	230092	14

2) DATA PREPARATION

Data preparation is an essential step to enhance data efficiency and detective ability of the IDS model. In the proposed framework, data preparation involves the following two main tasks:

- **Feature encoding:** is a process of mapping nonnumeric features to numeric values. The dataset used in the field of intrusion detection usually contain continuous, discrete and symbolic features. Most of the machine learning algorithms are designed to operate only with numeric values and are therefore incompatible with symbolic features. Hence, it is mandate to use an encoding scheme to map all symbolic features into numeric values.
  The two most commonly used schemes are Label encoding and One Hot encoding. The Label encoder transforms a symbolic feature with C categories by just mapping each category of the symbolic feature with an integer value ranging from 0 to C-1. Whereas, One Hot encoder transforms a symbolic feature with C categories by creating a set of C indicator variables and uses 1 to represent the occurrence of the respective category and 0 for absence of other categories. Thus, these indicator variables created for each category greatly increases the dimensionality of dataset and affects the performance of machine learning algorithms due to curse of dimensionality. Also, these indicator variables give importance for each feature level rather than each feature. Therein, this component of the proposed framework supports Label encoding for symbolic feature transformation.
  The symbolic features in KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 are {protocol_type, service, flag, attack type}, {protocol_type, service, flag, attack type}, {xProt, xServ, xState, attack type}, {attack type} respectively. For example, the 'protocol_type' feature in KDDCup'99 contains three categories such as 'icmp', 'tcp', and 'udp'. By label encoding, these three categories are replaced with a numeric integer value as: tcp: 0; udp: 1; icmp: 2. In the same way, {service, flag, attack type} in KDDCup'99, {protocol_type, service, flag, attack type} in NSL-KDD, {xProt, xServ, xState, attack type} in UNSW-NB15 and {attack type} in CICIDS2017 are also represented by numeric integer values. The application of label encoding to the feature ''service'' in KDDCup'99 might lead to biased result due to large number of category values (high cardinality). To combat this problem, feature scaling is essential and is discussed in the step that follows next.
- **Feature scaling:** is process of adjusting values of the data to a specific range and reduce the complexity of handling data with different range. This allows each feature in the dataset to contribute proportionately and improve the stability of the detection model from being biased by any features. Generally, the features collected from network traffic have different scales, different distributions and sometimes outliers. Hence, feature

scaling is an essential step of preprocessing after encoding all symbolic features into numeric integer values. For example: the feature 'duration' and 'src_bytes' in KDDCup'99 dataset are continuous features with different ranges. This may cause the features with large numerical values to dominate other feature in the detection process. Therefore, this component of the framework employs min-max normalization for adjusting the data values of all features in the [0,1] range using the formula given below,

$$X_i^* = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Here X_{min} and X_{max} are minimum and maximum value of i^{th} feature, and X_i is the original feature value and X^* is the feature value in the range [0,1].

3) MODEL PLANNING

Model planning is a vital step concerned with data exploration for selecting the key features and the best machine learning technique based on the end goal of the study. Thus, the two main tasks of this step are briefed in what follows,

- **Feature selection (FS):** is the process of identifying the most informative features eliminating the redundant and irrelevant features as much as possible. It is not only vital in improving the performance and accuracy of the detection model but also it helps to reduce the data acquisition cost and time in the future by minimizing the number of features required to achieve competitive detection accuracy in conjunction with the right model. For instance, the number of features in the NSL-KDD and CICIDS2017 are 41 and 78 respectively. In a network environment monitoring all these features of network traffic is computationally expensive and might fail to detect malicious activity. Thus, identifying the most informative features that can be used to detect all type of malicious activities is an open challenge in building an effective IDS. The proposed framework combats this challenge by employing a new ensemble FS method defined in next section.
- **Model selection** In this context, Model selection is a task of choosing the best classifier or identify a list of candidate classifier for building effective IDS. Some commonly used machine learning techniques are Logistic Regression (LR), Linear discriminant analysis (LDA), K-nearest neighbor (KNN), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF). In literature, empirical methods such as cross validation (CV) are extensively employed for model selection or parameter tuning of the model [35]. Thus, CV is essential to observe the variation in model performance across k-folds. In other words, it shows how the performance of the model varies from fold to fold. If the variation is low then, the model will tend to be stable and can be considered as best model for the study.

Accordingly, this component of the framework employs CV to the given datasets and enables to select the best classifier for building an IDS model. For illustration, a preliminary experiment was conducted applying CV with 10-folds on the old NSL-KDD and the recent UNSW-NB15 dataset to select the best classifier among the above mentioned seven common machine learning techniques. The obtained results are shown in Fig.2.

It can be clearly observed from Fig.2 that LDA and KNN shows better accuracy than other models for UNSW-NB-15 But the larger variability in accuracy with LDA and KNN indicate that these models have failed to maintain the stability of the accuracy across the CV folds. On other hand, less variability in accuracy with RF indicates its stability in producing the accuracy closer to 98% in all the 10-folds than its counter parts for both datasets. Hence from the results, it can be conformed that RF is the best and suitable model for building IDS and it is considered in this work for all experimental evaluations.

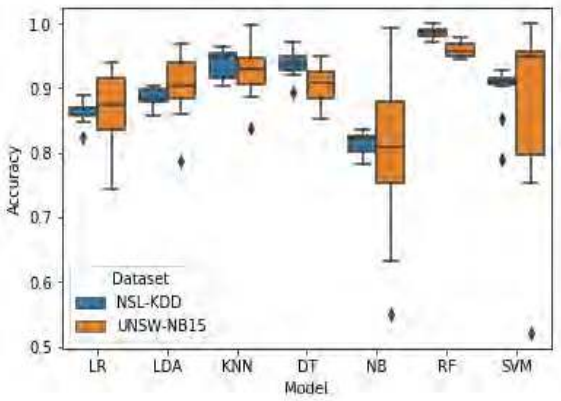


FIGURE 2. Model selection for IDS.

4) MODEL BUILDING

The core function of an IDS is to classify an activity into normal and intrusion based on the significant features identified in the previous phase. The objective of this phase is to build an IDS model using the best classifier selected in the previous phase. This comprises two key phases: training and testing. In training phase, the model is trained with both malicious and normal traffic data packets available in the training datasets along with the corresponding target class labels to learn and fit the parameters of the chosen classifier. In testing phase, the trained model is allowed to detect the target class labels for the traffic data packets available in testing dataset and the model is evaluated for its performance.

5) MODEL EVALUATION

The performance of an IDS is evaluated for its ability to correctly classify the given network traffic data packet as malicious or normal. A good IDS should pose high accuracy

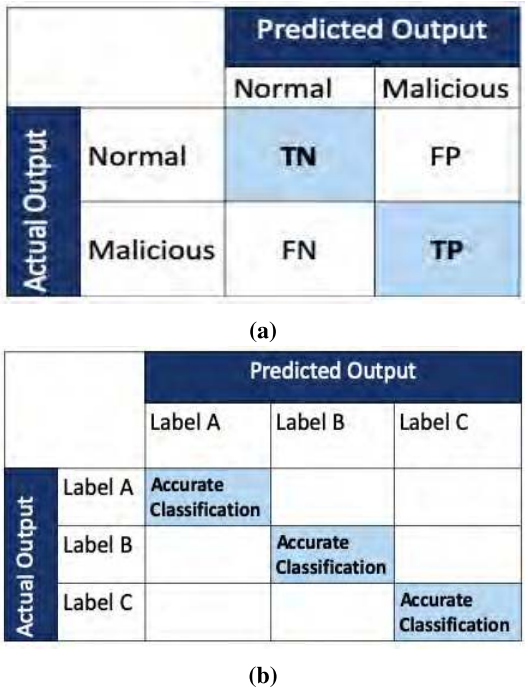


FIGURE 3. Confusion matrix for (A) binary and (B) multiLabel classification.

and detection rate with low FAR. In this regard, the current work uses confusion matrix given in Fig.3 to compute these three metrics as follows,

- **Detection rate (DR):** Also called True Positive Rate is defined as the ratio of number of network traffic data packet detected correctly by the IDS to the total number of network traffic data packets in the testing dataset.
- **False positive rate:** also termed as false alarm rate (FAR), it is the ratio of the number of normal packets detected as malicious packets (FP) to the total normal packets in the testing dataset. If this metric value increases consistently, it may cause the network administrator to deliberately ignore the system warnings. Consequently, this may put the entire network into a dangerous stage. Therein, this metric value should be kept as low as possible.
- **Accuracy (ACC):** can be defined as the proportion of the total number of the correct classification (detection) of malicious (TN) and normal packet (TP) to the actual size of testing dataset.

Another most important metric required to evaluate the efficiency of the IDS is the time taken to classify a network traffic packet. Because, if time taken is high then the cause is twofold, first the attackers may detect the presence of IDS and may try to paralyze it. Second, it may lead to packet loss. For these two reasons, this metric value should be kept as low as possible and is calculated as total time taken to classify all traffic data packets in test dataset divided by actual size of test dataset.

B. PROPOSED ENSEMBLE FS APPROACH

With the increasing number of FS methods in literature, the researchers who are not working in the area of feature selection will face a prime challenge in selecting the appropriate FS method for building an efficient IDS. Ensemble method that combines the output of multiple models instead of applying one single method is one of the optimal solution to confront this problem. In accordance to this, the present study proposes an ensemble of FS methods with the aim of obtaining the most informative features for intrusion detection than those resulting from single methods. The workflow of the proposed ensemble FS approach is shown in Fig.4. This approach comprises two steps: 1) Construction of Ensemble Components and Combining the Ensemble Components. These two steps are briefed below

1) CONSTRUCTION OF ENSEMBLE COMPONENTS

The key focus of this step is to create a set of different ensemble components. Therein from the available suite of filter methods, a set of filter methods based on four different evaluation measure is created in this study to ensure the

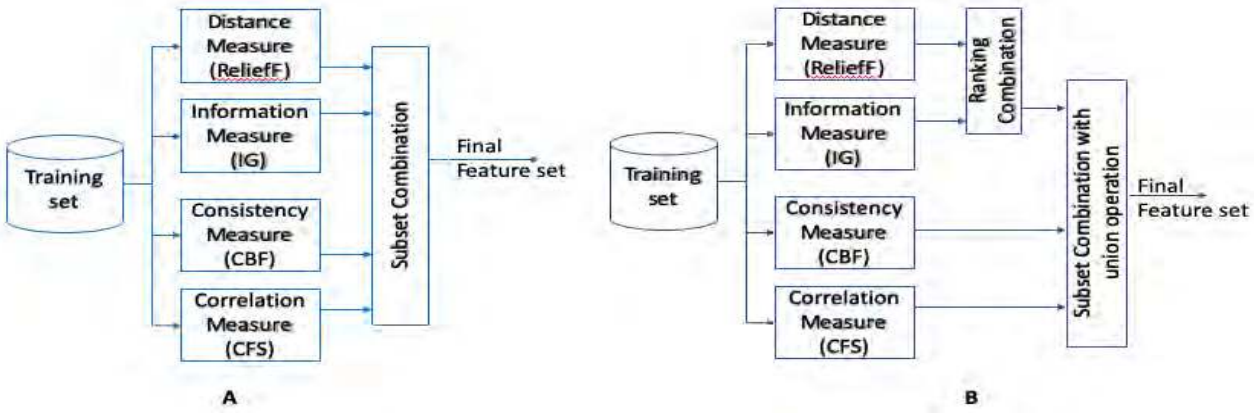


FIGURE 4. Proposed ensemble approach using only subset combination (A) and using ranking with subset combination.

diversity of the proposed ensemble method. Brief theoretical background of these four filter methods is provided below.

- **Consistency based Feature Selection (CBF):** This type of filter method evaluates the goodness of a candidate feature subset by computing the level of its consistency to the target attack class as defined in Eq. (2). Also It works in conjunction with search techniques to search through the feature space effectively and find the optimal candidate subset. For example, the filter starts with single feature and continues to search until a small subset of features that has better class consistency than the subset found thus far is reached. Thus, the outcome of consistency filter is a smallest subset that has the same consistency as the full set. In literature, it has proven to be the fast and best filter in removing redundant and irrelevant features with ability to handle noisy dataset. For the algorithm of CBF, please refer to the work by Liu et al [40]. This algorithm has a time complexity of O(N $\cdot M^2$); where N is the number of total instances in the datasets and M represents the number of selected features.

$$Consistency_s = 1 - \frac{\sum_{i=0}^{J} |D_i| - |M_i|}{N} \tag{2}$$

  where s and J, represents the candidate feature subset and the number of distinct combinations of feature values for S_i. $|D_i|$ and $|M_i|$ denotes the number of occurrence and the cardinality of the majority class for the i^{th} feature value in the combination.
- **Correlation based Feature Selection (CFS):** this type of filter method evaluates a subset of features that are highly correlated with target class but not correlated with each other. Thus, this filter is effective in removing irrelevant and redundant features on the grounds that they will have low relationship with target class and will be related with at least one of other features respectively. The best part of this filter is that it utilizes subset heuristic evaluation defined in Eq. (3) to determine the degree to which each individual feature in subset predicts the target class along with the level of inter-correlation among other features in the subset. Thus for a given candidate set of features, it employs search techniques to find the best optimal subset that maximizes the heuristic given below,

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3}$$

  where $\overline{r_{cf}}$ and $\overline{r_{ff}}$ represents the average feature-class correlation and feature-feature intercorrelation. $Merit_s$ denotes the heuristic merits of a feature subset S with k features. This work utilizes the algorithm defined by Hall [41] for CFS with time complexity of O(N $\cdot M^2$);
- **Information Gain based Feature Selection (IG):** This type of filter performs individual feature evaluation based on the quantity of information it shares to detect the target attack class. Let us represent, F as an feature and C as the class then the entropy of the class before and after observing the feature F is given by [42],

$$HC = -\sum_{c \in C} p(c) log_2 p(c) \tag{4}$$

$$HC|F = -\sum_{f \in F} p(f) \sum_{c \in C} p(c|f) log_2 p(c|f) \tag{5}$$

  Here, the decrease in the entropy of the class indicates the information provided by the feature F and is called Information Gain. Thus IG for a feature F is computed as given below,

$$\begin{aligned} IG &= H(C) - H(C|F), \\ &= H(F) - H(F|C), \\ &= H(F) + H(C) - H(F|C) \end{aligned} \tag{6}$$

  Later, features are sorted in descending order according to the IG value and first M features are chosen to form the subset of important informative features. Thus the feature with higher IG value is deemed to contribute more information required to detect the target attack class. The time complexity of IG is given by O(M $\cdot$ T2), where T2 is the time to calculate the IG;
- **ReliefF Filter:** is the only individual feature evaluation filter that has ability as wrapper methods to capture the feature dependencies in detecting the target attack class. As opposed to wrapper methods, It does not use search methods to capture these dependencies rather uses the concept of nearest neighbors. Therein they are fast when compared to wrapper methods with time complexity O($instances^2.N$)). Also, it has demonstrated its capability in handling missing values with noises and with multiple classes. Most importantly, in contrast to wrapper methods, the features selected by this filter are not classifier dependent. Therein, the selected features can be utilized confidently with different classifiers and save further downstream computational effort when applying ensembles of classifiers in designing IDS.
  The main idea of ReliefF is to estimate the quality features based on the assumption that good quality features will have similar values for instances from same class but different values if the instances are from different classes. For this purpose, Initially, ReliefF assigns a weight W_f to each feature based on how well it contributes to distinguish instances from different and same class. Then it randomly selects an instance R from training set and finds its two nearest neighbor, one from same class called nearest Hit(H) and other from different class called nearest miss(M). Now it updates the weight of each feature using the equation given below [43],

$$W_f = W_f - \left(\frac{diff_f(R, H)}{m} - \frac{diff_f(R, M)}{m} \right) \tag{7}$$

  where diff is the difference between two instances as defined below and is normalized to the range [0, 1].

$$diff_f(I1, I2) = \frac{|value(f, I1) - value(f, I2)|}{max(f) - min(f)} \tag{8}$$

As a result, the weight W_f of the feature increases if it distinguishes the instances from different classes and has same values for instances from same class. The above process is repeated by selecting m random instances from training set.

2) COMBINING THE ENSEMBLE COMPONENTS

This step is concerned with combining the output of the four different FS methods to produce a single final output. In this sense, the following two different kind of combination methods are investigated in the proposed approach,

- **Ranking combination method (RCM):** This kind of combination method combines the output of ranker-based filter methods using reduction function. Some of the reduction function investigated in this work are given in Table-2. These reduction functions produce a single reduced list of features that are ordered according to the calculated relevance value. Here, the higher and lower relevance value indicate the more informative and less informative features respectively.

TABLE 2. Reduction function for RCM.

Function	Description
Min	This reduction function is based on arithmetic operations and selects the minimum of the relevance values yielded by the rankings [44]
Median	This reduction function is based on arithmetic operations and selects the median of the relevance values yielded by the rankings [44]
RRA	This reduction function is based on statistical sorting distributions and uses the Beta distribution to obtain the ρ value [45]

- **Subset Combination method (SCM):** This kind of combination method combines the subset of features without taking into the ranking order [46]. Different types of SCM were used to merge the outputs of CFS, CBF, IG and ReliefF. For example, SCM1 and SCM4 produced the final subset through union and intersection of all the subsets respectively whereas SCM2 and SCM3 produced the final subset by selecting those features that appear in at least two or three subsets respectively.

V. EXPERIMENTAL SETUP

This section provides a detailed description of the datasets we used. Subsequently, a brief description about the design of training and testing dataset design is presented for a comprehensive understanding of the experimental results discussed in Section-VI. At the end, the tools used for the implementation of the proposed approach is discussed.

A. DATASETS

This subsection introduces the cybersecurity datasets utilized in this work for verifying the performance and efficiency of the proposed ensemble approach.

1) KDDCup '99 DATASET

DARPA 1998 was the first dataset that was made publicly available for IDS evaluation. It emanated from MIT Lincoln Lab with 35 days of network traffic traces as PCAP files [47]. Knowledge Discovery and Data Mining (KDD) Cup 99 dataset [36] was derived from DARPA 1998 transforming the network traces in to collection of connections' with large number of different attacks. Since then, both DARPA 1998 and KDDCup '99 became the de facto standard benchmarks for IDS evaluation. KDDCup '99 dataset is available in two forms either as full training dataset or as 10% training dataset. In this work, we use 10% training dataset which contains 494020 records, each characterized by 41 features and a class label to specify whether the connection is normal or an attack type. This dataset includes 22 different types of attacks that can be grouped into four major classes of attack as Denial-of- Service (DoS), unauthorized access to local supervisor privileges (U2R), unauthorized access from a remote machine (R2L), and scanning network to find known vulnerabilities (Probe).

2) NSL-KDD DATASET

In 2009, Tavallaee *et al.* [37] presented a new enhanced version of KDDCup '99 called as NSL-KDD. This provided NSL-KDD dataset resolves the inherent issues in the KDD-Cup '99 such as unreasonable distribution of records, huge number of redundant and duplicate records which would otherwise result in biased evaluation results when being used as an evaluation dataset. Thus, NSL-KDD was valued as most reliable benchmark resource in large number of academic research studies related to IDS evaluation and other security related tasks. Thus after cleansing and removal of redundant records, the NSL-KDD dataset consisted of 257673 records. But as in the KDDCup'99 dataset, each record in NSL-KDD comprised 41 features and a class label to characterize the network flow either as normal or as specific attack type. The distributions of data in these two datasets are shown in Table-3.

TABLE 3. Training and testing set of KDDCup '99 and NSL-KDD.

Attack Class	KDDCup'99		NSL-KDD	
	Train	Test	Train	Test
Normal	97,277	60,593	67,343	9,710
Dos	391,458	229,853	45,927	7,458
Probe	4,107	4,166	11,656	2,422
R2L	1,126	16,189	995	2,887
U2R	52	228	52	67
Total	494,020	311,029	125,973	22,544

3) UNSW-NB15 DATASET

UNSW-NB15 is a recent dataset introduced by the research team of Australian Centre for Cyber Security (ACCS)

in 2015 to reflect a more modern and complex threat environment [38]. This dataset has a hybrid of realistic normal activities and synthetic contemporary attack behavior of live network traffic. The UNSW-NB15 dataset represents nine categories of modern attack families such as Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode and Worms. The dataset contains a rich collection of 49 features involving features from packet headers and payload to effectively discriminate the network packets either as normal or malicious.

The full UNSW-NB15 dataset is available with 2 million and 540,044 connection records logged in four csv files. The original full dataset is partitioned and configured as training and testing dataset for the purpose of IDS evaluation. These partitioned datasets contain only 43 features after the removal of six features from the full dataset. The key advantage of UNSW-NB15 over other existing datasets is that the probability distribution of attack types in the training and testing sets are alike. This means that both training and testing dataset include only 9 attack types and enables the IDS classifier to perform accurately without being biased towards some specific attacks resulting in FAR. The statistics of different attack in the dataset are shown in Table-4.

TABLE 4. Training and testing set of UNSW-NB15.

Attack Class	No. of Instances Training set	No. of Instances Testing set
Normal	56,000	37,000
Fuzzers	18,184	6,062
Analysis	2,000	677
Backdoors	1,746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shell Code	1,133	378
Worms	130	44
Total	175341	82332

4) CICIDS2017 DATASET

This dataset was published by Sharafaldin *et al.* [39] in 2018 at Canadian Institute for Cybersecurity fulfilling the eleven important criteria [48] that are necessary for building a reliable benchmark dataset. The dataset includes contemporary benign and attack scenarios resembling the true real-world network traffic data like the ISCX dataset. Besides, it utilizes 79 features inclusive of class labels for representing six major recent attack profiles. The dataset is created capturing network traffic for five days from Monday to Friday with only recent normal network activities on Monday and injecting different modern attacks on other days. The injected attacks include Brute Force, Botnet, Heartbleed, DoS, DDoS, Web Attack and Infiltration. Considering the computing resource overhead, a subset of this dataset was created as illustrated in Table-5 by randomly choosing 230,092 instances for experimental evaluation.

TABLE 5. Training and testing set of CICIDS2017.

Attack Class	Total instances taken from [23]	60% Train set	40% Test set
Benign	61562	36937	24625
Bot	1966	1180	786
Brute Force	1507	904	603
DDos	58134	34880	23254
Dos GoldenEye	10293	6176	4117
Dos Hulk	10486	6292	4194
Dos slowhttptest	5499	3299	2200
Dos Slowloris	5796	3478	2318
FTP-Patator	7938	4763	3175
Heartbleed	11	7	4
Infiltration	36	22	14
PortScan	60294	36176	24118
SQL	21	13	8
SSH-Patator	5897	3538	2359
XSS	652	391	261
Total	230092	138055	92037

B. TRAINING AND TESTING DATASET

The success of a machine learning algorithm depends on the training and testing data used for model building. Diversified training and testing datasets plays a crucial role in achieving the true performance of the model because it eliminates the model from being biased due to over-fitting or under-fitting of the model to the training data. Taking into account this fact, the training and testing set created by the provider of the benchmarks KDDCup'99, NSL-KDD and UNSW-NB15 are used for model building and evaluation. But the provider of CICIDS2017 did not divide the original dataset into training and testing set. Therefore, random split was employed on the CICIDS2017 dataset to create training and testing set with the split ratio of 60% and 40%. The distribution of records in training and testing datasets of KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 that are used for the present study is reported in the Table-3, Table-4 and Table-5 respectively.

C. TOOLS

In the literature, different tools are used to implement and evaluate IDS. Java, visual C++, C# and WEKA are the most commonly used tools in this context. This work uses Weka version 3.9.2 to perform the classification process with RF. Weka is an opensource software written in Java from University of Waikato, New Zealand. It integrates most of the machine learning techniques for knowledge discovery. R software which is a free software for statistical computing

was used to implement the proposed ensemble FS approach with the ranking combination [45] and also to verify the results given by WEKA.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A series of experiments were conducted to demonstrate the effectiveness of the proposed ensemble approach and benchmark the significant feature subset for the chosen dataset. Indeed, the objectives of these experiments were,

- Evaluate the diversity and stability of the proposed ensemble using any one of the chosen benchmark datasets.
- Apply the proposed ensemble to identify the significant feature subset for the chosen benchmark datasets.
- Verify the performance of the identified feature subset for Multi-label classification using ROC curve and AUC value.
- Analyze the influence of the identified feature subset on IDS building and Detection time for all chosen benchmark datasets
- Verify the statistical significance of the identified feature subset and benchmark subset for the chosen datasets
- Compare the performance of the identified feature subset with feature set reported by state-of-art methods

A. PROPOSED ENSEMBLE METHOD EVALUATION

Conventionally, the performance of an ensemble FS method has been evaluated by measuring the classification accuracy. But in recent years, it is pointed that the performance of an ensemble depends to large extent on two measures such as diversity among the members of the ensemble and the stability of the ensemble to variation in data. Hence, the proposed ensemble approach was evaluated for these measures as described below,

1) DIVERSITY ANALYSIS

Several statistics are proposed in literature to measure diversity. Spearman's rank correlation co-efficient has been commonly used in recent years for diversity analysis [49]. Therefore, this work conducted diversity analysis applying Spearman's rank correlation co-efficient as defined in Eq. (9) [50] to compare the ranking list of the ranker ensemble components such as IG and ReliefF. Here R1, R2, f are the two ranking list and the total number of features respectively.

$$spear(R1, R2) = 1 - \frac{6\sum d^2}{f(f^2 - 1)} \tag{9}$$

While the subset produced by CFS and CBF members of the proposed ensemble were evaluated using Jaccard index (also called as Tanimoto distance) [49] using the equation given below,

$$Jac(R1, R2) = \frac{|R1 \cap R2|}{|R1 \cup R2|} \tag{10}$$

Here $|R1 \cap R2|$ and $|R1 \cup R2|$ represents the number of common features and the total number of features.

The obtained correlation results between each paired-ranking list is shown in Table-6. Here the values are in the range $[-1, 1]$ with 1 indicating no difference between the paired ranking list. Most of the correlation values in the table are far from 1. This indicates that there is great difference between the compared ranking list. From this analysis experiment, it is clear that the four filter FS methods chosen for the ensemble have ensured enough diversity in their behaviors.

TABLE 6. Diversity analysis of proposed ensemble method.

FS Methods	CFS	CBF	IG	ReliefF
CFS	1	0.27	0.34	0.14
CBF	0.27	1	0.38	0.28
IG	0.34	0.38	1	-0.103
ReliefF	0.14	0.28	-0.103	1

2) STABILITY ANALYSIS

A stability measure requires a similarity measure to determine the commonality between the given pair of feature subsets. This is reflected by most of the stability measures developed in the literature utilizing similarity measures. One of the commonly used stability measure was proposed by Kuncheva [51] with an improved similarity measure incorporating the effect of chance for consistency. But the key concern in applying this stability measure was that it required the final set of features to be equal in size which in some case does not happen. Therefore, the most famous similarity metric Kendall Tau [52] defined below was employed in the present study to measure the stability of the proposed ensemble approach.

$$Kend(R1, R2) = \frac{\sum_{i,j \in P} K_{i,j}(R1, R2)}{N} \tag{11}$$

Here P is the set of unordered pairs of distinct elements in R1 and R2. N is the number of pair combinations in P. $K_{i,j}(R1, R2) = 0$ if i and j are in the same order in R1 and R2 and $K_{i,j}(R1, R2) = 1$ if i and j are in the opposite order in R1 and R2.

For this experiment, five random set of subsamples without replacement from the available NSL-KDD training datasets were created and the proposed ensemble approach was applied on each one of these samples to obtain five feature sets. Then stability measure was computed on the obtained five feature sets and the results are shown in Fig.5 to demonstrate the stability of the proposed ensemble approach.

The observation of the Fig.5 clearly demonstrates that the stability measures are closer to 1 not only for the proposed ensemble but also for all four ensemble components. This ensures that all the four FS methods used in ensemble are stable to data variation. The stability among the ensemble members have also contributed in achieving an stability measure of 0.8 in the proposed ensemble approach.

B. APPLICATION OF PROPOSED ENSEMBLE METHOD

This subsection describes the experiments designed to apply and evaluate different configuration of proposed ensemble

TABLE 7. Application of proposed ensemble approach on KDDCup'99 dataset.

FS Methods	Selected Features	Evaluation Metrics		
		ACC	DR	FAR
NO	ALL 41 Features	99.98	1.000	0.001
CFS	{3 4 5 12 26 30 }	99.95	1	0.001
CBF	{3 5 6 12 23 33 35 40 }	99.97	1	0.000
IG	{5 3 30 4 6 29 35 23 33 34 }	99.98	1	0.001
ReliefF	{40 26 41 29 32 32 35 9 6 39 3 }	99.89	1	0.000
SCM1	{1,3, 4,5,6,9,12,23, 26,29,30,33,34,35,39,40,41}	99.98	0.999	0.000
SCM2	{ 3,4, 5,6, 12, 23, 26, 29, 30,33,35,40}	99.86	0.999	0.000
SCM3	{ 3, 5, 6, 35}	99.97	1	0.000
SCM4	{ 3}	97.17	0.987	0.089
RCM_{min}	{3,4,5,6,12,23,26,29,30,33,35,40}	99.98	1	0.000
RCM_{median}	{3, 4, 5, 6,9,12,23,26,29,30,33,35,40}	99.97	1	0.000
RCM_{RRA}	{3, 4, 5, 6,9,12,23,26,29,30,33,35,40,41}	99.97	1	0.000

TABLE 8. Application of proposed ensemble approach on NSL-KDD dataset.

FS Methods	Selected Features	Evaluation Metrics		
		ACC	DR	FAR
NO	ALL 41 Features	99.95	0.999	0.001
CFS	{4, 5, 6, 12, 26, 30, 37, 38}	99.62	0.994	0.002
CBF	{1, 3, 5, 6, 23, 32, 33, 34, 35, 37, 39 }	99.85	0.998	0.001
IG	{5, 6, 3, 4, 30, 23, 33, 34, 35, 38 }	99.87	0.998	0.002
ReliefF	{5,3, 6,4, 23, 30, 33, 35, 38, 37}	99.81	0.997	0.001
SCM1	{4,5,6,12,26,30,37,38,1,3,23,32,33,34,35,39}	99.89	0.999	0.002
SCM2	{ 3, 4, 5, 6 , 23, 30, 33,34, 35, 37, 38}	99.86	0.998	0.002
SCM3	{ 3, 4, 5, 6 , 23, 30, 33, 35, 37, 38}	99.89	0.998	0.001
SCM4	{ 5,6}	97.2	0.988	0.049
RCM_{Min} RCM_{Median} RCM_{RRA}	{1,3,4,5,6,12,23, 26,30,32,33,34,35,37,38,39}	99.89	0.999	0.001

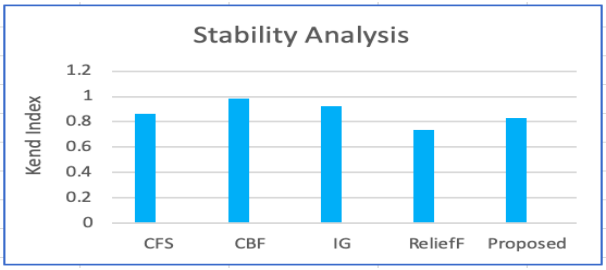


FIGURE 5. Stability analysis of proposed ensemble method.

approach and choose the best feature subset for the utilized benchmark datasets. To accomplish this task, training datasets discussed in Section-V(B) were used to train the chosen RF classifier and test datasets were used to evaluate the trained RF classifier. As first step of this experimental study, CBF and CFS filter was applied with greedy search algorithm to identify the compact set of promising features. While the ranking filters, IG and ReliefF was applied with a threshold of 25% of the total features, sorted in descending order of importance to select the promising features. The features selected by these four filter methods are reported in the rows labelled 'CFS', 'CBF', 'IG' and 'ReliefF' of Table-7 to Table-10 for the four utilized datasets respectively. For reasons of brevity, the indexes of the selected features are presented here. Nevertheless, Readers may refer to appendix for resolving feature index to name.

The second step of this experimental study applied different subset combination strategy to merge the four feature subsets and obtain a single final compact set of features that can best discriminate all attacks of the target class. As stated earlier, this strategy combines the features without considering the ranking order. For instance, the subset combination strategy SCM1 and SCM4 obtains the final compact set through union and intersection of these four subsets whereas the subset combination strategy SCM2 and SCM3 obtains the final subset by combining only those features that appear simultaneously in at least 2 and 3 feature subsets respectively. The obtained final set of features from subset combination strategy are shown in the rows labelled SCM1 to SCM4 of Table-7 to Table-10 for the four utilized datasets respectively.

The Third step of this experimental study applied different ranking combination strategy to merge the ranking lists of ReliefF and IG using the reduction functions given in Table-2. From the resultant list, only those features with score value less than 1 were selected and combined with the subsets of CFS and CBF through union set operation to obtain a single final set of features. The obtained final set of features from ranking combination strategy on the four chosen datasets are

TABLE 9. Application of proposed ensemble approach on UNSW-NB15 dataset.

FS Methods	Selected Features	Evaluation Metrics		
		ACC	DR	FAR
NO	ALL 44 Features	95.98	0.978	0.078
CFS	{11, 12}	92.29	0.997	0.236
CBF	{2,3,4,6,8,9,10,13,14,17,18,19,20,25,26,27,28,29,30,32,34,35,36,37,41,42 }	96.038	0.978	0.081
IG	{8, 9, 11, 12, 32, 10, 13, 28, 4, 42, 36, 7}	95.85	0.974	0.076
ReliefF	{8, 33, 9, 32, 36, 10, 28, 29, 4, 41, 18, 42}	95.63	0.973	0.079
SCM1	{2,3,4,6,8,9,10,13,14,17,18,19,20,25,26,27,28,29,30,32,34,35,36,37,41,42,11,12,7,33}	96.04	0.978	0.077
SCM2	{ 11,12,4,8,9,10,13,18,28,29,32,36,41,42}	95.95	0.975	0.074
SCM3	{ 4, 8, 9, 10, 28, 32, 36, 42 }	95.87	0.974	0.069
SCM4	{ }			
RCM_{Min}	{2,3,4,6,8,9,10,13,14,17,18,19,20,25,26,27,28,29	96.061	0.978	0.077
RCM_{Median}	30,32,33,34,35,36,37,41,42,11,12}			
RCM_{RRA}				

TABLE 10. Application of proposed ensemble approach on CICIDS2017 dataset.

FS Methods	Selected Features	Evaluation Metrics		
		ACC	DR	FAR
NO	ALL 78 Features	99.90	0.999	0.001
CFS	{1,2,16,17,18,19,20,21,22,23,24,25,37,38,53,67,68}	99.93.	0.999	0.001
CBF	{1,13,53,67,68,70 }	99.80	1.000	0.006
IG	{41,53,42,43,56,35,40,19,55,13,66,6,2,7,5,64,24,9,54,68 }	99.23	0.996	0.019
ReliefF	{53,41,64,5,42,43,7,35,56,67,54,9,19,2,15,55,13,40,6,66}	99.58	0.997	0.007
SCM1	{1,2,16,17,18,19,20,21,22,23,24,25,37,38,53,67,68 13,70,41,42,43,56,35,40,55,66,6,7,5,64,9,54,15}	99.91	0.999	0.001
SCM2	{1,2,13,19,53,67,68,24,41,42,43,56 35,40,55,66,6,7,5,64,9,54}	99.88	0.999	0.002
SCM3	{ 2, 13, 19, 53, 67, 68}	99.88	0.999	0.002
SCM4	{53}	94.92	0.984	0.146
RCM_{Min}	{1,2,16,17,18,19,20,21,22,23,24,25,37,38,53,67,68,13,70,41,42,63,43,5}	99.92	0.999	0.001
RCM_{Median}	{1,2,16,17,18,19,20,21,22,23,24,25,37,38,53,67,68,13,70,41,42}	99.93	0.999	0.001
RCM_{RRA}				

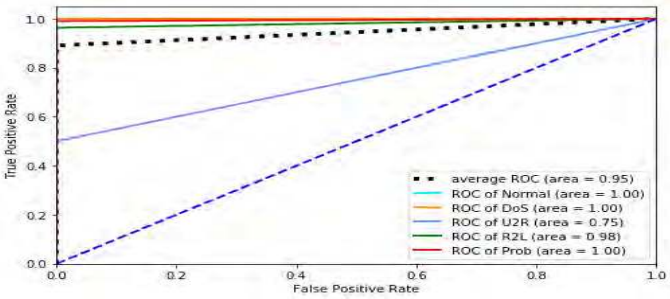
shown in the last three rows of Table-7 to Table-10 respectively. Finally, experiments were conducted to evaluate these feature subsets for security effectiveness with regard to ACC, DR and FAR. The results obtained are reported in column three, four and five respectively for the four utilized datasets from Table-7 to Table-10 respectively.

Analysis of the result on KDDCup'99 in Table-7 indicates that feature subset selected by the proposed ensemble approach with the subset combination (SCM3) and all three ranking combination strategies delivered the best performance with highest DR value of 1.0 and lowest FAR value of 0.0. Also, it can be noted that reliefF and CBF filter demonstrates comparably similar performance. However, among these six cases, the subset combination (SCM3) stands out as best for KDDCup'99 dataset to deliver the best performance with a compact feature subset with four features {3,5,6,35}. Similarly, from the result on NSL-KDD and UNSW-NB15 dataset in Table-8 and Table-9 respectively, it is evident that the subset combination strategy (SCM3) outperforms its counterparts with ten and eight features respectively. Regardless of the complexity of the UNSW-NB15 dataset which comprises variety of modern intrusion attack styles, the proposed ensemble approach with subset combination strategy (SCM3) have proven effective to learn the network traffic flow and deliver an ACC of 95.87, DR of 0.974 and FAR of 0.069. On other hand, results on CICIDS2017 in Table-10 clearly reveals that the ranking combination strategy performs slightly better than subset combination strategy in terms of FAR by 0.001 but with 16 features.

Overall, it is evident that the proposed ensemble approach with subset combination strategy (SCM3) outperforms delivering superior results identifying the best informative compact feature set for intrusion detection on all the chosen datasets. Hence, the feature subsets identified by proposed ensemble approach with subset combination strategy (SCM3) are considered as the most promising feature subset for all chosen dataset and can be taken forward to demonstrate its acceptability as benchmark feature subset.

C. BENCHMARKING THE IDENTIFIED KEY FEATURES

This subsection presents the experiments designed to evaluate the effective performance of the identified feature subset on the four benchmark datasets. First, experiments were conducted to demonstrate the performance of the identified feature subset for discriminating different attacks types (Multi-label classification). Second, experiments were conducted to investigate the significance of the identified feature

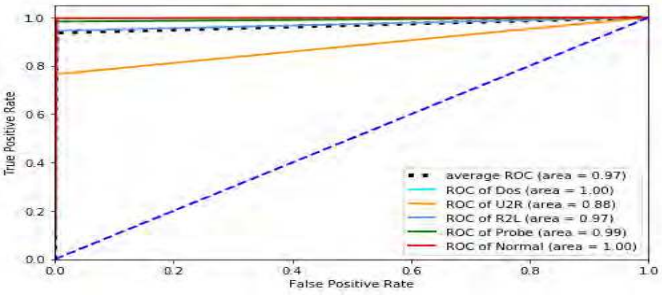


(a)

Classified as →	A	B	C	D	E
A=Normal	97245	11	4	13	4
B=Dos	3	391452	0	0	3
C=U2R	13	0	29	10	0
D=R2L	24	0	6	1096	0
E=Probe	30	47	0	0	4030

(b)

FIGURE 6. (a) ROC and (b) Confusion matrix for multi-label classification using the identified key features of KDDCup'99.

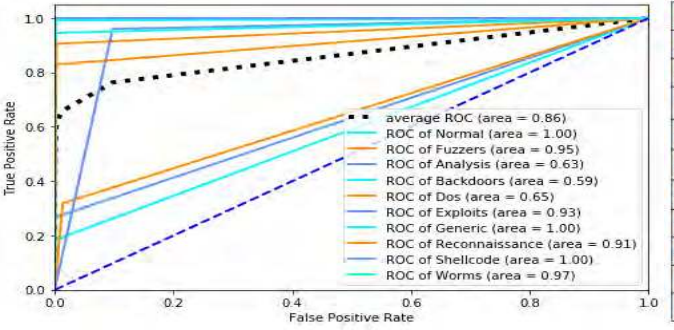


(a)

Classified as →	A	B	C	D	E
A=DoS	45919	0	0	3	5
B=U2R	0	27	0	1	21
C=R2L	0	1	942	0	52
D=Probe	6	0	0	11599	51
E=Normal	9	5	23	19	67287

(b)

FIGURE 7. (a) ROC and (b) confusion matrix for multi-label classification using the identified key features of NSL-KDD.



(a)

Classified as →	A	B	C	D	E	F	G	H	I	J
A=Normal	51827	3596	86	0	33	362	5	55	35	1
B=Fuzzers	3065	13204	12	11	28	14	15	15	98	0
C=Analysis	241	12	259	15	259	1208	0	1	0	0
D=Backdoors	9	20	10	206	259	1216	2	14	7	1
E=Dos	57	104	13	15	2261	9681	18	77	81	8
F=Exploits	376	403	24	32	2635	29029	60	621	135	36
G=Generic	21	45	1	4	91	582	39261	4	12	4
H=Reconnaissance	20	14	1	9	420	2119	4	7891	4	3
I=Shellcode	41	149	0	0	29	133	6	16	772	0
J=Worms	1	10	0	0	2	52	2	1	0	67

(b)

FIGURE 8. (a) ROC and (b) confusion matrix for multi-label classification using the identified key features for UNSW-NB15.

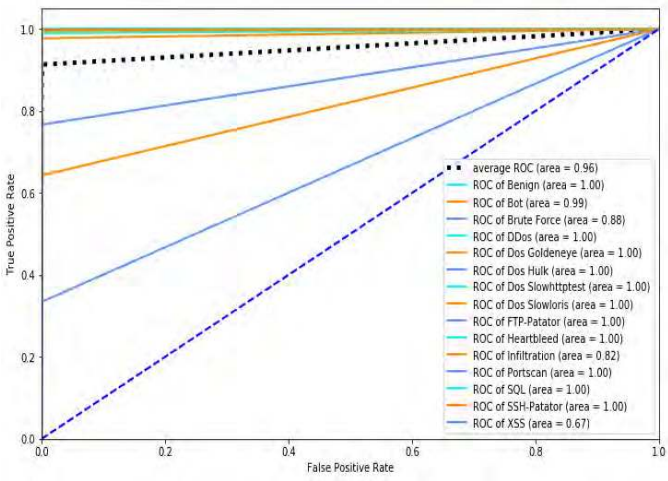
subset on intrusion detection time. Third, experiments were conducted to assess the statistical significance of the identified feature subset for intrusion detection. Finally, experiments were conducted to compare the performance of the identified feature subset with the feature set reported by other state-of-the-art IDS methods.

1) ROC ANALYSIS

The receiver operating characteristics (ROC) curve is a 2D graphs that depict the relative trade-off between DR and error cost (FAR). Also, the area under ROC curve (AUC) is a measure that states how well the IDS is capable of distinguishing an attack from normal traffic. Thus, this performance criterion is one of the most important visualization tool used commonly to determine whether the built IDS is appropriate in terms of cost sensitivity.

The ROC curve for an ideal IDS Model with perfect discrimination will climb towards the upper left corner with highest AUC value of 1. For an IDS model with no better accuracy ROC curve will coincide with diagonal having an AUC value of 0.5. In this work, ROC curve and AUC measure were employed to evaluate the performance of the identified feature subset for discriminating different attack types (multi-label classification) present in the utilized benchmark datasets, KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017.

The ROC and confusion matrix of the four datasets are depicted in Fig.6 to Fig.9 respectively. Analysis of these



(a)

Classified as →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A=Benign	20914	0	17	2	8	5	3	0	0	0	0	2	0	1	4
B=Bot	6	649	0	0	0	0	0	0	0	0	0	0	0	0	0
C=Brute Force	2	0	373	0	0	0	0	0	0	0	0	1	1	0	122
D=DDos	1	0	0	19806	0	0	0	0	0	0	0	0	0	0	0
E=Dos GoldenEye	2	0	0	0	3387	0	0	0	0	0	0	0	0	0	0
F=Dos Hulk	1	0	0	0	0	3630	0	0	0	0	0	0	0	0	0
G=Dos slowhttptest	5	0	0	0	2	0	1861	8	0	0	0	0	0	0	0
H=Dos Slowloris	3	0	1	0	0	0	7	1917	0	0	0	0	0	0	0
I=FTP-Patator	3	0	1	0	0	0	0	0	2641	0	0	0	0	0	0
J=Heartbleed	1	0	0	0	0	0	0	0	0	5	0	0	0	0	0
K=Infiltration	2	0	0	0	0	0	0	0	0	0	9	0	0	0	0
L= PortScan	3	0	0	0	0	0	0	0	0	0	0	20596	0	0	0
M=SQL	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0
N=SSH-Patator	8	0	0	0	0	1	0	0	0	0	0	0	0	1999	0
O=XSS	7	0	115	0	0	0	0	0	0	0	0	0	1	0	91

(b)

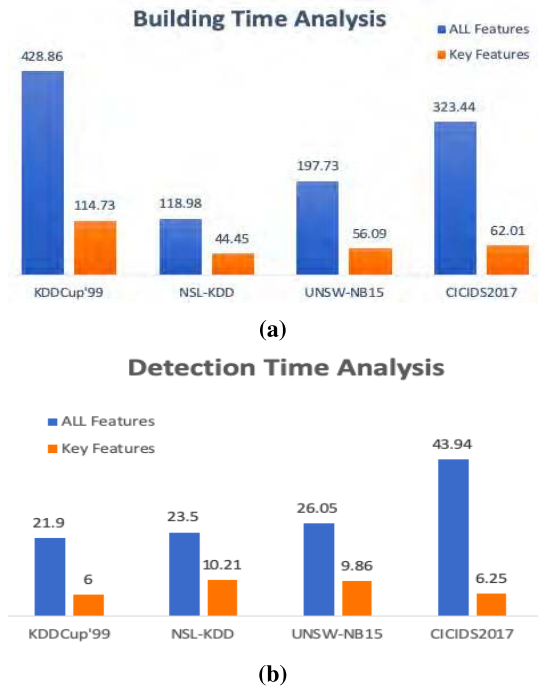
FIGURE 9. (a) ROC and (b) confusion matrix for multi-label classification using the identified key features for CICIDS2017.

figures indicate that the identified feature subsets were significant on all datasets to enable the model learn effectively the network traffic data and deliver promising results with AUC values above 0.95 except for UNSW-NB15 dataset with slight lesser AUC value of 0.86. These AUC values are close to the accuracies that can be obtained from the presented confusion matrix. Also, from the analysis of confusion matrix for UNSW-NB15, it is evident that the reason for the slight lesser AUC value may be due to the imbalanced class distribution in this dataset especially with minority classes such as Analysis, Backdoors and Dos. Nevertheless, the overall results prove the effectiveness and robustness of identified feature subsets in discriminating different attacks from normal network traffic flow.

2) INFLUENCE ON IDS BUILDING AND DETECTION TIME

As discussed earlier, IDS building (training) and detection (testing) time are the most critical metrics of an IDS. For example, only an IDS that takes less time to detect an intrusion with high accuracy can ensure the security of the network. In this regard, experiments were conducted on the four utilized benchmark datasets to investigate the influence of the identified key features for IDS development and detection time. Comparison of IDS building time with the identified key feature subset and all features is illustrated in Fig.10(a). Similarly, the time taken by the IDS built using identified key features and all features to detect an intrusion is illustrated in Fig.10(b).

In concordance with previous studies, the obtained results also clearly demonstrates that compared to retaining all features, the key feature subsets identified by the proposed ensemble significantly reduces both the time required to build an IDS model and time required to detect an intrusion for all the four benchmark datasets. This is because the proposed ensemble method identifies the key features and reduces



(a)

(b)

FIGURE 10. Influence of identified key features on IDS building and detection time.

significantly the number of features required for training the RF classifier. The findings from this analysis suggest that the proposed ensemble method yields significant improvements in IDS building and detection time with the selection of key features.

3) STATISTICAL ANALYSIS

As stated in literature [53], [54], statistical analysis was performed to conform the statistical significance of the identified feature subset. In this regard, ANOVA, one of the most

popular and appropriate hypothesis testing that investigates for the proportion of variance attributed by a feature or group of features to the total variance in the data for discriminating the target class label was considered for statistical analysis. The ANOVA results on the identified feature sets for KDD-Cup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 are reported in Fig.11(a) to Fig.11(d) respectively.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
service	1	21774	21774	201687.1	<2e-16	***
src_bytes	1	1495	1495	13846.9	<2e-16	***
dst_bytes	1	40	40	367.3	<2e-16	***
dst_host_diff_srv_rate	1	1481	1481	13716.0	<2e-16	***
Residuals	494015	53333	0			

(a)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
service	1	32151	32151	4.130e+04	< 2e-16	***
flag	1	114389	114389	1.469e+05	< 2e-16	***
src_bytes	1	5	5	6.279e+00	0.0122	*
dst_bytes	1	20	20	2.593e+01	3.55e-07	***
count	1	108183	108183	1.390e+05	< 2e-16	***
diff_srv_rate	1	1875	1875	2.409e+03	< 2e-16	***
dst_host_srv_count	1	46318	46318	5.950e+04	< 2e-16	***
dst_host_diff_srv_rate	1	18312	18312	2.352e+04	< 2e-16	***
dst_host_srv_diff_host_rate	1	57	57	7.348e+01	< 2e-16	***
dst_host_serror_rate	1	21900	21900	2.813e+04	< 2e-16	***
Residuals	125962	98054	1			

(b)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
xServ	1	948	948	5455.56	< 2e-16	***
sbytes	1	7	7	42.29	7.89e-11	***
dbytes	1	223	223	1280.42	< 2e-16	***
rate	1	4026	4026	23161.30	< 2e-16	***
smean	1	23	23	131.70	< 2e-16	***
ct_srv_src	1	210	210	1210.30	< 2e-16	***
ct_dst_sport_ltm	1	2107	2107	12121.00	< 2e-16	***
ct_srv_dst	1	92	92	530.28	< 2e-16	***
Residuals	175332	30478	0			

(c)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Flow.Duration	1	75014	75014	4272	<2e-16	***
Bwd.Packet.Length.Mean	1	219951	219951	12527	<2e-16	***
Flow.IAT.Max	1	46940	46940	2673	<2e-16	***
Average.Packet.Size	1	38494	38494	2192	<2e-16	***
Init_Win_bytes_forward	1	51094	51094	2910	<2e-16	***
Init_Win_bytes_backward	1	58146	58146	3312	<2e-16	***
Residuals	230085	4039915	18			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(d)

FIGURE 11. Statistical analysis on the identified key features of (a) KDDCup'99 (b) NSL-KDD (c) UNSW-NB15 (d) CICIDS2017.

The reported results reveal that all the identified features for all the utilized benchmark datasets significantly contribute with significant code of "***" for intrusion detection except the feature 'src_bytes' in NSL-KDD dataset which contributes with a significant code of "*". Nevertheless, the results confirm that all the identified features are statistically significant in discriminating the network traffic flow as normal or attack.

4) COMPARATIVE ANALYSIS

To further demonstrate the significance of the identified feature subset for the utilized four benchmark datasets, comparative analysis was conducted with the feature subset reported by the recent literature on the respective datasets in terms of ACC, DR, FAR, Kappa statistic and detection time (DTime). In these experiments, IDS models were built using RF as classifier on the feature subsets reported in the literature on the respective benchmark datasets.

Kappa statistic found by Manel *et al.* [63] was used as one of the comparison measure as it is essential measure than ACC, precision and recall to provide the complete picture of the model performance with multi-class and unbalanced class problem. Since most of the utilized benchmark datasets are unbalanced, kappa statistic was also taken into consideration for comparison. In essence, Kappa statistic is a measure of agreement between predicted class of a dataset and the observed label as ground truth, while correcting the agreement that occurs by chance.

The comparison results on KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 datasets are presented in Table-11 to Table-14 respectively. From these tables, it is evident that the identified feature subset shows consistently promising results in identifying less number of key relevant features and achieving comparably good results on all the utilized benchmark datasets.

For instance, regarding the results presented on KDDCup'99, it can be seen that the identified compact feature subset not only achieves comparable ACC, DR and Kappa statistic results with other models but notably outperforms other models achieving the lowest FAR of 0.0 with reduced DTime. Similarly, the identified feature subset of NSL-KDD and UNSW-NB15 dataset shows comparably good results in terms of ACC, DR and Kappa statistic but with less number of features of 10 and 8 respectively. On other hand, the identified feature subset for CICIDS2017 dataset yielded lower Kappa statistic and higher FAR of 0.002 in contrast to the models reported in [61], [62] though it demonstrate similar DR performance. Overall, it can be confirmed from these results that the proposed ensemble approach has contributed the more potential features leading to achieve an average performance gain in DR of 3.2%, FAR of 38%, DTime of 12% and Kappa statistics of 0.62% together with advantage of reduced number of features.

In summary, it is clear from ROC analysis that the identified feature subset for all the utilized benchmark datasets can precisely and perfectly detect all attacks in network traffic with AUC value of 0.95. The significance of the identified features is also revealed from the computation time and statistical analysis. Further, the identified key features proved to

TABLE 11. Comparative analysis on KDDCup'99 dataset.

Method	Identified Feature set	No. of Features	ACC	DR	FAR	KAPPA Statistic	DTime
[21]	{2, 3, 5, 6, 9, 12, 17, 23, 24, 26, 29, 31, 32, 33, 34, 35, 36, 37, 39}	19	99.98	1	0.001	0.999	0.017
[24]	{2, 3, 4, 6, 8, 10, 12, 17, 22, 23, 24, 26, 27, 33, 35, 37, 38, 39}	18	99.96	1	0.001	0.998	0.017
[55]	{2, 3, 4, 7, 8, 10, 19, 23, 36}	9	99.90	1	0.003	0.996	0.013
[17]	{4, 10, 13, 22, 23, 24, 29, 36, 41, 35 }	10	99.68	0.997	0.005	0.990	0.013
Proposed Ensemble	{3, 5, 6, 35}	4	99.96	1	0.0	0.998	0.007

TABLE 12. Comparative analysis on NSL-KDD dataset.

Method	Identified Feature set	No. of Features	ACC	DR	FAR	KAPPA Statistic	DTime
[21]	{3, 4, 5, 6, 12, 23, 25, 26, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39}	18	99.89	0.999	0.001	0.998	0.018
[56]	{5, 20, 22, 23, 24, 27, 28, 31, 32, 33, 34, 37, 38}	13	99.69	0.996	0.002	0.993	0.014
[57]	{23, 24, 27, 30, 31, 32, 33,34}	8	97.29	0.966	0.021	0.945	0.011
[55]	{3, 4, 7, 8, 10, 12, 30, 35, 36, 37}	10	99.5	0.996	0.006	0.989	0.011
[58]	{1,3,4,5,6,8,10,12,16,19,23,24,25,27,30,32,33,35,37,38,40,41}	22	99.90	0.999	0.001	0.998	0.021
Proposed Ensemble	{3, 4, 5, 6, 23, 30, 33, 35, 37, 38}	10	99.89	0.998	0.001	0.997	0.011

TABLE 13. Comparative analysis on UNSW-NB15 dataset.

Method	Identified Feature set	No. of Features	ACC	DR	FAR	KAPPA Statistic	DTime
[59]	{4, 8, 11, 28, 36}	5	94.56	0.968	0.102	0.873	0.006
[56]	{36, 25, 24, 19, 33, 35, 9, 37, 34, 28, 29, 3, 4, 23, 41}	15	95.83	0.976	0.08	0.902	0.013
[57]	{36, 25, 24, 35, 37, 34, 28, 4}	8	94.96	0.969	0.09	0.881	0.010
[60]	{12, 25, 27, 26, 5, 36, 43, 35, 39}	9	92.86	0.975	0.170	0.829	0.010
Proposed Ensemble	{4, 8, 9, 10, 28, 32, 36, 42}	8	95.87	0.974	0.074	0.904	0.010

TABLE 14. Comparative analysis on CICIDS2017 dataset.

Method	Identified Feature set	No. of Features	ACC	DR	FAR	KAPPA Statistic	DTime
[61]	{1,6,8,13,14,15,24, 25,39,59,60,62,63}	13	99.93	0.999	0.001	0.998	0.017
[62]	{6, 8, 12, 14, 17, 20,25, 26, 27, 28, 30, 37, 38, 39, 43, 47, 48, 52, 53, 54, 64, 67, 68, 71, 78}	25	99.89	0.999	0.001	0.997	0.026
[39]	{2,12,14,64,67}	5	99.40	0.996	0.01	0.986	0.009
Proposed Ensemble	{2, 13, 19, 53, 67, 68}	6	99.88	0.999	0.002	0.996	0.009

be better than those reported by other existing techniques with respect to DR, FAR and DTime. Although experimental studies in real-time scenario are needed to conform the findings, in the light of the results reported above, it is evident that the potential features identified by our proposed ensemble model will stand out to be precise and benchmark feature subset for the utilized intrusion detection datasets in terms of DR, FAR and DTime.

VII. CONCLUSION

The key impetus of present work was to identify and benchmark optimal feature subset for the available benchmark datasets that can maximize the intrusion detection rate while minimizing the FAR and detection time. In achieving this, the present work has proposed an ensemble FS approach. This approach for the first time has attempted to merge four filter methods based on correlation, consistency, information and distance to select the most critical features from the actual available features. In addition, it adopts subset combination strategy to aggregate the features list selected by these four filter methods. Diversity and stability analysis was conducted to show the effectiveness of the proposed ensemble. Later, the application of the proposed approach on four intrusion detection evaluation datasets, namely, KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 exhibited its significance in contributing the more critical feature subset compared to other state-of-the-art approaches, that are accomplished on the same dataset. The ROC and statistical analysis results for the identified feature subsets demonstrated its promising gain in performance especially for classes with fewer instances alleviating the imbalance problem. Experiments on computation time required for training and testing revealed the prominence of the identified feature subset for real-time detection. In the light of the

TABLE 15. Feature list of KDDCup'99 and NSL-KDD dataset.

No.	Feature Name	No.	Feature Name
1	duration	21	is_hot_login
2	protocol_type	22	is_guest_login
3	service	23	count
4	src_bytes	24	serror_rate
5	dst_bytes	25	rerror_rate
6	flag	26	same_srv_rate
7	land	27	diff_srv_rate
8	wrong_fragment	28	srv_count
9	urgent	29	srv_serror_rate
10	hot	30	srv_rerror_rate
11	num_failed_logins	31	srv_diff_host_rate
12	logged_in	32	dst_host_count
13	num_compromised	33	dst_host_srv_count
14	root_shell	34	dst_host_same_srv_rate
15	su_attempted	35	dst_host_diff_srv_rate
16	num_root	36	dst_host_same_src_port_rate
17	num_file_creations	37	dst_host_srv_diff_host_rate
18	num_shells	38	dst_host_serror_rate
19	num_access_files	39	dst_host_srv_serror_rate
20	num_outbound_cmds	40	dst_host_rerror_rate

TABLE 16. Feature list of UNSW-NB15 dataset.

No.	Feature Name	No.	Feature Name
1	Id	23	dtcpb
2	dur	24	dwin
3	xProt	25	tcprtt
4	xServ	26	synack
5	xState	27	ackdat
6	spkts	28	smean
7	dpkts	29	dmean
8	sbytes	30	trans_depth
9	dbytes	31	resp_body_len
10	rate	32	ct_srv_src
11	sttl	33	ct_state_ttl
12	dttl	34	ct_dst_ltm
13	sload	35	ct_src_dport_ltm
14	dload	36	ct_dst_sport_ltm
15	sloss	37	ct_dst_src_ltm
16	dloss	38	is_ftp_login
17	sinpkt	39	ct_ftp_cmd
18	Dinpkt	40	ct_flw_http_mthd
19	sjit	41	ct_src_ltm
20	djit	42	ct_srv_dst
21	swin	43	is_sm_ips_ports
22	stcpb	44	attack_cat

analysis results, it is confirmed that the identified optimal set of features for all the utilized benchmark datasets will not only be effective and capable in improving the detection rate and false alarm rate but also is expected to speed up the detection process with reduced compact set of features. Thus, the proposed model can be a valuable tool in the development of effective IDS and the feature subset identified for the four chosen datasets has the potential to serve as future benchmark for network security research communities.

APPENDIX
UTILIZED BENCHMARK DATASETS

See Tables 15–17.

TABLE 17. Feature list of CICIDS2017 dataset.

No.	Feature Name	No.	Feature Name
1	Destination Port	40	Max Packet Length
2	Flow Duration	41	Packet Length Mean
3	Tot Fwd Packets	42	Packet Length Std
4	Tot Backward Packets	43	Packet Length Variance
5	Tot Len of Fwd Packets	44	FIN Flag Count
6	Tot Len of Bwd Packets	45	SYN Flag Count
7	Fwd Packet Len Max	46	RST Flag Count
8	Fwd Packet Len Min	47	PSH Flag Count
9	Fwd Packet Len Mean	48	ACK Flag Count
10	Fwd Packet Len Std	49	URG Flag Count
11	Bwd Packet Len Max	50	CWE Flag Count
12	Bwd Packet Len Min	51	ECE Flag Count
13	Bwd Packet Len Mean	52	Down/Up Ratio
14	Bwd Packet Len Std	53	Average Packet Size
15	Flow Bytes/s	54	Avg Fwd Segment Size
16	Flow Packets/s	55	Avg Bwd Segment Size
17	Flow IAT Mean	57	Fwd Header Length
18	Flow IAT Std	58	Fwd Avg Bytes/Bulk
19	Flow IAT Max	59	Fwd Avg Packets/Bulk
20	Flow IAT Min	60	Fwd Avg Bulk Rate
21	Fwd IAT Total	61	Bwd Avg Bytes/Bulk
22	Fwd IAT Mean	62	Bwd Avg Packets/Bulk
23	Fwd IAT Std	63	Bwd Avg Bulk Rate
24	Fwd IAT Max	64	Subflow Fwd Packets
25	Fwd IAT Min	65	Subflow Fwd Bytes
26	Bwd IAT Total	66	Subflow Bwd Packets
27	Bwd IAT Mean	67	Subflow Bwd Bytes
28	Bwd IAT Std	68	Init_Win_bytes_forward
29	Bwd IAT Max	69	Init_Win_bytes_backward
30	Bwd IAT Min	70	act_data_pkt_fwd
31	Fwd PSH Flags	71	min_seg_size_forward
32	Bwd PSH Flags	72	Active Mean
33	Fwd URG Flags	73	Active Std
34	Bwd URG Flags	74	Active Max
35	Fwd Header Length	75	Active Min
36	Bwd Header Length	76	Idle Mean
37	Fwd Packets/s	77	Idle Std
38	Bwd Packets/s	78	Idle Max
39	Min Packet Length		

REFERENCES

[1] V. Barhatov, A. Campa, and D. Pletnev, "The impact of Internet-technologies development on small business success in russia," *Procedia-Social Behav. Sci.*, vol. 238, pp. 552–561, Jan. 2018.

[2] M. N. Sadiku, *Emerging Internet-Based Technologies*. Boca Raton, FL, USA: CRC Press, 2019.

[3] C. Castelli. (2018). *Revitalizing Privacy and Trust in a Data-Driven World Key Findings From the Global State of Information Security Survey*. PWC. [Online]. Available: https://www.pwc.com/gsiss

[4] H. Thompson and S. Trilling. (Nov. 2019). *Cyber Security Predictions: 2019 and Beyond*. Symantec Blog. [Online]. Available: https://www.symantec.com/blogs/feature-stories/cyber-security-predictions-2019-and-beyond

[5] S. Anuraj, P. Premalatha, and T. Gireeshkumar, "High speed network intrusion detection system using FPGA," in *Proc. 2nd Int. Conf. Comput. Commun. Technol.* New Delhi, India: Springer, 2016, pp. 187–194.

[6] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Comput. Netw.*, vol. 136, pp. 37–50, May 2018.

[7] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[8] J. E. Rubio, R. Roman, and J. Lopez, "Analysis of cybersecurity threats in Industry 4.0: The case of intrusion detection," in *Proc. Int. Conf. Critical Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2017, pp. 119–130.

[9] D. Kobialka. (Jun. 2017). *Cyber Intrusion Detection Improving*. Trustwave Global Security Report. [Online]. Available: https://www.msspalert.com/cybersecurity-news/trustwave-global-security-report-cyber-intrusion-detection-improving/
[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.
[11] Y. Chen, J. Hong, and C.-C. Liu, "Modeling of intrusion and defense for assessment of cyber security at power substations," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2541–2552, Jul. 2018.
[12] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," *Comput. Secur.*, vol. 30, no. 8, pp. 625–642, 2011.
[13] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, 2010.
[14] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.
[15] T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," *Comput. Secur.*, vol. 73, pp. 137–155, Mar. 2018.
[16] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," in *Future Generation Computer Systems*. Amsterdam, The Netherlands: Elsevier, 2019.
[17] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019.
[18] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 70–91, Jan. 2015.
[19] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
[20] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.
[21] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016.
[22] F. Zhao, J. Zhao, X. Niu, S. Luo, and Y. Xin, "A filter feature selection algorithm based on mutual information for intrusion detection," *Appl. Sci.*, vol. 8, no. 9, p. 1535, 2018.
[23] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Comput. Secur.*, vol. 77, pp. 304–314, Aug. 2018.
[24] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017.
[25] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, 2015.
[26] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, Jul. 2016.
[27] Y. Zhu, J. Liang, J. Chen, and Z. Ming, "An improved NSGA-III algorithm for feature selection used in intrusion detection," *Knowl.-Based Syst.*, vol. 116, pp. 74–85, Jan. 2017.
[28] J. Song, W. Zhao, Q. Liu, and X. Wang, "Hybrid feature selection for supporting lightweight intrusion detection systems," *J. Phys., Conf. Ser.*, vol. 887, no. 1, 2017, Art. no. 012031.
[29] W. Wang, Y. He, J. Liu, and S. Gombault, "Constructing important features from massive network traffic for lightweight intrusion detection," *IET Inf. Secur.*, vol. 9, no. 6, pp. 374–379, 2015.
[30] A. Sharma, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Syst. Appl.*, vol. 88, pp. 249–257, Dec. 2017.
[31] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 130, 2016.
[32] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Comput. Secur.*, vol. 81, pp. 148–155, Mar. 2019.
[33] D. Dietrich, "Data analytics lifecycle processes," U.S. Patent 9 262 493 B1, Feb. 16 2016.
[34] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018.
[35] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[36] *KDD Cup 1999*, Univ. California, Irvine, Irvine, CA, USA, Oct. 2007. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[37] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[38] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.
[39] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.
[40] H. Liu and R. Setiono, "A probabilistic approach to feature selection—A filter solution," in *Proc. ICML*, vol. 96, 1996, pp. 319–327.
[41] M. A. Hall, "Correlation-based feature selection of discrete and numeric class machine learning," Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, Tech. Rep. 8, May 2000.
[42] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[43] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Conf. Mach. Learn.* Springer, 1994, pp. 171–182.
[44] P. Willett, "Combination of similarity rankings using data fusion," *J. Chem. Inf. Model.*, vol. 53, no. 1, pp. 1–10, 2013.
[45] R. Kolde, S. Laur, P. Adler, and J. Vilo, "Robust rank aggregation for gene list integration and meta-analysis," *Bioinformatics*, vol. 28, no. 4, pp. 573–580, 2012.
[46] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "On developing an automatic threshold applied to feature selection ensembles," *Inf. Fusion*, vol. 45, pp. 227–245, Jan. 2019.
[47] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000.
[48] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1–6.
[49] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Inf. Fusion*, vol. 52, pp. 1–12, Dec. 2019.
[50] M. P. Sesmero, J. M. Alonso-Weber, A. Giuliani, G. Armano, and A. Sanchis, "Measuring diversity and accuracy in ANN ensembles," in *Proc. Conf. Spanish Assoc. Artif. Intell.* Springer, 2018, pp. 108–117.
[51] L. I. Kuncheva, "A stability index for feature selection," in *Artificial Intelligence and Applications*. Innsbruck, Austria: IASTED, 2007, pp. 421–427.
[52] A. I. McLeod, "Kendall rank correlation and mann-Kendall trend test," in *R Package Kendall*. London, ON, Canada: Western Univ., 2005.
[53] B. Xiao and I. Benbasat, "An empirical examination of the influence of biased personalized product recommendations on consumers' decision making outcomes," *Decis. Support Syst.*, vol. 110, pp. 46–57, Jun. 2018.
[54] M. Sarstedt and E. Mooi, "Hypothesis testing and anova," in *A Concise Guide to Market Research*. Berlin, Germany: Springer, 2019, pp. 151–208.
[55] W. Wang, X. Du, and N. Wang, "Building a cloud IDS using an efficient feature selection method and SVM," *IEEE Access*, vol. 7, pp. 1345–1354, 2019.
[56] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Trans. Big Data*, to be published.
[57] N. Moustafa, G. Creech, and J. Slay, "Big data analytics for intrusion detection system: Statistical decision-making using finite Dirichlet mixture models," in *Data Analytics and Decision Support for Cybersecurity*. Cham, Switzerland: Springer, 2017, pp. 127–156.

[58] K. Selvakumar, M. Karuppiah, L. SaiRamesh, S. H. Islam, M. M. Hassan, G. Fortino, and K.-K. R. Choo, "Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs," *Inf. Sci.*, vol. 497, pp. 77–90, Sep. 2019.
[59] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1881–1886.
[60] F. Gottwalt, E. Chang, and T. Dillon, "CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques," *Comput. Secur.*, vol. 83, pp. 234–245, Jun. 2019.
[61] Y.-Y. Zhou and G. Cheng, "An efficient network intrusion detection system based on feature selection and ensemble classifier," Apr. 2019, *arXiv:1904.01352*. [Online]. Available: https://arxiv.org/abs/1904.01352
[62] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," *J. Phys., Conf. Ser.*, vol. 1192, no. 1, 2019, Art. no. 012018.
[63] S. Manel, H. C. Williams, and S. J. Ormerod, "Evaluating presence–absence models in ecology: The need to account for prevalence," *J. Appl. Ecol.*, vol. 38, no. 5, pp. 921–931, 2001.

ADEL BINBUSAYYIS is currently an Assistant Professor with the College of Engineering and Computer Science, Prince Sattam Bin Abdulaziz University, where he is a specialist in cybersecurity and technology transfer. He is also the Vice-Dean of e-learning with the Deanship of Information Technology and Distance Learning, Prince Sattam Bin Abdulaziz University. He is also an Advisor of Vice Rector with Prince Sattam Bin Abdulaziz University, where he is responsible for monitoring the performance executions of the university strategic goals.

THAVAVEL VAIYAPURI is currently an Assistant Professor with the College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University. Her research interests include the fields of data science, security, computer vision, and high-performance computing. With nearly 20 years of research and teaching experience, she has published more than 50 research publications in impacted journals and international conferences. She is also a member of the IEEE Computer Society, and also a Fellow of HEA, U.K.