IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MACHINE LEARNING DESIGNS, IMPLEMENTATIONS AND TECHNIQUES

Received December 18, 2019, accepted January 22, 2020, date of current version February 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969428

An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble

BAYU ADHI TAMA[1], LEWIS NKENYEREYE[2], S. M. RIAZUL ISLAM[3], (Member, IEEE), AND KYUNG-SUP KWAK[4], (Member, IEEE)

[1]Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Gyeongbuk 37673, South Korea
[2]Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea
[3]Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea
[4]Department of Information and Communication Engineering, Inha University, Incheon 22212, South Korea

Corresponding authors: Lewis Nkenyereye (nkenyele@sejong.ac.kr) and Kyung-Sup Kwak (kskwak@inha.ac.kr)

This work was supported in part by the National Research Foundation of Korea-Grant funded by the Korean Government (Ministry of Science and ICT)-NRF-2017R1A2B2012337.

ABSTRACT A Web attack protection system is extremely essential in today's information age. Classifier ensembles have been considered for anomaly-based intrusion detection in Web traffic. However, they suffer from an unsatisfactory performance due to a poor ensemble design. This paper proposes a stacked ensemble for anomaly-based intrusion detection systems in a Web application. Unlike a conventional stacking, where some single weak learners are prevalently used, the proposed stacked ensemble is an ensemble architecture, yet its base learners are other ensembles learners, i.e. random forest, gradient boosting machine, and XGBoost. To prove the generalizability of the proposed model, two datasets that are specifically used for attack detection in a Web application, i.e. CSIC-2010v2 and CICIDS-2017 are used in the experiment. Furthermore, the proposed model significantly surpasses existing Web attack detection techniques concerning the accuracy and false positive rate metrics. Validation result on the CICIDS-2017, NSL-KDD, and UNSW-NB15 dataset also ameliorate the ones obtained by some recent techniques. Finally, the performance of all classification algorithms in terms of a two-step statistical significance test is further discussed, providing a value-added contribution to the current literature.

INDEX TERMS Random forest, gradient boosting machine, Web attack, performance benchmark, anomaly-based IDSs, significance tests.

I. INTRODUCTION

In today's information age, every organization attempts to place their business on the Internet. Internet-based applications enable companies to increase their revenue as well as to improve or even redesign their business process, i.e. virtualization in supply chain or adopting futuristic business-to-business (B2B) platform [1]. The Internet has been employed in the last two decades by companies and many organizations worldwide. It helps an organization to place a Web-based application such as e-commerce to offer timely services or getting closer to its customers [2], for instance. Furthermore, it has changed people's life dramatically, in which the users could stay online to communicate with each other anywhere and anytime [3]–[5]. Nowadays, a high-speed Internet has brought a significant contribution to the development of various types of Internet-based computing such as ubiquitous computing [6], cloud computing [7], and mobile cloud computing [8], among others. People are not dependent on on-the-spot computing resources to run the application services, yet various services, i.e. storage, applications, and servers are delivered to the user's computers or devices over the Internet [9].

Apart from several above-mentioned merits, the number of attacks is mushrooming overwhelmingly as the number of Web applications increase [10]. Attackers attempt to make a resource unavailable so that they might take advantage by sending an anomalous request to it [11]. A resource containing a vulnerability might be exploited by the attackers, consequently can jeopardize the confidentiality, integrity, and availability properties of the organization's crucial resources [12]–[14]. This might result in an

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

immeasurable financial loss as well as unrecoverable damage to an organization. There exist several possible attacks in Web applications such as structured query language (SQL) injection [15], CRLF injection, cross-site scripting (XSS) [16], and server-side include [17], [18] to name a few. The injection attack is placed in the top-rank by the Open Web Application Security Project (OWASP) Top 10 Security Vulnerabilities 2013 [19]. It might cause data corruption, lack of accountability, or even denial of service or access. Due to its severe impact and easy exploitability, this type of attack, among others, is at the top risk of security threats.

In contrast to injection attack, XSS is the most common Web application defect, making it fairly easy to be exposed via testing or code analysis [20]. By executing codes in a target's browser, an attacker can either hijack user session, redirect user, vandalize website, or hijack the user's browser using malware [21]. Even though the impact of this attack is moderate, it is necessary to take into account the business value of the affected system and all the data it processes. Previous researches indicate that the number of Web protection systems is still limited due to the lack of datasets. Furthermore, Web traffic data obtained from KDD Cup 99 [22] is not applicable for attack detection in Web applications. Anomaly detection for a Web application by using a stacked ensemble is adopted in this study. We employ CSIC-2010 [23] and CICIDS-2017 [24] dataset which is specifically generated for intrusion detection research in Web-based applications. Three above-mentioned ensembles are chosen due to their distributed processing implementation, which enables an efficient classification task in comparison with other native bagging or boosting techniques, i.e. Adaboost [25] and bagging predictor [26].

Anomaly detection is widely known as the most significant approach in an intrusion detection system since it could detect a novel attack by probing any network profiles which are different from the normal traffic profile [27], [28]. Anomaly detection was firstly proposed by Denning in [29]. Since then, a plethora of anomaly-based IDS has been proposed by many researchers worldwide. For instance, an anomaly detection approach was proposed by Mukkamala *et al,* in [30]. The authors proposed an ensemble approach of three machine learning algorithms, i.e. neural network, support vector machine, and multivariate regression splines. The authors applied the proposed approach to the KDD Cup 99 dataset with accuracy as a performance metric. However, the dataset has received many criticisms from intrusion detection community [10], [22], [28]. Although the dataset contains HTTP requests, it is not proper for current Web attack detection (note that Web attacks have shifted dramatically in the last decade).

We compose the remainder of the paper as follows. Section II reviews the state-of-the-art studies of anomaly detection in Web traffic, whilst Section III provides material and method which includes material (dataset), a brief review of base classifiers and the proposed stacked ensemble. Following this, Section IV presents the experimental results and discussion. Lastly, Section V concludes the paper.

II. RELATED WORK

Regarding anomaly detection in a Web application, the prior works are discussed in chronological order as follows. The first attempt is suggested by the authors in [23] and they proposed an approach that could detect two kinds of attacks, i.e. static and dynamic attack. Generic feature selection (GeFS) is proposed by [31], [32] and employed 30 relevant features of the CSIC-2010 dataset for the detection experiment. The selected features are then used for classification analysis. From their experimental result, the algorithm decision tree (C4.5) achieves superior performance compared to CART, RF, and random tree (RT) either using a full feature set or feature subset. An adaptive intrusion detection system (A-IDS) is introduced in [33]. It is developed based on an ensemble of four different base classifiers, i.e. naive Bayes, Bayes network, decision stump, and radial basis function (RBF) network. The proposed method yields 90.52% of accuracy, which outperforms similar ensemble algorithms, i.e. majority voting and boosting. In [34], an effective algorithm for cyber-attack detection in a Web application is proposed. Similar to the previous works, C4.5 is the top performer in comparison with naive Bayes, Adaboost, and PART.

In [35], a normal class (HTTP request) of Web traffic is designed using a regular expression. It classifies Web application attacks by analyzing HTTP request headers. The proposed method achieves 94.46% of detection rate and 4.34% of FPR, which outperforms the previous work presented in [31]. In the recent work discussed in [36], the authors proposed a one-class meta-earning for anomaly detection in Web traffic (OC-WAD). Such meta-learner is designed using a new binary artificial bee colony algorithm, namely BeeSnips, to reduce the initial ensemble size of a one-class support vector machine. The proposed method results in an ameliorate detection rate and reduced FPR in comparison with [31] and [35].

A one-class SVM for Web traffic anomaly detection is suggested in [37]. By applying on CSIC-2010v2 and CSIC2012, the proposed method has gained a reasonable performance in terms of TPR, FPR, and F_1. Most recent works have been carried out on CICIDS-2017 dataset. The proposed methods include ensemble learning, i.e. [38], [39], and [40]; deep neural networks, i.e. [41], [42], and [43]; and some individual classifiers, i.e. neural network [44], K-nearest neighbor [45], and local outlier factor [46]. Table1 sums up the cutting-edge techniques for Web attack detection in chronologically order.

For the sake of differentiation between this study and prior studies, some remarks from different perspectives are broken down and this paper shares some value-added contributions to the literature as follows:

- We compare the performance of stacked ensemble and its base classifiers, i.e. RF, GBM, and XGBoost for

TABLE 1. Summarization of state-of-the-art techniques for Web traffic anomaly detection.

Study	Year	Technique	Feature selection	Dataset	Performance metric	Significance test
[31], [32]	2011	Decision tree	Yes	ECML/PKDD-2007, CSIC-2010v1	Detection rate, FPR	Not mentioned
[33]	2012	A-IDS (Voting ensemble)	Not mentioned	ECML/PKDD-2007, CSIC-2010v1	Accuracy	Not mentioned
[34]	2014	Decision tree	Not mentioned	CSIC-2010v1	Detection rate, FPR, AUC	Not mentioned
[35]	2014	Regular expression	Not mentioned	CSIC-2010v1	Detection rate, FPR, AUC	Not mentioned
[36]	2015	One-class ensemble	Not mentioned	CSIC-2010v1	Detection rate, FPR, accuracy	ANOVA
[37]	2017	One-class SVM	Yes	CSIC-2010v1, CSIC2012	TPR, FPR, F_1	Not mentioned
[38]	2019	Random forest	Not mentioned	CICIDS-2017	Precision,recall, F_1	Not mentioned
[40]	2019	Ensemble approach	Yes	CICIDS-2017	Accuracy, detection rate, FPR	Not mentioned
[46]	2019	Local outlier factor	Yes	CICIDS-2017	Accuracy, precision, recall, F_1, AUC	Not mentioned
[44]	2019	Neural network	Yes	CICIDS-2017	Accuracy	Not mentioned
[45]	2019	K-nearest neighbors	Yes	CICIDS-2017	Accuracy, precision, recall, F_1	Not mentioned
[43]	2019	Deep neural network	Yes	CICIDS-2017	Accuracy, precision, detection rate, FPR, F_1, AUC	Not mentioned
[42]	2019	Deep neural network	Yes	CICIDS-2017	Accuracy, precision, recall, F_1	Not mentioned
[41]	2019	Recurrent neural network	Not mentioned	CICIDS-2017	Accuracy	Not mentioned
[39]	2019	Random forest	Not mentioned	CICIDS-2017	Precision, recall, F_1	Not mentioned
This study	2019	Stacked ensemble	No	CICIDS-2017, CSIC-2010v2, NSL-KDD, UNSW-NB15	Accuracy, F_1, AUC, FPR	Two-step test

anomaly detection in a Web application. Even though RF has been considered in [31], [32], the number of classifiers used in the ensemble is not clearly defined. Hence, we conduct several experiments to choose the best possible ensemble (tree) size of RF, GBM, and XGBoost. Furthermore, to obtain the optimal hyperparameter setting for each classifier, a *grid* search is also carried out.

- Current literature lacks discussion on whether the performance differences among classifiers are significant or not since there exists only one statistical test reported, which is a parametric test ANOVA [36]. ANOVA requires an underlying statistical distribution of data and it has less power than other parametric tests [47]. In this study, we extensively compare the performance differences among classifiers by conducting several statistical tests.

III. MATERIAL AND METHOD

A. WEB TRAFFIC INTRUSION DATASETS

It is deemed necessary to use an appropriate dataset for anomaly detection in a Web application since the dataset is a representation of specific attacks targeting a Web-based application. Nevertheless, it is not an easy task to obtain a standard dataset for benchmarking since the dataset is not publicly available. The currently available intrusion dataset, KDD Cup 99, is not a real one because it does not represent the actual attack and there is a large number of redundant samples, leading to a biased result for classification. Moreover, the dataset suffers from the deficiency of traffic variability, some known attacks are not covered, and some attacks do not reflect the current evolution. For those reasons, two datasets, i.e. CISC-2010v2 and CICIDS-2017 that are specifically designed for Web traffic anomaly detection are chosen.

Besides, as we suffer from a limited number of datasets for evaluating the proposed classification model, thus for the sake of a reasonable comparison and benchmark, two other widely known intrusion datasets, i.e. NSL-KDD and UNSW-NB15 are also incorporated in the experiment. Both datasets contain network traffic from HTTP request and several attacks targeting a Web application such as Denial of Service (DoS) [48]. We outline the four datasets as follows.

- **CSIC-2010v2** [23]. It is made up of 104,000 legitimate and 119,585 malevolent Web requests sent to an e-commerce application in which a customer can buy items using a shopping cart. The dataset is the improvement version of the prior dataset (CSIC 2010v1)[1], in which some raw samples were not encoded properly.

[1] http://www.isi.csic.es/dataset/

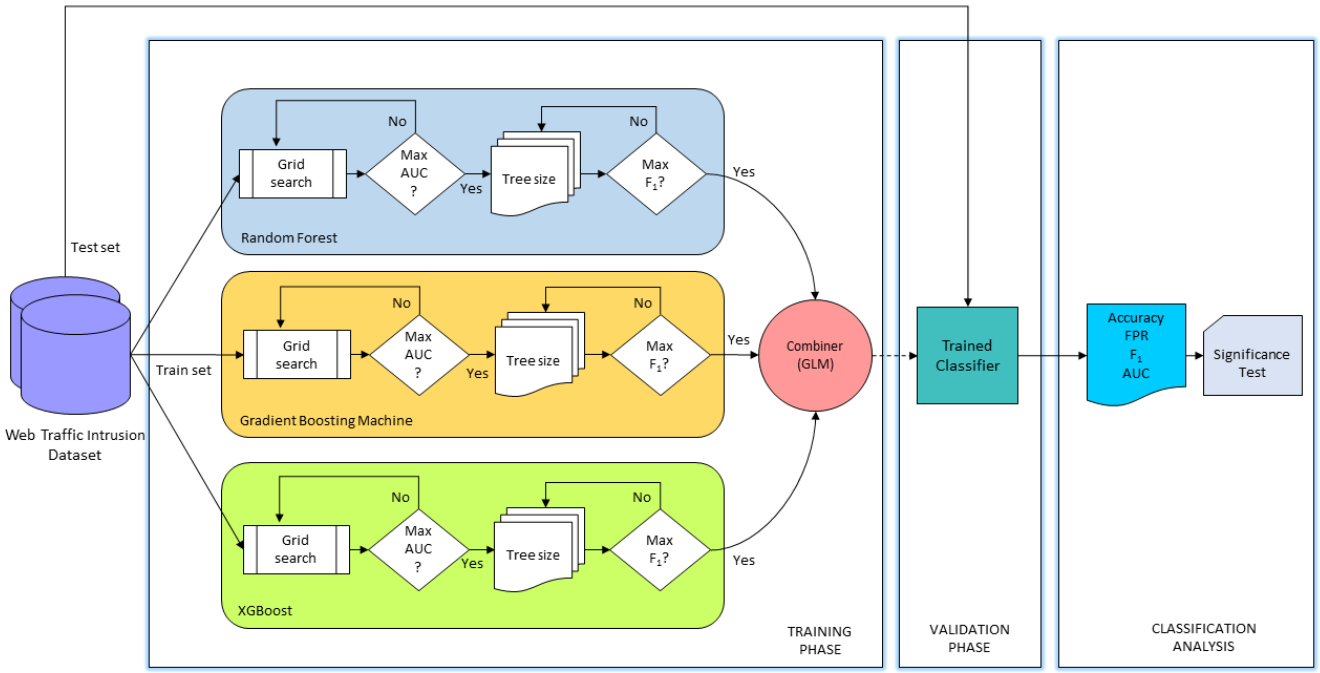

FIGURE 1. Conceptional procedure of Web traffic anomaly detection.

As a result, the former dataset leads to the bias results for some classification algorithms. Due to this reason, we take into account the latter dataset in our benchmark analysis, however, we also report performance accuracy of the best model in our experiment with the existing approaches applied to the former dataset. There exist 17 input features that have been generated from the dataset such as index, method, url, protocol, userAgent, pragma, cacheControl, accept, accepEncoding, acceptCharset, acceptLanguage, host, connection, contentLength, contentType, cookie, and payload.

- **CICIDS-2017** [24]. The dataset is collected by generating realistic traffic using the B-Profile system. It includes benign and current novel attack patterns, which are some characteristics of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols. Numerous attacks are injected in the experiment, leading to some labeled known attacks that are available for IDS research. As our objective is to perform binary classification, all attack types are labeled with malicious. A total of 78 features and 170,366 instances are used in this study, where the number of instances for benign and malicious is 168,186 and 2,180, respectively.
- **NSL-KDD** [22]. It is made up of 42 input variables, whilst 20% of training samples dataset, so-called KDDTrain+, are taken into account for model construction. KDDTrain+ consists of 25,192 samples, with 13,499 anomalous and 11,743 normal samples. In addition, we consider an independent test set, i.e., KDDTest+ (22,544 samples) which is particularly provided for performance evaluation of IDS techniques [22].
- **UNSW-NB15** [48]. This dataset, unlike NSL-KDD, is an original version of an intrusion detection dataset that has appeared more recently. The full training set (UNSW-NB15$_{train}$) is composed of 42 features, with 37,000 samples in the normal class and 45,332 samples in the anomaly class. A specialized testing set (UNSW-NB15$_{test}$) is also used in the experiment. UNSW-NB15$_{test}$ has 175,341 samples.

B. CLASSIFICATION METHODS

1) THEORETICAL WORKFLOW

A theoretical workflow of Web traffic anomaly detection is provided in Figure1. The workflow is made up of three phases, i.e. training the stacked ensembles, validation, and classification analysis. The first phase bears upon the process of designing the structure of the stacked ensemble for detecting Web attack. Following this, the best base model (classifier with optimized hyper-parameter settings and tree-size) for each dataset can be taken into account. Hence, a total of four classifiers are available for further benchmark using the statistical significance test given in Section IV. The first phase is broken down in detail in Section III-B.2.

In the second phase, a validation method, e.g. train-test (hold out) with a ratio 80/20 is adopted for CSIC-2010v2 and CICIDS-2017. We also consider a more reasonable and realistic validation technique using independent test sets for NSL-KDD and UNSW-NB15 dataset, which are hereafter

TABLE 2. Hyperparameter setting for random forest.

Learning parameter	Range value	Optimal hyperparameter value			
		CSIC-2010v2	CICIDS2017	NSL-KDD	UNSW-NB15
sample_rate	$\{0.20, 0.21, ..., 1.00\}$	0.63	0.48	0.48	0.95
max_depth	$\{1, 2, ..., 30\}$	26	8	12	11
col_sample_rate_per_tree	$\{0.20, 0.21, ..., 1.00\}$	0.42	0.84	0.84	0.73
col_sample_rate_change_per_level	$\{0.90, 0.91, ..., 1.10\}$	1.1	1.02	1.02	1.07
min_rows	$2^{\{0,1,...,\log_2(nts)-1\}}$	512	32	32	16
nbins	$2^{\{4,5,...,10\}}$	32	32	32	512
nbins_cats	$2^{\{4,5,...,12\}}$	2048	32	32	256
min_split_improvement	$\{0, ..., 1 \times 10^{-4}\}$	1×10^{-06}	0	0	1×10^{-06}

*nts = number of training samples

TABLE 3. Hyperparameter setting for gradient boosting machine and XGBoost.

Learning parameter	Range value	Optimal hyperparameter value			
		CSIC-2010v2	CICIDS2017	NSL-KDD	UNSW-NB15
sample_rate	$\{0.20, 0.21, ..., 1.00\}$	0.37	0.64	0.57	0.58
max_depth	$\{1, 2, ..., 30\}$	26	7	8	12
col_sample_rate_per_tree	$\{0.20, 0.21, ..., 1.00\}$	0.85	0.89	0.84	0.68
col_sample_rate_chage_per_level	$\{0.90, 0.91, ..., 1.10\}$	1.02	1.02	1.04	1.1
min_rows	$2^{\{0,1,...,\log_2(nts)-1\}}$	4	2	64	1024
nbins	$2^{\{4,5,...,10\}}$	128	16	256	16
nbins_cats	$2^{\{4,5,...,12\}}$	4096	128	16	4096
min_split_improvement	$\{0, ..., 1 \times 10^{-4}\}$	0	0	0	0

*nts = number of training samples

called as KDDTest+ and UNSW-NB15$_{test}$, respectively. This procedure provides a reliable demonstration to what extent the proposed model performs well on independent samples, e.g. unseen data samples. Please note that CSIC-2010v2 and CICIDS-2017 do not provide independent test sets, thus we pick 20% from each original dataset for being used as test sets. Four performance measures that are typically used in anomaly detection research are employed, i.e. accuracy, false positive rate (FPR), F_1, and AUC. Lastly, in the third phase, a comparative study using a statistical test is performed, including an analysis of the classification result.

2) CLASSIFIER STRUCTURE DESIGN

A stacked ensemble trains multiple individual classifier ensembles in a parallel fashion. To build a strong ensemble, we utilize three original homogeneous ensembles, i.e. RF, GBM, and XGBoost. It is worth mentioning that we employed the proposed method in *R* and H_2O package [49]. When we conduct the experimentation for this work, we make of a different number of trees indicating the size of each classifier ensemble, i.e. 50, 100, 150, . . . , 500. For instance, RF-50 is a random forest with 50 trees, GBM-450 is a gradient boosting machine with 450 trees, and so on. The best possible learning parameters of all base classifiers are acquired using *grid* search by trying all possible values as listed in the second column of Table1 and Table 2. A *grid* search is chosen because compared to other search methods, it would provide good coverage of the search space when we deal with a small number of variables [50], [51]. It is noted that we use the same optimal hyperparameters for GBM and XGBoost since both algorithms share the same principle of gradient boosting,

TABLE 4. The best base models used to construct the proposed stack of ensemble w.r.t F_1 metric.

Dataset	Best base model		
	GBM	RF	XGBoost
CICIDS-2017	GBM-450	RF-50	XGB-200
CSIC-2010	GBM-50	RF-100	XGB-500
NSL-KDD	GBM-500	RF-450	XGB-500
UNSW-NB15	GBM-500	RF-500	XGB-500

but their modeling details are different. More precisely, XGB uses a more regularized model formalization to control over-fitting, which gives it better performance [52].

All classifiers forming the stacked ensemble are briefly discussed as followings.

- **Random Forest (RF).**
  It produces a certain number of trees, where a random selection algorithm is used to pick the variables to put into each model [53]. It takes the opportunity of bagging [26] and random features selection for tree construction. The tree is produced fully with no pruning, thus each tree has a lower bias and the correlation of individual tree has low variance. The strategy of merging the generated trees yields a satisfactory prediction accuracy while reducing the over-fitting. There exist several trimmable learning criterions in RF, e.g. variable selection in each node, which is typically kept unchanging for all nodes, tree size, that make up the forest. As opposed to other ensemble learners, RF possesses some merits such as a less computation effort as every single tree is built on a small number of variables and simpler implementation

TABLE 5. Performance results of all classifiers w.r.t accuracy metric (%) along with Friedman rank (best value is indicated in bold).

	CICIDS-2017	CSIC-2010	NSL-KDD	UNSW-NB15	Average	Friedman average rank	Quade test p-value
GBM	**99.9941**	98.4926	91.0442	93.4562	95.7468	2.125	0.09203
RF	99.9234	89.2557	91.1373	93.3581	93.4186	3.25	
XGB	**99.9941**	92.7879	90.0062	92.4530	93.8103	3.125	
Proposed	99.9882	**98.5217**	**92.1664**	**93.6256**	**96.0755**	**1.5**	

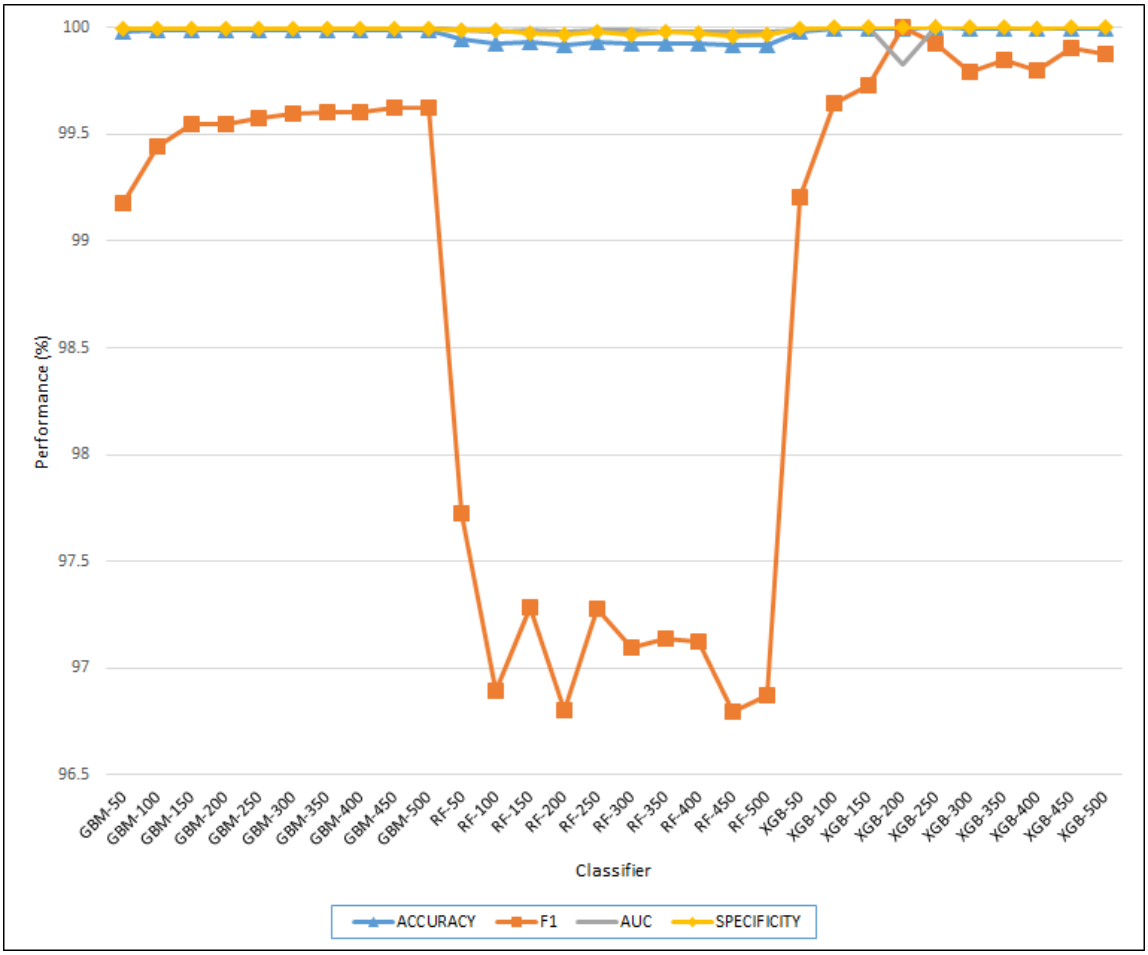

FIGURE 2. Performance average of each base model with different size of trees on CICIDS-2017 dataset.

in a parallel computing approach that can further speed up the algorithm.

- **Gradient Boosting Machine**.
Gradient boosting machine (GBM) [54] is constructed based on classification and regression trees (CART) [55], where classification and regression task are undertaken at one process when building the classification model. It is a part of monolithic ensemble family, where a number of undiversified base classifiers, e.g. weak prediction models are blended to produce the classification model. Given a dataset with m samples and s variables $D = (x_i, y_i)(|D| = m, x_i \in \Re^s, y_i \in \Re)$, a tree ensemble employs L additive function to predict the final output [54].

$$\hat{y}_i = \phi(x_i) = \sum_{l=1}^{L} f_l(x_i), \quad f_l \in F \tag{1}$$

where the space of CART (classification and regression trees) is defined as: $F = f(x) = w_{p(x)}(p : \Re^s \rightarrow T, w \in \Re^T)$. The p represents the configuration of each tree that maps a sample to an appropriate leaf index. T represents the tree size, while f_k is a stand-alone tree configuration p and leaf weight w. The decision guidelines in the trees (p) is utilized to predict a given sample into the leaves and compute the outcome through the total score in the

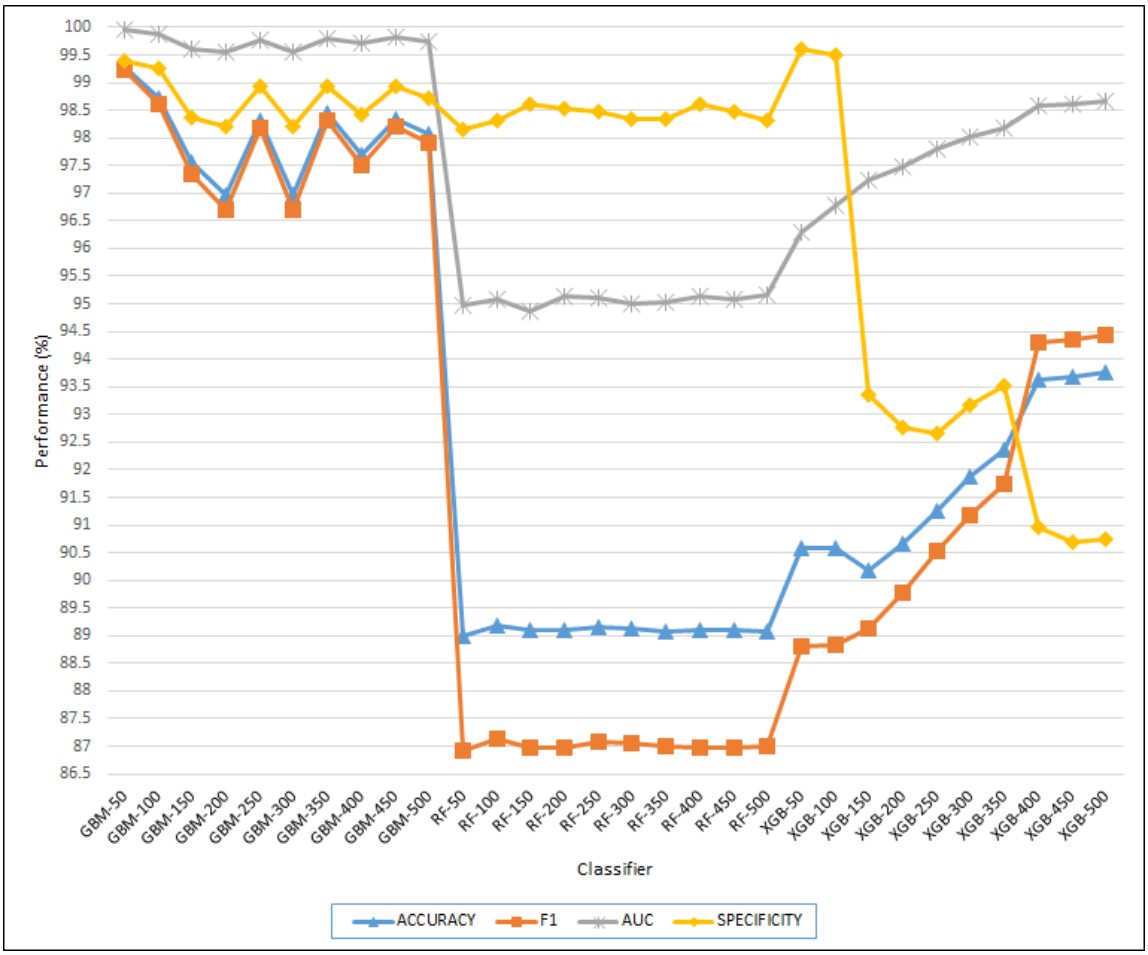

FIGURE 3. Performance average of each base model with different size of trees on CSIC-2010v2 dataset.

corresponding leaves (w). Obtaining the best split is one of the drawbacks in the tree learning. To figure out this, we take into account an exact greedy algorithm in H_2O.

- **XGBoost**.
  Both GBM and XGBoost adheres to the principle of gradient boosting [56]. Gradient boosting is an optimization algorithm that can find optimal solutions to a large variety of problems. The fundamental concept of the algorithm is to fine-tune learning parameters repetitively to lower a cost function. XGboost has many advantages compared to GBM in terms of speed and memory utilization such as a better processor cache utilization and support multicore processing. Furthermore, XGBoost employs a more regularized model for regression tree structure, thus providing better performance and reducing the model complexity to avoid overfitting [57].
- **Generalized Linear Model (GLM)**
  Generalized linear models works by estimating the relationship between outcome and features with an exponential distribution. Depending on distribution and function, GLM can be employed either for regression and classification [58]. In this paper, since we deal with an anomaly-based IDS, a logistic regression is used for binary classification problem where the outcome is a categorical variable with two levels, i.e. attack and normal. Logistic regression models the probability of a sample belonging x to a outcome category y. The fitted model $\hat{y}$ can be written as follows.

$$\hat{y} = Pr(y = 1|x) = \frac{e^{x^T\beta+\beta_0}}{1 + e^{x^T\beta+\beta_0}} \tag{2}$$

This paper considers a stacked architecture of ensemble, which is made up of three base learners (level-0 classifiers), i.e. RF, GBM, and XGBoost and a combiner (level-1 classifier), e.g. GLM. First and foremost, the base learners are trained using training data, then GLM is trained to make a final prediction based on the prediction mixtures of the base learners. Unlike other conventional ensembles that blend single weak classifiers, i.e. decision tree, neural network, or support vector machine, the aim of our stacked ensemble is to construct a diverse group of strong base learners.

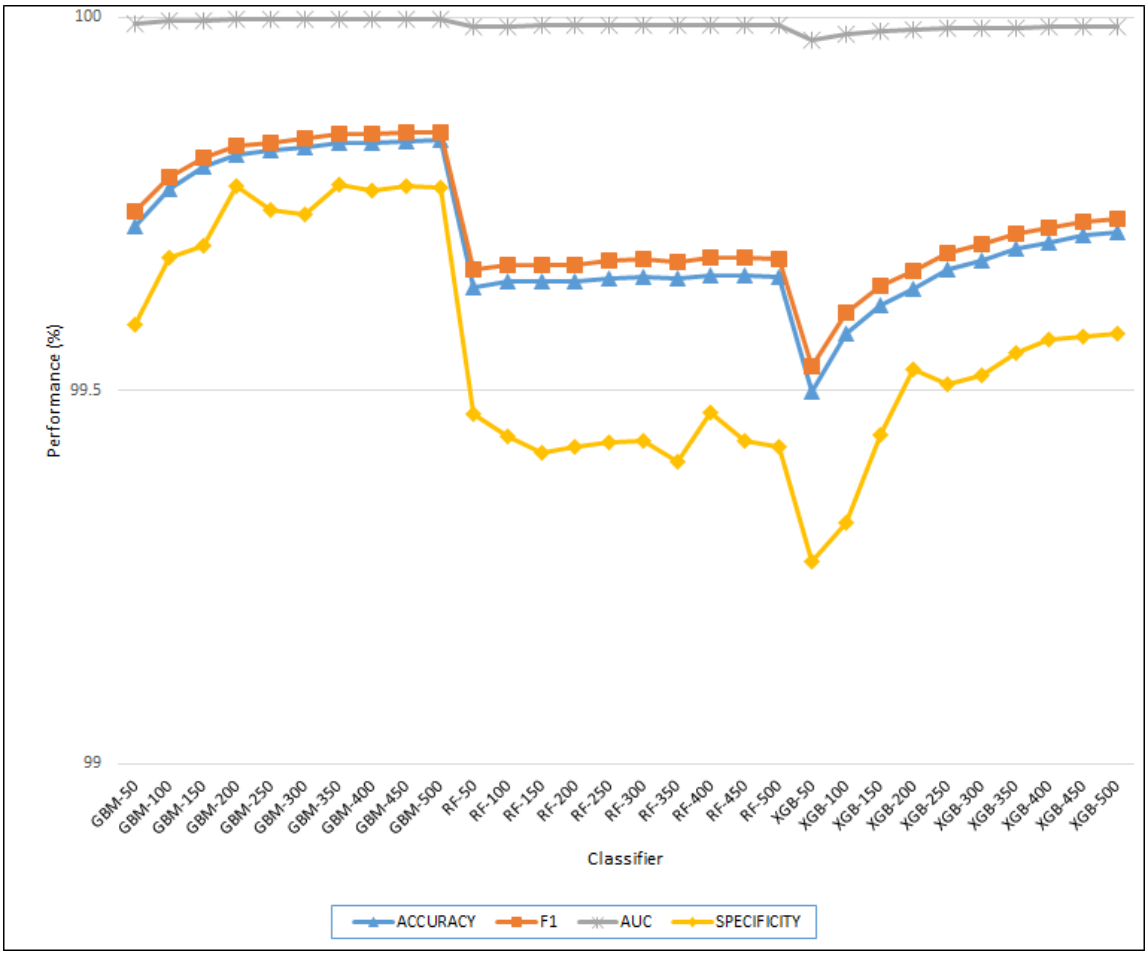

FIGURE 4. Performance average of each base model with different size of trees on NSL-KDD dataset.

3) PROCEDURE TO COMBINE BASE LEARNERS

The proposed stacked ensemble consists of the following steps:

1) Set up the stacked ensemble.
   - Supposed we have a training data, e.g. level-0 data with m instances and s variables. The level-0 data can be represented as an input matrix, X, with response matrix, y.

$$m\left\{\overbrace{\left[\quad X \quad\right]}^{s}\left[y\right]\right. \tag{3}$$

   - Specify a list of E base learners (along with their optimal hyperparameter values and tree-sizes).
     Each base learner undergoes parameter tuning using *grid* search, where an area under the ROC curve (AUC) metric [59] is specified as a stopping criterion of the best possible parameter settings. In addition, different numbers of trees are also searched for each base learner before it is ready for being used in the classification task. By trying all possible tree-sizes within the [50,500] range, the performance of each base learner is then evaluated with respect to F_1 metric. It is worth mentioning that there is no generalized evaluation metric for various kinds of classification problem [60]. However, we take into account AUC and F_1 since both metrics are suitable to evaluate the learners for binary classification with imbalanced datasets.
   - Specify the level-1 classifier.
     Generalized linear model is specified for the combiner, e.g. level-1 classifier. GLM is recommended for level-1 classifier, while several other classifiers are demonstrated not to be suitable [61], [62].

2) Train the stacked ensemble.
   - Train each of the E base learner on the training set.
   - Do 10-fold cross validation on each base learner and gather the prediction results, $cv_1, cv_2, \ldots, cv_E$. Here, the same k-type cross validation should be used. In this case, we use a stratified cross validation which stratifies the folds based on the class outcome value [55]. The stratification can improve the performance of the cross validation, providing

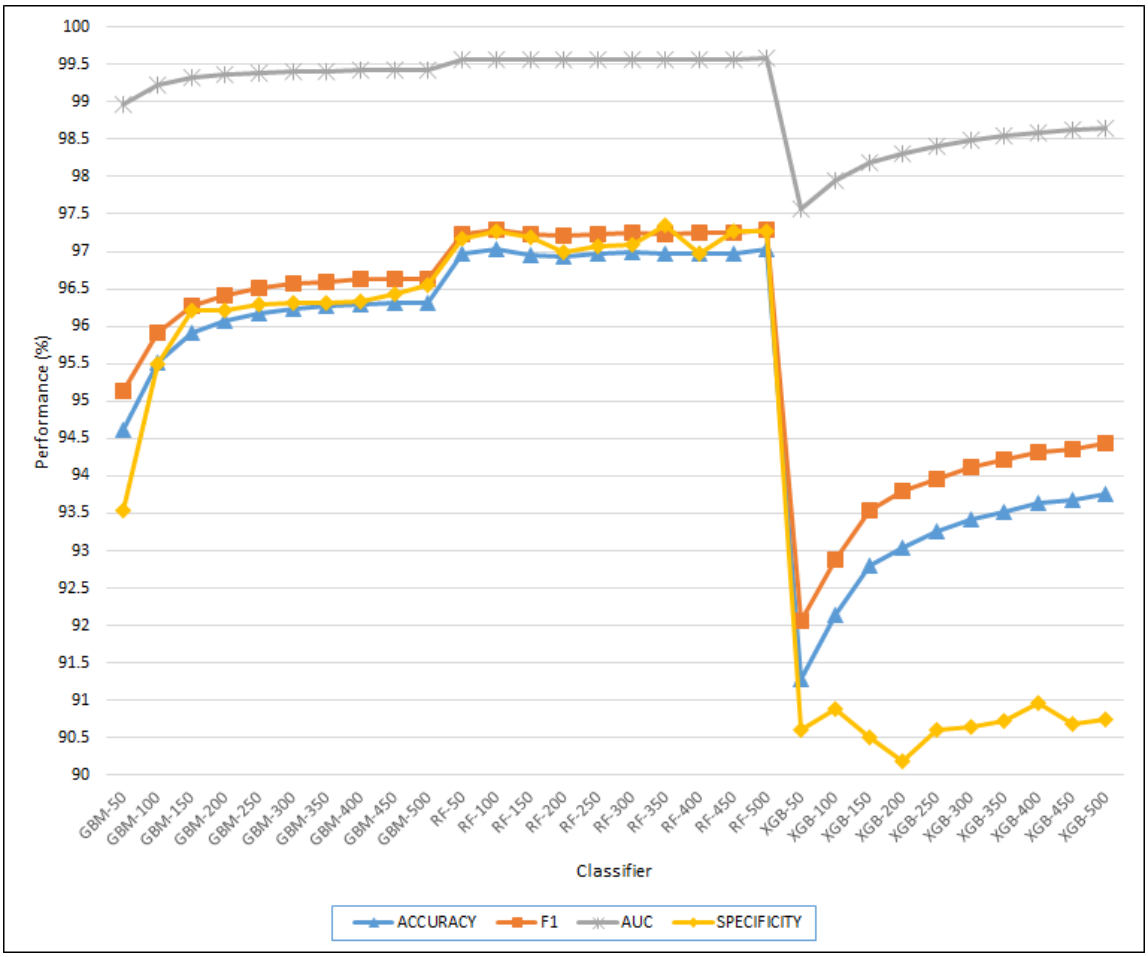

FIGURE 5. Performance average of each base model with different size of trees on UNSW-NB15 dataset.

lower biases and small variances in estimated accuracy [63].

- The M prediction result values from each of the E base learners is blended in such a way that a feature matrix $M \times E$ (denoted as W in Equation 4) is formed. Together with original response vector y, train the the combiner on the level-1 data, $y = f(W)$.

$$m\left\{\left[cv_1\right]\ldots\left[cv_E\right]\left[y\right]\right\} \rightarrow m\left\{\overbrace{\left[\quad W \quad\right]}^{E}\left[y\right]\right\} \tag{4}$$

- Train the combiner on the level-1 data. The stacked ensemble model consisting of the E base learner models and the combiner model can now be used to generate predictions on new testing set.

3) Predict on new testing set.

- Obtain the predictions from the base learners and feed into the combiner to get the final stacked ensemble prediction.

IV. EXPERIMENTAL RESULT AND DISCUSSION

A. BENCHMARK ANALYSIS

This section provides and discusses the whole experimental results of anomaly detection in Web-traffic. We take into consideration 30 classifiers and 4 intrusion datasets, leading to a total of 120 experiment combinations. Furthermore, we conduct a hyper-parameter search for each classifier and dataset, where the best-obtained learning parameters are shown in Table 2-3. The test results are the mean value of 10 items provided by 10-fold cross-validation. Figure 2-5 denote the performance average values of the 30 classifiers concerning each particular intrusion dataset. These graphs visualize the characteristic of each classifier with different tree sizes. It can be observed that a tree size variation in RF does not have an impact on the performance. More specifically, RF is a stable classifier irrespective of the number of trees to build the ensemble. In contrast to this, the performance of GBM and XGB vary considerably as the number of trees increase.

To construct a stack of classifier ensembles, the best model for each dataset is chosen based on the maximum accuracy of

TABLE 6. Performance results of all classifiers w.r.t F_1 metric (%) along with Friedman rank (best value is indicated in bold).

	CICIDS-2017	CSIC-2010	NSL-KDD	UNSW-NB15	Average	Friedman average rank	Quade test p-value
GBM	**99.7685**	98.3654	89.3597	95.3062	95.7000	1.875	0.1152
RF	96.9697	87.1370	89.3429	95.2604	92.1775	3.5	
XGB	**99.7685**	92.0085	88.7199	94.4423	93.7348	3.125	
Proposed	99.5370	**98.3991**	**90.3518**	**95.4468**	**95.9337**	**1.5**	

TABLE 7. Performance results of all classifiers w.r.t AUC metric (%) along with Friedman rank (best value is indicated in bold).

	CICIDS-2017	CSIC-2010	NSL-KDD	UNSW-NB15	Average	Friedman average rank	Quade test p-value
GBM	99.9996	99.7837	96.4548	98.7051	98.7358	2.5	0.03682
RF	99.9773	95.0939	96.7150	98.6701	97.6141	3.25	
XGB	99.9998	98.2658	95.6943	98.1275	98.0219	3.25	
Proposed	**99.9999**	**99.7875**	**97.1037**	**98.7302**	**98.9053**	**1.0**	

TABLE 8. Performance results of all classifiers w.r.t specificity metric (%) along with Friedman rank (best value is indicated in bold).

	CICIDS-2017	CSIC-2010	NSL-KDD	UNSW-NB15	Average	Friedman average rank	Quade test p-value
GBM	**100**	**100**	99.1662	**100**	99.7916	2.75	0.8395
RF	**100**	**100**	**100**	99.9946	99.9987	2.625	
XGB	**100**	**100**	99.9922	**100**	99.9981	2.5	
Proposed	**100**	**100**	**100**	**100**	**100**	**2.125**	

F_1 metric. Table 4 presents the best base models used to build the proposed classifier. As a result, different architectural models are available, tailoring with the specified intrusion dataset. For instance, GBM-450, RF-50, and XGB-200 are employed as the base models of the proposed classifier when dealing with CICIDS-2017, and so forth. Table 5-8 compare the performance results of the base classifiers and the proposed classifier concerning the accuracy, F_1, AUC, and specificity metric, respectively. On average, the proposed classifier outperforms the base classifiers in terms of all performance metrics.

Concerning a fair comparative analysis, we are interested in applying several statistical significance tests. The tests measure how significant are the performance differences among the considered classifiers. For this purpose, the non-parametric Friedman rank test [64], Quade omnibus test, and Quade posthoc test [65] are used in our benchmark study. Quade test is deemed to be powerful in case the number of classifiers is less than five [66]. The procedure for performing statistical significance test can be broken down as follows.

- Using Friedman rank, calculate the rank for each dataset independently, according to the performance metrics, in ascending order, from the best-performing algorithm to the worst-performing algorithm.
- Take the mean rank of the classifier over all datasets. The best-performing classifier is judged by the lowest value of Friedman rank. The ranking is a metric where merit is inversely proportional to numeric values.

TABLE 9. Results of a Quade post-hoc test w.r.t AUC metric (the proposed classifier is considered as a control classifier).

Comparison	Quade post-hoc p-value
GBM v.s. Proposed	0.2530
RF v.s. Proposed	0.0725
XGB v.s. Proposed	0.0500

TABLE 10. A representation of confusion matrix for anomaly-based intrusion detection.

		Actual	
		Anomaly	Normal
Predicted	Anomaly	True Negative (TN)	False Positive (FP)
	Normal	False Negative (FN)	True Positive (TP)

- Perform an omnibus test using Quade test which checks whether at least one classifier has performed differently than others. If the test indicates significant, e.g. p-value is less than a threshold (0.05 in our case), a pair-wise Quade posthoc test is then performed.
- Do a Quade posthoc test for multiple comparison. We are given the option of the pairwise comparison, all-pairwise, or comparison with control. In this paper, the best-performing classifier, e.g. the proposed classifier, is picked as a control algorithm for being compared with the remaining classifiers.

Table 5-8 also show the mean rank of classifiers over the whole intrusion datasets, as well as the results of the Quade omnibus test. The proposed classifier appears as the

TABLE 11. Prediction results of proposed model on testing set (20%) of CSIC2010v2.

		Actual	
		Anomaly	Normal
Predicted	Anomaly	23795	178
	Normal	350	20324

TABLE 12. Prediction results of proposed model on testing set (20%) of CICIDS-2017.

		Actual	
		Benign	Malicious
Predicted	Benign	33526	0
	Malicious	2	431

TABLE 13. Prediction results of proposed model on KDDTest+.

		Actual	
		Anomaly	Normal
Predicted	Anomaly	12509	324
	Normal	1442	8269

TABLE 14. Prediction results of the proposed model on UNSW-NB15$_{test}$.

		Actual	
		Anomaly	Normal
Predicted	Anomaly	49674	6326
	Normal	6907	112434

champion of the benchmark, considering that it has the lowest average rank in all performance metrics. Bear in mind that the lower the rank, the better performance of the classifier is. Surprisingly, the results of the Quade test is significant ($p < 0.05$) in terms of the AUC metric, whilst other metrics are not significant ($p > 0.05$). This means that the null hypothesis can be rejected in case the significance is found. As a result, Quade posthoc test is performed based on AUC values. Table 9 shows that the proposed classifier is superior compared to XGB (p-value $= 0.005$), however, there is no significance between the proposed classifier and GBM (p-value $= 0.2530$). Moreover, the performance difference between the proposed classifier and RF is not too significant (p-value $= 0.0725$).

B. COMPARISON WITH THE BASELINES

We broaden our comparative analysis by elaborating on the performance results of several existing detection models. Prediction results of the proposed model are summarized with a confusion matrix, denoting the number of correct and incorrect predictions which are decomposed by each class. A two-class confusion matrix is illustrated in Table10. The table allows us to get the accuracy and false positive rate (FPR) as followings.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

Prediction results of the proposed model are presented in Table 11-14. The proposed model is considerably robust in handling data imbalanced problem such as CICIDS-2017. As a matter of fact, the CICIDS-2017 is highly imbalance with a ratio of benign to malicious on the testing set is 33,526 to 433. All benign samples are correctly classified as benign, whilst 2 malicious samples are incorrectly classified as benign. Based on the results given in those tables, two performance metrics, i.e. accuracy and FPR can be obtained for further benchmark with some existing works. Table15-18 provide an unbiased comparison with the existing techniques grouped by each intrusion dataset.

Table15 shows the performance comparison in terms of accuracy and FPR metric over the CSIC2010v2 dataset.

TABLE 15. Performance benchmark with other existing methods on CSIC 2010v2 (best value is indicated in bold).

Technique	Feature selection	Validation method	Accuracy (%)	FPR(%)	Significance test
Proposed	No	Hold-out 80/20	**98.82**	**0.74**	Two step statistical test
OC-WAD [36]	No	10cv	95.90	2.82	Paired t-test
HTTP Header Analysis [35]	No	10cv	94.46	4.34	Not reported
J48 [34]	No	10cv	95.97	3.54	Not reported
A-IDS [33]	No	10cv	90.52	-	Not reported
C4.5 [31]	GeFS	10cv	94.49	5.9	Not reported

TABLE 16. Performance benchmark with some existing methods on CICIDS-2017 (best value is indicated in bold).

Technique	Feature selection	Validation method	Accuracy (%)	FPR(%)	Significance test
Proposed	No	Hold-out 80/20	**99.99**	**0.46**	Two step statistical test
Deep neural network [43]	No	Hold-out	99.92	0.05	No
Ensemble method [40]	Yes	10cv	99.88	0.002	No
k-NN [45]	Yes	Hold out 70/30	99.46	Not reported	No
Random forest [24]	Yes	Not reported	99.40	0.01	No
Ensemble classifier [67]	Yes	Hold-out	96.8	0.032	No
Deep neural network [42]	Yes	Hold-out	96.30	Not reported	No
GRU-RNN [41]	Yes	Hold-out	89.00	Not reported	No
Adaboost [68]	Yes	5×10cv	81.83	Not reported	No
Local outlier factor [46]	No	Hold-out	68.0	Not reported	No

TABLE 17. Comparative evaluation with several baseline methods on independent testing set, e.g. KDDTest+ (best value is indicated in bold).

Method	Year	Feature selection	Accuracy (%)	FPR (%)	Statistical test
Proposed	2019	No	**92.17**	**2.52**	Yes
GBM [28]	2019	No	91.82	4.19	Yes
Two-stage ensemble [27]	2019	Hybrid	85.797	11.7	Yes
SVM [69]	2019	MBGWO	81.58	Not reported	No
Bagging (J48) [70]	2018	Gain ratio	84.25	2.79	No
Two-tier classifier [71]	2017	LDA	83.240	4.8	No
TDTC [72]	2016	Two-layer	84.82	5.56	No
GAR-Forest [73]	2016	No	82.399	14.3	No
GAR-Forest [73]	2016	InfoGain	83.641	13.3	No
GAR-Forest [73]	2016	CFS	82.976	14.9	No
GAR-Forest [73]	2016	SU	85.056	12.2	No
SVM [74]	2014	Yes	82.37	15	No

TABLE 18. Comparative evaluation with several baseline methods on independent testing set, e.g. UNSW-NB15$_{test}$ (best value is indicated in bold).

Method	Year	Feature selection	Accuracy (%)	FPR (%)	Statistical test
Proposed	2019	No	**92.45**	**11.3**	Yes
GBM [28]	2019	No	91.31	8.60	Yes
Two-stage ensemble [27]	2019	Hybrid	91.27	8.90	Yes
Two-stage classifier [75]	2018	Information gain	85.78	15.64	No
Decision tree [76]	2017	GA-LR	81.42	6.39	No
Decision tree [77]	2016	No	85.56	15.78	No
Logistic regression [77]	2016	No	83.15	18.48	No
Naive Bayes [77]	2016	No	82.07	18.56	No
Neural network [77]	2016	No	81.34	21.13	No
Expectation-maximization [77]	2016	No	78.47	23.79	No

Compared to OC-WAD, the proposed model improves the detection accuracy by 2.92%, whilst significantly reducing the FPR rate to 0.74. This outperforms other Web traffic detection techniques, i.e. HTTP Header Analysis, J48, and other individual classification algorithms. The proposed model shows a promising result when it is applied on CICIDS-2017, yielding a near-perfect detection accuracy, e.g. 99.99% and a perfect false prediction rate, e.g. 0.46%. The result significantly surpasses the latest IDS approach, i.e. deep neural network [43] and ensemble method [40]. The accuracy is the highest result this far using full feature set since our stacked ensemble is built based upon optimal learning parameters of its base classifiers. The ability to design this improved framework has led to achieving higher accuracy compared to state of the art techniques.

Subsequently, it is meaningful to compare the proposed model on a testing set, i.e. KDDTest+ which is designed as independent testing samples of the NSL-KDD dataset. The result would be necessarily important to see how well the proposed model performs in detecting unseen attacks. Table17 confirms the superiority of our proposed model tested on KDDTest+. The proposed model has performed better than state-of-the-art IDS techniques, i.e. two-stage classifier ensemble [27], gradient boosting machine [28], and other techniques. The detection accuracy could be enhanced by 0.35%, while maintaining a meaningful FPR rate, e.g. 2.52%, in comparison with the best existing detection techniques. Similarly, the performance result of the proposed classifier has demonstrated a considerable improvement over the existing works when dealing with the UNSW-NB15$_{test}$ data samples (see Table18). By employing our detection model, the detection accuracy is boosted remarkably at 1.14%, whereas the FPR rate can be achieved at 11.3%. This result is very competitive in comparison with a two-stage classifier [75] and decision tree [77]. To sum up, our IDS classification technique exhibits a promising solution, regardless of the use of intrusion datasets.

V. CONCLUSION

This study has explored the use of stack architecture to combine multiple classifier ensembles, i.e. gradient boosting machine (GBM), random forest (RF), and extreme gradient boosting machine (XGB) for detecting anomaly in a Web application scenario. To prove the generalizability of our proposed model, we have tested on multiple IDS datasets such as CSIC-2010v2, CICIDS-2017, NSL-KDD, and UNSW-NB15. Unlike a conventional stacking technique that usually considers a weak individual classification algorithm, our proposed model is built based on a combination of strong classifier ensembles that work as base learners. To build such strong base learners, each learner undergoes fine-tuned hyperparameter search and optimal tree-size. Our proposed approach yields an almost-perfect detection performance, concerning the accuracy and false positive rate (FPR) measure. This study possesses several limitations such as a limited number of datasets used, as well as incomplete discussion about multi-class classification. Among many possible ways to improve this paper, future work might include more intrusion datasets and take into consideration multi-class classification.

It would also be interesting to consider deep neural networks as base models.

REFERENCES

[1] C. Verdouw, J. Wolfert, A. Beulens, and A. Rialland, "Virtualization of food supply chains with the Internet of Things," *J. Food Eng.*, vol. 176, pp. 128–136, May 2016.

[2] B. Durmuş, Y. Ulusu, and Ş. Erdem, "Which dimensions affect private shopping E-customer loyalty?" *Procedia Social Behav. Sci.*, vol. 99, pp. 420–427, Nov. 2013.

[3] V. Shankar, A. Venkatesh, C. Hofacker, and P. Naik, "Mobile marketing in the retailing environment: Current insights and future research avenues," *J. Interact. Marketing*, vol. 24, no. 2, pp. 111–120, May 2010.

[4] E. Pantano, "Ubiquitous retailing innovative scenario: From the fixed point of sale to the flexible ubiquitous store," *J. Technol. Manage. Innov.*, vol. 8, no. 2, pp. 13–14, Aug. 2013.

[5] E. Pantano and C.-V. Priporas, "The effect of mobile retailing on consumers' purchasing experiences: A dynamic perspective," *Comput. Hum. Behav.*, vol. 61, pp. 548–555, Aug. 2016.

[6] M. Friedewald and O. Raabe, "Ubiquitous computing: An overview of technology impacts," *Telematics Informat.*, vol. 28, no. 2, pp. 55–65, May 2011.

[7] J. Lee, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[9] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing-the business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, 2011.

[10] B. A. Tama and K.-H. Rhee, "HFSTE: Hybrid feature selections and tree-based classifiers ensemble for intrusion detection system," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 8, pp. 1729–1737, 2017.

[11] T. G. Booth and K. Andersson, "Elimination of DoS UDP reflection amplification bandwidth attacks, protecting TCP services," in *Proc. Int. Conf. Future Netw. Syst. Secur.* Cham, Switzerland: Springer, 2015, pp. 1–15.

[12] L. B. A. Rabai, M. Jouini, A. B. Aissa, and A. Mili, "A cybersecurity model in cloud computing environments," *J. King Saud. Univ.-Comput. Inf. Sci.*, vol. 25, no. 1, pp. 63–75, 2013.

[13] I. A. Sumra, H. B. Hasbullah, and J.-L. B. AbManan, "Attacks on security goals (confidentiality, integrity, availability) in VANET: A survey," in *Vehicular Ad-Hoc Networks for Smart Cities*. Singapore: Springer, 2015, pp. 51–61.

[14] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for SCADA systems," *Comput. Secur.*, vol. 56, pp. 1–27, Feb. 2016.

[15] W. G. Halfond, J. Viegas, and A. Orso, "A classification of SQL-injection attacks and countermeasures," in *Proc. IEEE Int. Symp. Secure Softw. Eng.*, vol. 1, Mar. 2006, pp. 13–15.

[16] R. Johari and P. Sharma, "A survey on Web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection," in *Proc. Int. Conf. Commun. Syst. Netw. Technol.*, May 2012, pp. 453–458.

[17] P. Kumar and R. K. Pateriya, "A survey on SQL injection attacks, detection and prevention techniques," in *Proc. 3rd Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2012, pp. 1–5.

[18] D. A. Kindy and A.-S.-K. Pathan, "A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques," in *Proc. IEEE 15th Int. Symp. Consum. Electron. (ISCE)*, Jun. 2011, pp. 468–471.

[19] T. O. W. A. S. Project. (2013). *OWASP Top Ten 2013 Project*. [Online]. Available: https://www.owasp.org/index.php/Top_10_2013-Top_10

[20] S. Gupta and B. B. Gupta, "Cross-site scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art," *Int. J. Syst. Assurance Eng. Manag.*, vol. 8, no. S1, pp. 512–530, Jan. 2017.

[21] T. K. Saha and A. B. M. S. Ali, "Web application security attacks and countermeasures," in *Case Studies in Secure Computing: Achievements and Trends*. Boca Raton, FL, USA: CRC Press, 2015, pp. 343–371.

[22] 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.

[23] C. Torrano-Gimenez, A. Perez-Villegas, and G. A. Marañón, "An anomaly-based approach for intrusion detection in Web traffic," *J. Inf. Assurance Secur.*, vol. 5, no. 4, pp. 446–454, 2010.

[24] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[26] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[27] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[28] B. A. Tama and K.-H. Rhee, "An in-depth experimental study of anomaly detection using gradient boosted machine," *Neural Comput. Appl.*, vol. 31, no. 4, pp. 955–965, Apr. 2019.

[29] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[30] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *J. Netw. Comput. Appl.*, vol. 28, no. 2, pp. 167–182, Apr. 2005.

[31] H. T. Nguyen, C. Torrano-Gimenez, G. Alvarez, S. Petrović, and K. Franke, "Application of the generic feature selection measure in detection of Web attacks," in *Computational Intelligence in Security for Information Systems*. Berlin, Germany: Springer, 2011, pp. 25–32.

[32] H. T. Nguyen, K. Franke, and S. Petrović, "Reliability in a feature-selection process for intrusion detection," in *Reliable Knowledge Discovery*. Boston, MA, USA: Springer, 2012, pp. 203–218.

[33] H. T. Nguyen and K. Franke, "Adaptive intrusion detection system via online machine learning," in *Proc. 12th Int. Conf. Hybrid Intell. Syst. (HIS)*, Dec. 2012, pp. 271–277.

[34] R. Kozik, M. Choraé, R. Renk, and W. Hołubowicz, "A proposal of algorithm for Web applications cyber attack detection," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.* Berlin, Germany: Springer, 2014, pp. 680–687.

[35] R. Kozik, M. Choraé, R. Renk, and W. Hołubowicz, "Modelling HTTP requests with regular expressions for detection of cyber attacks targeted at Web applications," in *Proc. Int. Joint Conf.* Cham, Switzerland: Springer, 2014, pp. 527–535

[36] E. Parhizkar and M. Abadi, "OC-WAD: A one-class classifier ensemble approach for anomaly detection in Web traffic," in *Proc. 23rd Iranian Conf. Electr. Eng.*, May 2015, pp. 631–636.

[37] N. Epp, R. Funk, C. Cappo, and S. Lorenzo-Paraguay, "Anomaly-based Web application firewall using HTTP-specific features and one-class svm," in *Proc. Workshop Regional Segurança Informação Sistemas Computacionais*, 2017, pp. 1–6.

[38] Y. Yao, L. Su, C. Zhang, Z. Lu, and B. Liu, "Marrying graph kernel with deep neural network: A case study for network anomaly detection," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2019, pp. 102–115.

[39] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time distributed-random-forest-based network intrusion detection system using apache spark," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Nov. 2018, pp. 1–7.

[40] A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach," *IEEE Access*, vol. 7, pp. 106495–106513, 2019.

[41] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in sdn-based networks: Deep recurrent neural network approach," in *Deep Learning Applications for Cyber Security*. Cham, Switzerland: Springer, 2019, pp. 175–195.

[42] R. Vinayakumar, M. Alazab, K. P. 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.

[43] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "Intelligent approach to build a Deep neural network based IDS for cloud environment using combination of machine learning algorithms," *Comput. Secur.*, vol. 86, pp. 291–317, Sep. 2019.

[44] S. Park and H. Park, "Ann based intrusion detection model," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2019, pp. 433–437.

[45] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *Proc. Int. Conf. Secur., Privacy Anonymity Comput., Commun. Storage*. Cham, Switzerland: Springer, 2019, pp. 277–288.
[46] D. Pérez, S. Alonso, A. Morán, M. A. Prada, J. J. Fuertes, and M. Domínguez, "Comparison of network intrusion detection performance using feature representation," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 463–475.
[47] E. DePoy and L. N. Gitlin, "Statistical analysis for experimental-type designs," in *Introduction to Research*, 5th ed, E. DePoy and L. N. Gitlin, Eds. Maryland Heights, MO, USA: Mosby, 2016, pp. 282–310.
[48] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Military Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
[49] S. Aiello, E. Eckstrand, A. Fu, M. Landry, and P. Aboyoun, "Machine learning with R and H_2O," 6th Ed. H2O.ai, Mountain View, CA, USA, Tech. Rep., 2016.
[50] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.
[51] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 1–9.
[52] T. Chen, "Introduction to boosted trees," *Univ. Washington Comput. Sci.*, vol. 22, p. 115, Oct. 2014.
[53] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[54] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," in *Proc. Ann. Statist.*, 2001, pp. 1189–1232.
[55] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
[56] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 785–794.
[57] K. V. Rashmi and R. Gilad-Bachrach, "DART: Dropouts meet multiple additive regression trees," in *Proc. AISTATS*, 2015, pp. 489–497.
[58] N. E. Breslow, "Generalized linear models: Checking assumptions and strengthening conclusions," *Statistica Applicata*, vol. 8, no. 1, pp. 23–41, 1996.
[59] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[60] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," in *Proc. Int. Symp. Intell. Comput. Appl.* Berlin, Germany: Springer, 2009, pp. 461–471.
[61] S. Džeroski and B. Ženko,, "Is combining classifiers with stacking better than selecting the best one?" *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, Mar. 2004.
[62] K. M. Ting and I. H. Witten, "Issues in Stacked Generalization," *Jair*, vol. 10, pp. 271–289, Jul. 2018.
[63] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. (AI)*, Aug. 1995, vol. 14. no. 2, pp. 1137–1145.
[64] M. Friedman, "A comparison of alternative tests of significance for the problem of *m* rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, Mar. 1940.
[65] D. Quade, "Using weighted rankings in the analysis of complete blocks with additive block effects," *J. Amer. Stat. Assoc.*, vol. 74, no. 367, pp. 680–683, Sep. 1979.
[66] W. Conover, *Practical Nonparametric Statistics*. Hoboken, NJ, USA: Wiley, 1999.
[67] Y.-Y. Zhou and G. Cheng, "An efficient intrusion detection system based on feature selection and ensemble classifier," Apr. 2019, *arXiv:1904.01352*. [Online]. Available: https://arxiv.org/abs/1904.01352
[68] 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, Mar. 2019, Art. no. 012018.
[69] Q. M. Alzubi, M. Anbar, Z. N. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Comput. Appl.*, pp. 1–13, Apr. 2019.
[70] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)*, 2018, p. 2.
[71] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017.
[72] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K.-R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 2, pp. 314–323, Apr. 2019.
[73] N. K. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using GAR-Forest with feature selection," in *Proc. 4th Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*. New Delhi, India: Springer, 2016, pp. 539–547.
[74] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2014., pp. 1–6.
[75] W. Zong, Y.-W. Chow, and W. Susilo, "A two-stage classifier approach for network intrusion detection," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.* Cham, Switzerland: Springer, 2018, pp. 329–340.
[76] 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.
[77] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., A Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

BAYU ADHI TAMA received the Ph.D. degree from Pukyong National University, South Korea. He was a Postdoctoral Researcher at the School of Management Engineering, Ulsan National Institute of Science and Technology (UNIST), South Korea. He is currently a Postdoctoral Researcher at the Industrial Artificial Intelligence Laboratory, Pohang University of Science and Technology (POSTECH), South Korea. He has published more than 30 papers in international journals and conferences. His research interests include machine learning and artificial intelligence techniques applied for cyber security, medical informatics, and industrial applications.

LEWIS NKENYEREYE received the Ph.D. degree in information security from Pukyong National University, Busan, South Korea. He was a Visiting Scholar at Thompson Rivers University, BC, Canada, and Georgia Southern University, Statesboro, GA, USA. He is currently an Assistant Professor with the Department of Computer and Information Security, Sejong University, Seoul, South Korea. Before joining Sejong University, he was a Research Fellow at Creative Human Resource Development Program for IT Convergence, Pusan National University. His research spans across the wide range of security and privacy related techniques with a particular interest in the Internet of Things (specifically Internet of Vehicles). He is involved in privacy preserving techniques projects for BlockChain-based applications; interoperability challenges in IoT and M2M standards (with a special focus on oneM2M). He is actively involved in IoT/M2M ventures, data interoperability initiatives, and security/privacy related projects.

S. M. RIAZUL ISLAM (Member, IEEE) received the B.S. and M.S. degrees in applied physics and electronics from the University of Dhaka, Bangladesh, in 2003 and 2005, respectively, and the Ph.D. degree in information and communication engineering from Inha University, South Korea, in 2012. From 2014 to 2017, he worked at Inha University, South Korea, as a Postdoctoral Fellow at the Wireless Communications Research Center. He has been working at the Department of Computer Science and Engineering, Sejong University, South Korea, as an Assistant Professor, since March 2017. He was with the University of Dhaka, Bangladesh, as an Assistant Professor and Lecturer at the Department of Electrical and Electronic Engineering, from September 2005 to March 2014. In 2014, he worked at the Samsung R&D Institute Bangladesh (SRBD) as a Chief Engineer at the Department of Solution Lab for six months. His research interests include wireless communications, 5G & IoT, wireless health, bioinformatics, and machine learning.

KYUNG-SUP KWAK (Member, IEEE) received the M.S. and Ph.D. degrees from the University of California at San Diego, in 1981 and 1988, respectively. From 1988 to 1989, he was with Hughes Network Systems, San Diego, CA, USA. From 1989 to 1990, he was with the IBM Network Analysis Center, Research Triangle Park, NC, USA. Since then, he has been with the School of Information and Communication Engineering, Inha University, South Korea, as a Professor. He was the Dean of the Graduate School of Information Technology and Telecommunications, Inha University, from 2001 to 2002. He has been the Director of the UWB Wireless Communications Research Center, South Korea, since 2003. In 2006, he has served as the President of the Korean Institute of Communication Sciences and the Korea Institute of Intelligent Transport Systems, in 2009. His research interests include wireless communications, ultrawideband systems, sensor networks, wireless body area networks, and nanocommunications. He received a number of awards, including the Engineering College Achievement Award from Inha University, the LG Paper Award, the Motorola Paper Award, the Haedong Prize of Research, and various government awards from the Ministry of ICT, the President, and the Prime Minister of Korea, for his excellent research performances.