IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 13, 2019, accepted June 25, 2019, date of publication July 1, 2019, date of current version July 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925838

HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems

IZHAR AHMED KHAN, DECHANG PI, ZAHEER ULLAH KHAN, YASIR HUSSAIN, AND ASIF NAWAZ

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Corresponding author: Dechang Pi (nuaacs@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U1433116, and in part by the Fundamental Research Funds for the Central Universities under Grant NP2017208.

ABSTRACT Critical infrastructures, e.g., electricity generation and dispersal networks, chemical processing plants, and gas distribution, are governed and monitored by supervisory control and data acquisition systems (SCADA). Detecting intrusion is a prevalent area of study for numerous years, and several intrusion detection systems have been suggested in the literature for cyber-physical systems and industrial control system (ICS). In recent years, the viruses *seismic net*, *duqu*, and *flame* against ICS attacks have caused tremendous damage to nuclear facilities and critical infrastructure in some countries. These intensified attacks have sounded the alarm for the security of the ICS in many countries. The challenge in constructing an intrusion detection framework is to deal with unbalanced intrusion datasets, i.e. when one class is signified by a lesser amount of instances (minority class). To this end, we outline an approach to deal with this issue and propose an anomaly detection method for the ICS. Our proposed approach uses a hybrid model that takes advantage of the anticipated and consistent nature of communication patterns that occur among ground devices in ICS setups. First, we applied some preprocessing techniques to standardize and scale the data. Second, the dimensionality reduction algorithms are applied to improve the process of anomaly detection. Third, we employed an edited nearest-neighbor rule algorithm to balance the dataset. Fourth, by using the Bloom filter, a signature database is created by noting the system for a specific period lacking the occurrence of abnormalities. Finally, to detect new attacks, we combined our package contents-level detection with another instance-based learner to make a hybrid method for anomaly detection. The experimental results with a real large-scale dataset generated from a gas pipeline SCADA system show that the proposed approach HML-IDS outperforms the benchmark models with an accuracy rate of *97%*.

INDEX TERMS Bloom filters, zero-day attacks, intrusion detection, SCADA, industrial control systems.

I. INTRODUCTION

Industrial control systems (ICS) are composed of groupings of software, hardware, setups, networks, links and operators, orchestrate, and govern numerous tasks required to perform complex chores such as the distribution of useful facilities and the implementation of complex and distinct industrial procedures. The variety of ICS usage setups comprise of applications like water treatment plants [1], manufacturing industries [2], gas pipelines [3] and power plants [4]. *Traditional ICS are not networked;* therefore, are reflected as to be sound secure by apparent *air-gapped* separation. To additionally indorse greater output and proficient remote-control, adaptations of smart information and communication technologies (ICT) have been broadly merged into ICS where the utmost number of modules are long-standing, initially not secure by design and difficult to upgrade. Such progression of ICS shapes up an association between cyber worlds and the physical world, but also expose them to cyber-attacks.

The attacks that targeted ICS have been constantly growing in the past few years according to ICS-CERT [5].

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

SCADA systems are centered on computer-based process control that communicate and control physical processes that are remotely held. These systems have a tactical significance due to the element that they are embraced by nations serious infrastructure. A little harm to these kinds of critical infrastructure may have an influence on the nation's economy. Numerous real-world familiar cases and cyber-attacks that affect these systems are reported [5], which undoubtedly prove vulnerabilities of such infrastructures.

Even though this issue has been briefly considered for research in the IT security community, narrow work has been piloted to build anomaly detection systems (ADS) that are specific to ICS. Accurate anomaly detection methods have a potential to additionally simplify firm mechanism of response to any incident, and also institute the collaboration between safety and controller specialists such that impending security leaks can be identified more speedily and precisely. Hence, there is a crucial necessity of an effective system of intrusion (anomaly) detection specific to ICS.

Detecting intrusions or anomalies in ICS is a thought-provoking task due to the following causes.

- The communication protocols used by ICS (e.g. DNP3, Modbus) are not taken into account by traditional Intrusion Detection System (IDS).
- To properly train and evaluate IDS, there is a deficiency of ICS datasets that are based on real-world structures.
- Detecting anomaly in ICS cannot merely hang on information from network protocol; some more supplementary material associated to the control of physical process also required to be inspected. This considerably raises the complexity and dimensions of data samples.
- High rate of false positive for detecting anomalies and low rate of attacks detection is likely to result due to noisy behavior by physical process control variables.

IDS that are specific to ICS has been a vigorous focus for numerous years, and there present some studies in the field. Orthodox frameworks for IDS have been used such as the approaches that are based on model [6] and the approaches that are based on behaviors [7]. These studies explicitly integrated protocols of ICS in IDS (e.g. Modbus/DNP3 [8], IEC standards [9], [10]). A thorough conversation about them is presented in section II. Nevertheless, there are numerous boundaries with utmost present efforts, which are listed as follows:

- Most approaches count vastly on predefined methods to notice abnormal behaviors, demanding an enormous amount of human work at the initial phase.
- Such methods are habitually fashioned from acknowledged outbreaks and therefore are not able to detect unseen (zero-day) attacks.
- Current IDS techniques are typically personalized for explicit systems and protocols, which has a shortage of adequate generalization and elasticity to acclimatize to other systems.
- Efficient feature construction from the available data has been least considered in the studies, which, though, is of countless benefit to recognize unconventional persistent outbreaks and anomalies.

To the best of our knowledge; First, no literature considers efficient feature construction approach while developing IDS for SCADA-specific systems. Second, while developing IDS, the literature does not consider the balancing/re-sampling of the dataset. Third, the current approaches have either low detection rate or high false positive issues. Confronting these concerns, we have established an intrusion (anomaly) detection approach based on reliable feature construction and machine learning (ML) methods to support the development of a generalized ICS-specific IDS, with vastly shortened human efforts.

Towards reliable feature extraction, in this paper a feature construction approach is presented. The dimensionality reduction algorithms were used to improve accuracy. Then a multi-level intrusion detection approach has been proposed. At level one, Bloom filter based classifier is implemented and an instance-based classifier at level two is used for training of the data to predict the anomalies in the gas control network. Moreover, at present, most of the anticipated IDS lack mutual datasets for the purpose of testing and assessment, making it tough to match with other techniques. We apply our proposed approach to an open ICS dataset [11], generated from a gas pipeline SCADA system, for confirmation, and demonstrate that it considerably outclasses state-of-the-art methods. Experimental results show that the proposed approach HML-IDS outperforms the benchmark models with an accuracy rate of *97%*.

This study makes the following contributions:

- An automated multi-level intrusion detection approach is proposed to detect anomalies in SCADA networks to help prevent damage to critical infrastructures.
- An algorithm is proposed for both reliable feature extraction and multi-level anomaly detection.
- Assessment results of the suggested approach on a real gas pipeline data advocate that the anticipated approach is precise in predicting anomalies and outperforms the existing approaches.

The remaining sections of the paper are settled as follows: Section II presents the related work, while detailed presentation of the proposed approach is outlined in section III. The evaluation process and results are explained in Section IV. Section V describes the advantages of the proposed approach while the threats to validity are described in Section VI. Lastly, Section VII concludes the paper and advises forthcoming work.

II. RELATED WORK

This segment presents the current associated research to cultivate intrusion detection system for ICS. It is worth noticing that anomaly detection has been broadly applied as a protective measure for many years, e.g. programs that detect abnormal behaviors [12], [13], botnet detection [14] and detecting intrusions in Internet of Things [15]. Though conservative protective mechanisms may be reformed and usefully arrayed

to guard ICS against cyber-outbreaks [16], [17], still there exist numerous complications that obstruct this course.

The first broadly recognized cyber-attack on ICS was Stuxnet [18], revealed in 2011. Industrial control systems were targeted by the Stuxnet worm and changed the behavior of system at both the server level and client. This attack was initiated by a resentful engineer who infiltrated the system of sewage control in Maroochi Australia and triggered raw sewage outflow of about 264,000 gallons into adjoining watercourses [19]. The attacker injects the virus by making use of a removable drive, and ultimately succeeded to alter the program regulating ground nodes. Another incident on the Davis-Besse nuclear plant in Oak Harbor Ohio was reported in 2003. This attack was done through the Slammer Worm, the plant safety monitoring system went disconnected for a period of about five hours [20]. Some of the latest illustrations are taken place in Germany, where the security of a steel mill is breached [21]. Here the attack was originated by a process called javelin phishing emails, and another attack in 2015 on Ukrainian power companies which leads to substantial outage [22].

In a survey paper [23], the authors provide categorization and metrics for detecting intrusion and prevention in systems that are specific to SCADA networks. They discuss the complications and explicit necessities to build IDS specific to SCADA networks such as components that are having design insecurities, rigid timeliness in real time, restricted resources of computing and firm obligation of availability. The authors assessed their suggested techniques in relation to a group of metrics, such as specific-ness degree with respect to SCADA, self-defense and misjudgment investigation.

The writers also discuss open concerns and recommendations for IDS that are SCADA-specific. The authors of interrelated survey paper [24] emphases for the development of IDS for the wider category of structures: Cyber-Physical Systems (CPS). The authors designed their study on the basis of two metrics: the technique used for detection and audit files. The paper initially converses the key dissimilarities between ICT and CPS intrusion discovery tools, followed by comparing current approaches with respect to two above-mentioned design metrics. Principally the authors sum-ups the significant benefits and drawbacks of each dissimilar type of IDS, also they discuss the effectiveness of each IDS while implementing to CPS.

Cheung *et al.* [6] suggested one of the primary IDS for SCADA systems by creating regular behaviors of the system. The authors proposed a method based on level protocol to distinguish Modbus TCP behaviors and then encrypted by rules of Snort to detect behaviors that are anomalous. Owing to the stationary network structure and consistent communication design contained by process control systems, the writers state that such methodology based on model is reasonable for control networks and are capable to identify unseen outbreaks.

On the way to counterpart conservative methods that are blacklist-based, which are mostly in effect to identify recognized attack designs, an IDS centered on anomalies is suggested in [7], [25] to develop the regular model of behaviors over time by associating actions system with respect to their dependencies and happenings. Predominantly, the authors of the paper recommend that their suggested system delivers an encouraging base to fight Advanced Persistent Threat where knowingly slow-moving outbreak approaches are generally applied. The suggested IDS has been validated in a limited experimental situation in a real-world environment. The authors of [26] introduced a related kind of attack for ICS by presenting a technique that is aware of the sequence to identify attacks comprising as semantic attacks.

Both the survey papers highlighted that IDS that are specific to SCADA systems require to integrate standards and rules that are specific to SCADA environment. Yang *et al.* [9] proposed an IDS explicitly for IEC 60870-5 for ICS by making use of an inspection method on Deep Packet Inspection, where the suspicious behaviors are identified by implementing signature-based rules through Snort. The writers in [10] present a framework for smart grid systems by making use of stateful intrusion detection and determine the use of such context for IEC 61850 developed using an IDS tool Suricata. The suggested technique describes a group of rules (stateful in nature) which are then inspected with arriving network packets.

In another work [8], the authors focused on IDS for Modbus/DNP3 explicitly in addition to the IEC criterions. The authors recommend that IDS based on network is more appropriate for SCADA than IDS based on host level as they involve fewer assets and impeccably assimilate with SCADA. This paper proposed an IDS based on state in which a virtual image is formatted by predefined systems knowledge. This virtual appearance retains on updated by analyzing the incoming network packets altering the system corporeal conditions. If any incoming network package takes the system into dangerous state an alert would be upraised. The authors of [8] proposed a method capable of detecting unseen outbreaks, because the alert would be generated by configurations that are in critical state rather than an explicit attack. The writers additionally did State Proximity and Critical State Analysis to explain their approach for using critical state patterns [27]. Mutual abnormalities that are resulted using a sequence of apparently acceptable instructions can be identified by this approach.

In recent times, several ML techniques have been used to build abnormality-based IDS for ICS. Unambiguously, these IDS take advantage of the existing data to generate normal behavior state of ICS environment, then identify abnormalities which are irregular with the generated states, and consequently are capable of detecting new outbreaks. For instance, the authors of [28] applied classification methods based on one-class (Support Vector Data Description -SVDD and the Kernel Principal Component Analysis -KPCA) for detecting anomalies in SCADA systems. In another attempt [29], Statistical Bayesian Networks were applied to optimize the accuracy of detecting anomalies in SCADA networks. Their study commendably condensed the rate of false positives by

merging several abnormality detection techniques such as invariant induction and n-grams. The authors of [30] proposed an IDS for smart grid SCADA networks based on Bloom filter.

Linda *et al.* [31] proposed a deep learning based IDS for critical infrastructures. The authors combined two neural network learning algorithms to develop their IDS. Similarly, the authors of [32] used a backpropagation algorithm to form the neural network in order to detect anomalies in a network-based intrusion detection system. In another study [33] a deep learning based anomaly detection method is proposed. The authors implement Restricted Boltzmann Machine (RBM) and a deep belief network to build their IDS. Feng *et al.* [34] proposed multi-level anomaly detection method for ICS using LSTM networks. They also used bloom filter to detect anomalies in SCADA system. The authors of [35] proposed IDS using Deep belief neural networks. The authors of [36] proposed techniques based on Hierarchical Neuron Architecture based Neural Network (HNA-NN) and an Intrusion Weighted Particle based Cuckoo Search Optimization (IWP-CSO). Their main focus was to detect SCADA network intrusions on the basis of optimization. Similarly, Yin *et al.* [37] proposed a deep learning based IDS by making use of recurrent neural networks (RNN-IDS). In another study [38] the authors of proposed a classification model based on deep learning. The authors applied stacked non-symmetric deep autoencoder (NDAE) to develop their IDS. The authors of [39] proposed an anomaly detection scheme based on hybrid deep learning method. The applied their detection technique on software-defined networks (SDNs) in order to enhance the reliability. In another study [40] the authors of proposed a deep learning based anomaly detection (DLAD). The authors detect anomalies by making use of video processing techniques.

The authors [41] of presented an anomaly based semi-supervised IDS. They used Generative Adversarial Networks (GANs) to train their model and suggested an end-to-end deep design for IDS. They build their model by training only from normal flow of traffic data. In an attempt to detect novelty, the authors of [42] also proposed one-class classification method (ALOCC) based on GANs. Their proposed architecture comprises of two deep networks: one for the detection of novelty and one for detecting the outliers. They used image and video dataset to support their framework. In a similar attempt the authors of [43] proposed *AnoGAN* method for anomaly detection. They used deep convolutional GAN to build their model by making use of unsupervised learning. Sabokrou *et al.* [44] proposed a method for detecting irregularities in images and videos. They used GANs based deep models in unsupervised or self-supervised configurations for the training of their classifiers. The authors of proposed a Deep-Cascade [45] based method for the detection of anomalies. Their approach is based on cubic-patch based method. They developed cascade of classifiers using deep networks to build their model. In another study [46], the authors proposed deep-anomaly method for fast detection of anomalies. They build their model by using supervised fully convolutional neural networks (FCNs), and then shifted it into an unsupervised FCN to detect anomalies.

III. APPROACH

A. OVERVIEW

The suggested approach initiates by reading the dataset and then performing some series of operations. The overview of the proposed approach is shown in Fig. 1. For the anomaly detection process, first, we download the data from the source. Next, we apply the preprocessing techniques on the collected data, which involves categorical labeling, standardization and normalization. Then, we employ dimensionality reduction techniques to select optimum features. Next, we create a signature database using Bloom filter and predict anomalies in the first level. Finally, we train and test an instance-based categorizer to predict anomalies at the second level. Each of the fundamental steps is presented in the subsequent sections.

B. RELIABLE FEATURE EXTRACTION

Feature extraction is a wide-ranging word for means of fabricating groupings of variables to cope with the complications (analyzing multifaceted data) whereas still defining the numbers with adequate correctness. Numerous ML experts believe that accurately augmented feature extraction is a vital step to construct an enhanced model.

Conferring the above motives while accomplishing the aims such as reliable extraction of features from the available dataset, increasing detection rate (DR), reducing false alarm rate (FAR) and the cost of computation, a reliable feature extraction approach for the dataset at data feature retrieval (DFR) is discussed in this section.

A feature extraction algorithm is presented in Algorithm 1. The loop on line 1 of the algorithm 1 initializes the reading process by loading the dataset. The for loop on line 4 standardize and normalize the dataset values. The next segment on the algorithm 1 reduce the dimensionality of the dataset by making use of three dimensionality reduction techniques: Principal Component Analysis (PCA), Canonical Correlation Analysis (CCA) and Independent Component Analysis (ICA). After balancing/re-sampling the dataset by using AllKNN sampling technique, the fuse features from the three reduction techniques are then used by the classifiers for prediction of anomalies.

1) PREPROCESSING

One of the vital preprocessing steps in data mining is standardization. This step is used to normalize values of features from a diverse vibrant array into a definite range. Normally, a dataset usually contains some values, with an outsized or boundless variability, these type of values will intensely upset the outcome of the analysis.

Al Shalabi *et al.* [47] proposed that preprocessing is essentially crucial before consuming any data investigation

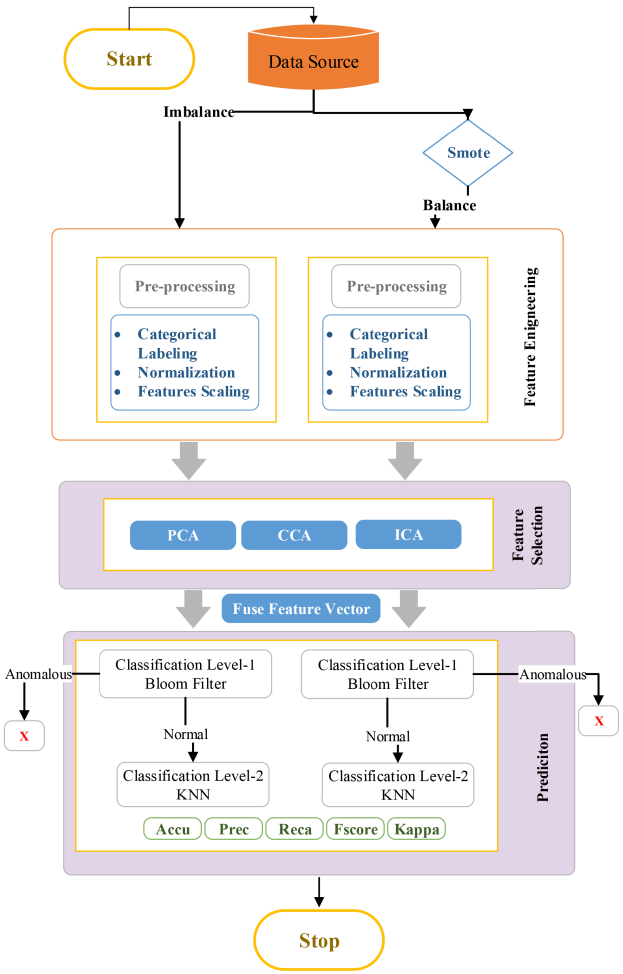


FIGURE 1. Overview of the proposed approach.

procedures to boost performance outcomes. Dataset standardization or feature scaling is amongst the preprocessing procedures in data examination, in which the data attributes are mounted to collapse in an indicated series. In order to make the raw data clean, consistent and noise free, data preprocessing techniques [48] are useful. The accuracy of the classifying algorithm can be increased significantly by making use of data normalization techniques. It is observed from the literature that there is no commonly well-structured statute for standardizing the datasets and therefore the selection of a specific scaling method is fundamentally up to the decision of the user.

2) STANDARDIZATION

The process of standardization makes sure that features are positioned nearby 0 with a standard deviation of 1. This step is not only a general requirement for numerous ML algorithms, but it is also important if we are matching values that have dissimilar units. The values will be rescaled after the standardization so they will have the standard usual dissemination properties with $\mu = 0$ and $\sigma = 1$. Where the mean is μ and the standard deviation is σ, standard totals of the instances

Algorithm 1 Feature Extraction and Generating Feature Matrix

Entail: W

1: *For* $i = 1$ *to* O *do*
2: $df \leftarrow read - dataset : [initialization]$
3: *End for*
4: *For j steps do* $//standardization$
5: *Scale values according to Eq.* (1)
6: *End for*
7: *For* $L = 1$ *to n do*$//dimensionality\ reduction$
8: *For* $Q = 1$ *to k do*
9: $P^1 = PCA(Y_L)$
10: $P^2 = CCA(Y_L)$
11: $P^3 = ICA(Y_L)$
12: *End for*
13: *End for*
14: $df \leftarrow Allknn()\ //re - sampling\ the\ dataset$

Confirm: FM // The order of FM is $n \times P^k$ and FM can be used for testing or training

are computed as follows:

$$Z = \frac{\chi - \mu}{\sigma} \tag{1}$$

Instinctively, we can take gradient descent as a noticeable example; with features exist on diverse ranges, certain values may update quicker than others as the feature values x_j show a part in the updates

$$\Delta w_j = -\eta \frac{\partial J}{\partial_{wj}} = \eta \sum_i (t^{(i)} - o^{(i)}) x_j^{(i)}, \tag{2}$$

so that

$w_j := w_j + \Delta w_j$, where $\mu = 0$ is the rate of learning, target class label being t, and o is the concrete outcome.

3) DIMENSIONALITY REDUCTION

Dimensionality reduction in ML is a manner of decreasing the number of arbitrary variables under contemplation by attaining a group of key features. This process can be separated into either feature selection or feature extraction.

Employing the concept of extracting features, assume the succeeding definitions to comprehend the use of notion upon data, which is essentially the original dataset:

- Let $W =$ [instances of original dataset] and $Y \subseteq W$.
- Y_N is the n^{th} instance in W and $N = 1, 2, 3, \ldots, m$, in which m is the complete amount of records/instances in W.
- Let $Y_N \times P^k$ is the order of the feature vector V, in which $k = 1, 2, 3$, $P^1 =$ Principal component analysis (PCA), $P^2 =$ Canonical correlation analysis (CCA) and $P^3 =$ Independent Component Analysis (ICA).

From algorithm 1, *FM* (feature matrix) can be generated for training data (i.e W) is given, where W is given as input from the original dataset. The repetition of algorithm 1 is

TABLE 1. Feature selection comparison before and after dimensionality reduction.

Reduction Method	Primary Features	Reduced Features
PCA	20	15
CCA	20	16
ICA	20	15

relative to n, and k is reliant on the quantity of features. The feature matrix's dimensionality is also reliant on k which can be amplified by embracing further statistical likenesses of Y_N. The function PCA() in algorithm 1 is used to capture difference (variance) in data. Features that have low dimensions data and are uncorrelated are ignored. Similarly, the function CCA() finds the supreme correlation. The intrinsic subspace of the primary feature space is defined by this method and rest of the features are eliminated. Moreover, the function ICA() latches the directions where the data have maximum statistical independence. The features with negative correlations are neglected in the feature selection process. Finally, the feature matrix FM is spawned. A comparison of primary and reduced features can be found in table 1.

4) BALANCING/RE-SAMPLING THE DATASET

As most of ML algorithms are constructed on the standard basis of Occam's razor principle, classifiers usually perform below par on datasets that are imbalanced in nature. With imbalanced data, the most naive assumption is habitually the one that categorizes nearly all samples as negative [49]. This issue of imbalanced datasets can be reached out from several core ways. One technique is to undersample the majority class of instances or oversample the minority class in the preprocessing step. Another approach is to devise a method based on SMOTE [50] technique.

a: THE AllKNN METHOD

The technique used in this paper (AllKNN [51]) is based on an instance-based learning algorithm known as the k-nearest neighbor (k-NN). The neighbors of given samples can be found by making use of the k-NN algorithm. This technique categorizes an occurrence conferring to the majority session of its k nearest neighbors. In order to illuminate the evaluated method, let assume the following:

- Let TS = [actual training set] and $S_s \subset TS$, acquired by using k-NN based technique.
- Now, assume that TS has n samples $x^1, x^2, \ldots, x^n$.

Every sample x of TS (of S_s also) has k nearest neighbors. AllKNN is basically a batch algorithm that makes k repetitions. At each iteration, it reports as anomaly any occurrence that is not categorized as acceptable by its k nearest neighbors. After finishing all repetitions, the algorithm eliminates all samples identified as anomalies.

C. PROBLEM STATEMENT

ADS for SCADA networks are habitually built by observing the system data amongst ground devices like sensors, PLCs, and actuators. To preserve the notion of generality, we exemplify the network data traded between devices in SCADA network as a series based on time, $A = \{a^{(1)}, a^{(2)}, \ldots, a^{(n)}\}$, in which every point $a^{(t)}$ in the chain is a vector in k-dimensions $\{a_1^{(t)}, a_2^{(t)}, \ldots, a_k^{(t)}\}$, where the elements tally to k features that can be mined from data (packet) among the devices. This approach based on package contents will categorize whether the incoming network package $a^{(t)}$ is abnormal exclusively contingent on the structures of $a^{(t)}$. While the classification based on time-sequence level will categorize the packets on the basis of packet contents along with a restricted quantity of formerly perceived packets. In this study, we present an approach which combines two stages of anomaly detection into a single hybrid IDS approach.

D. LEVEL 1: PACKET CONTENTS LEVEL DETECTION USING BLOOM FILTER

As, in voluminous situations the communication and network configuration exists amongst devices in SCADA are deliberated to be reasonably steady, we assume the regular behavior of network data exchanged among devices can be witnessed using a satisfactory dataset based on time-series A^N without the existence of abnormalities (A^N can be acquired by controlling the SCADA network in a tight *air-gapped* parting for a time frame). We record a regular sketch of system data by instituting a database of signatures for the A^N dataset, and during the detection stage, those packages are classified as anomalies whose signature cannot be found in the signature database.

1) PACKETS SIGNATURES GENERATION

Defining signatures for network packages is a vital step in Bloom Filters. Signatures can be generated by making use of all the network packages features. We attempt to make the most use of the available features. The signatures generated in this work are based on the method proposed in [34]. Unambiguously, the packet signature generation implicates a significant phase in which we convert the actual feature vector $a^{(t)} = \{a_1^{(t)}, a_2^{(t)}, \ldots, a_k^{(t)}\}$ of a subjective packet to a vector in o-dimensions ($o \leq$ m) $b^{(t)} = \{b_1^{(t)}, b_2^{(t)}, \ldots, b_o^{(t)}\}$, where every component $b_i^{(t)}$ is either the discretized valued feature or a discrete feature. Then, the following function generates the package signatures:

$$d(a^{(t)}) = f(b_1^{(t)}, b_2^{(t)}, \ldots, b_o^{(t)}) \tag{3}$$

which fulfills:

$$f(b_1^{(t)}, b_2^{(t)}, \ldots, b_o^{(t)}) = f(b_1^{(t')}, b_2^{(t')}, \ldots, b_o^{(t')}) \Leftrightarrow b_i^{(t)} \forall i \in (1, 2, \ldots, o) \tag{4}$$

Instinctively, the generating function is $f(\cdot)$, which allocates a distinctive value to every dissimilar grouping of its strictures. The modest approach to describe $f(\cdot)$ is to do

parameters concatenation to a string by making use of a separator such as a special character.

2) ANOMALY DETECTION USING BLOOM FILTER

Since the SCADA network monitors have limited resource both in terms of memory and computing, a Bloom filter has been used to proficiently stock the signatures of regular network patterns and identify anomalies afterward.

Explicitly, a Bloom filter is a memory efficient probabilistic data structure to the customary hash lookup. This method is applied to check whether an element is a participant of a group or not. It can be also applied to neatly exemplify a group of input data through two constituents: a group of k autonomous hash functions $h_1, h_2, \ldots, h_k$ with series $\{1, 2, \ldots, m\}$ and an m-bit vector v. All elements in the m-bit vector are initialized to 0, every element in the hash functions set is map to 1 of the m locations in v. To *insert* e (an element) into the vector v, the element is hash k times by making use of the (predefined hash functions) $h_1, h_2, \ldots, h_k$ to yield a sequence of k values every reaching from one to m. Then the vector bits v at the positions equivalent to $h_1(e), h_2(e), \ldots, h_k(e)$ are fixed to 1.

A specific bit may be fixed to one several time by a diverse number of inputs; consequently, a component of v fixed to 1 constantly remains 1. The checking of e (an element) in the group can merely be done by again hashing e k intervals by means of similar $h_1, h_2, \ldots, h_k$ (the identical hash functions), and then testing if the entire locations $h_1(e), h_2(e), \ldots, h_k(e)$ in v are 1. The element is anticipated not to exist in the set if any of these k bits is 0. Else, it is likely that the element e exists with a finite probability.

There is a possibility of high false positive rate but a very low possibility of false negatives. The *trade-off* amongst the required memory and the rate of false positive can be governed by *fine-tuning* the m and k parameters.

We use Bloom Filter as anomaly detector because of its distinctive advantages such as memory efficiency, constant lookup time, fast and light-weighted. Explicitly, let our Bloom filter is BF and SN is the group of all regular signatures of the data in the dictionary, we add each generated signatures $d \in SN$ into BF throughout the period of training. Therefore, the anomaly prediction function can be defined as:

$$F_p(a^{(t)}) = \begin{cases} 1, \ if \ d(a^{(t)}) \notin BF \\ 0, \ if \ d(a^{(t)}) \in BF \end{cases} \tag{5}$$

where $a^{(t)}$ is categorized as abnormality if $F_p(a^{(t)}) = 1$, else it is concluded that the incoming network package is approved by our proposed Bloom filter based anomaly identifier.

E. LEVEL 2: ANOMALY DETECTION USING INSTANCE-BASED LEARNER

Network packages can still reveal abnormal behavior after they pass our Bloom filter anomaly detector, which can only be identified given the reflection of seeing previous data packages. Hence, in order to detect zero-day attacks, we also intend to use an instance-based ML classifier, K-nearest neighbors (Knn) to further refine the anomaly detection process. The use of this technique will also assist our approach to decrease the rate of false positive, unlike Bloom Filter.

KNN algorithm is centered on a function (distance function) that calculates the dissimilarity or resemblance between two occurrences. The standard Euclidean distance $d(a, b)$ concerning two occurrences a and b is defined as

$$d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{6}$$

where a_i is the i^{th} featured component of sample a, b_i is the i^{th} featured component of the occurrence b and n being the entire amount of features in the data set.

Consider DS as the design group for knn based classifier. The entire quantity of instances in the group is T. Assume $DC = \{DC_1, DC_2, \ldots, DC_{CL}\}$ are the CL discrete class tags that are existing in T. Consider input vector $= av$, for which the category need to be identified. Assume b_i symbolizes the i^{th} vector in the design group T. Knn is used to calculate k closet vectors in the design group T to the av. If the most number of k closet vectors have their tag as DC_j, then av (input vector) is identified to class DC_j.

One of the noticeable things to look in knn is to choose the finest value of k. It normally rests on the distribution and size of the dataset. If k is too large unrelated instances may be incorporated in the neighborhood, or if k is too small some related instances may not be counted in. Whichever way this would lead to a drop in accuracy during classification. We investigated with diverse values of k, but for simplicity, uncomplicatedness and regularity a k of 3 is used in this work. And also, we got the highest accuracy rate at k of 3.

F. THE COMBINED ANOMALY DETECTION APPROACH

After presenting both our package contents level and instance-based ADS, now we present the joined hybrid multilevel approach of these two models.

Explicitly, Fig. 2 presents the schematic organization of the collective proposed approach. As can be realized from the diagram, the pooled approach is reasonably forthright. Apparently, when packet data is being investigated, firstly our Bloom filter abnormality identifier will check its signature in the dictionary, and the package will be categorized as an anomaly if Bloom filter does not contain its signature. As the anomalous packages will always be classified as anomalous by second level classification, so there is no necessity to send anomalous packages identified by Bloom filter to the instance-based abnormality detector. It is worth noticing that this approach should work sound as the Bloom filter's rate of false positive can be projected by the error of validation all through the teaching stage, and therefore can be well tuned. The proposed approach's flow diagram is presented in Fig. 1.

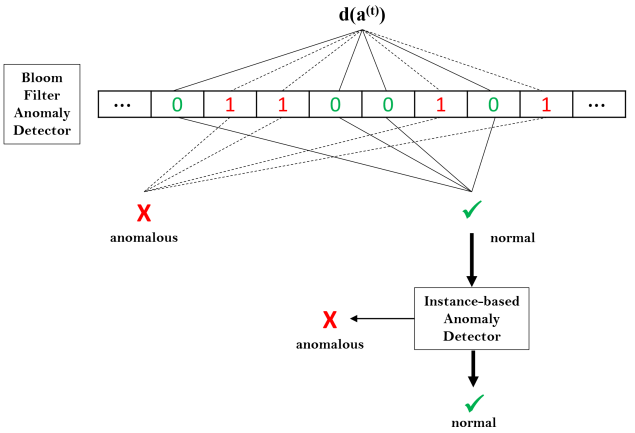


FIGURE 2. The schematic structure of the proposed approach.

Moreover, if the packet contents are approved by our first level of anomaly detector, at that point our instance-based anomaly identifier will categorize whether it is truly abnormal or not by testing whether its behavior lies within the anticipated credible signatures (on the basis of feature vectors of earlier seen packets) or not. The package data will be used as input to the classifier for categorization of future incoming packets whether they are categorized as regular or abnormal.

IV. EVALUATION

In this segment, we assess the effectiveness of the suggested approach on SCADA gas pipeline dataset.

A. RESEARCH QUESTIONS

The assessment explores the succeeding investigation queries:

- **RQ1:** How accurate is the suggested approach in detecting anomalies?
- **RQ2:** Does the planned approach outclass the state-of-the-art approaches in detecting anomalies in SCADA systems? If yes, to what extent?
- **RQ3:** Does dataset balancing/re-sampling help to improve the performance of the proposed approach? If yes, to what extent?
- **RQ4:** Does instance-based learner perform better than other ML algorithms in predicting anomalies in SCADA systems?

In reply to the investigation query (RQ1), we present the results of our suggested approach in table 4. The accuracy rate of our proposed approach is *97%*, which means the proposed approach is significant enough to be used as an intrusion detector in SCADA networks.

In reply to the investigation question (RQ2), we match the effectiveness of the suggested approach with the state-of-the-art approaches [34], [52]–[54] in order to find out the efficiency enhancement of the projected approach. The comparison shows that our approach shows significant performance improvement in terms of accuracy over the benchmark models.

In reply to the investigation question (RQ3), we figure out that dataset balancing/re-sampling indeed improves the performance of the anticipated approach to a great extent. The balancing/re-sampling process improved the accuracy by *7%*, precision by *6%* and recall by *29%*.

In reply to the investigation question (RQ4), several ML algorithms are considered and matched with the suggested approach to assess the performance in terms of accuracy, precision, recall, and f1-score.

B. DATASET

We used the dataset proposed in [11] for learning and testing of our joined anomaly identifying approach. The attacks on critical infrastructures like SCADA can be detected with the use of an IDS. The operator can be notified to any unusual activity which will help to prevent damage. These SCADA systems IDSs are enhanced by training them with data logs that signify the actual traffic of SCADA network. The gas pipeline system dataset used here was provided by Mississippi State University's in-house SCADA lab. In order to analyze the performance of an IDS this SCADA dataset can be used. This dataset contains artifacts of 35 cyber-attacks and can be used to train and test classifiers used by IDS.

The data contain both the regular operation data and real cyber outbreaks. Explicitly, the gas pipeline system comprises of a slight sealed pipeline coupled to a compressor, a relief valve based on solenoid controlling and a pressure meter. The overall system maintains pipeline pressure by means of a control scheme known as a proportional integral derivative (PID). The accompanying SCADA system makes use of the Modbus protocol for communication at the application layer. An automatic attack generation technique AutoIt [55] is used to initiate attacks. This scripting language based technique can drop, delay, inject and modify network traffic data. The exchanged network packages are documented in a log file with a timestamp. Every network package comprises of Modbus payload and a header, with twenty distinct features that are warehoused in Attribute Relationship File Format (ARFF). Table 2 itemizes some of these features in particulars.

The AutoIt [55] script arbitrarily selects to send legitimate instructions or execute cyber-attacks. The dataset contains 214,580 regular network packages and 60,048 attacked packets. After balancing/re-sampling, the dataset contains an equal number of normal and abnormal packets (218,979 normal packets and 218,979 abnormal packets).The attack categories that are considered in this dataset are: reconnaissance attacks, denial of service attacks, command injection attacks and response injection attacks. These four groupings are supplementary separated into 7 particular kinds of attacks as defined in table 3.

Fig. 3 represents the graphical structure of the original and balanced/re-sampled dataset. Additional information regarding this dataset can be found in [11]. The dataset is publically available at the webpage

TABLE 2. Features in ARFF [11].

Feature/Attribute	Description
address	MODBUS slave device's station address.
setpoint	The set point of the pressure, only for the use in Automatic system mode.
length	MODBUS packet length.
function	MODBUS function code.
gain	PID gain.
deadband	PID dead band.
pressure measurement	Pressure measurement.
system mode	(2) for automatic system mode, (1) for manual, or (0) for off.
cycle time	PID cycle time.
solenoid	Relief control valve; (1) for opened or (0) for closed. Only for the use in manual mode.
control scheme	The control scheme to govern set point. (0) for pump or (1) for solenoid.
time	Time stamp.
reset rate	PID reset rate.
command response	(1) for Command or (0) for response.
pump	Pump control; (1) for on or (0) for off. Only for the use in manual mode.
rate	PID rate.
binary result	Binary class; (1) for attack data or (0) for normal data.

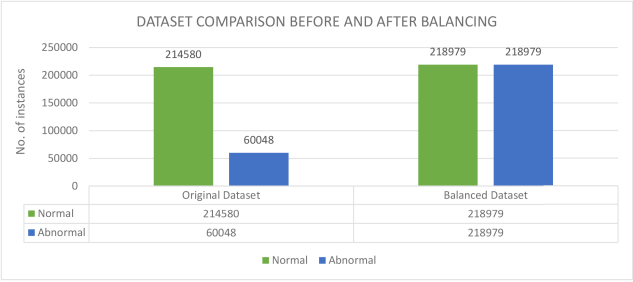


FIGURE 3. Dataset before and after balancing/re-sampling.

TABLE 3. Categories of attacks in the dataset [11].

Attack ID	Attack Abbreviation	Attack Description
1	NMRI	Naive Malicious Response Injection
2	CMRI	Complex Malicious Response Injection
3	MSCI	Malicious State Command Injection
4	MPCI	Malicious Parameter Command Injection
5	MFCI	Malicious Function Code Injection
6	DoS	Denial of Service
7	Recon	Reconnaissance

https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets.

C. PROCESS AND METRICS

1) PROCESS

Following points shows the steps of implementation for the assessment of the suggested approach.

- We first download data from the source and perform preprocessing as stated in Section III.
- Second, we employ dimensionality reduction techniques to select optimum features.
- Third, we apply data balancing/re-sampling technique to balance the dataset.
- Fourth, we manually remove the abnormalities to capture the normal behavior of the system.
- Fifth, we create a signature database using Bloom filter.
- Sixth, we train instance-based, Random Forest, AdaBoost, Net(MLP) and Quadratic Discriminant Analysis classifiers on the training data.
- Seventh for each training data we predict anomalies using the above mentioned trained approaches.
- Lastly, we compute the accuracy, precision, recall, F1-score, sensitivity, specificity and kappa for every classifier for their assessment.

2) METRICS

In most of the problem assessment process, accuracy might not be adequate to evaluate and define the reliability and validity of the model. In every diverse picture of problem assessment scheme, some statistical evaluating tools may provide different evaluation results [56].

We have assessed the performance of our proposed approach by making use of 10-fold cross-validation technique. Normally, in sculpting a binary cataloging problem, a confusion matrix results in the following four diverse proceedings: i) true positive (TP) defines the total of positive instances being categorized as positive, ii) true negative (TN) signifies the number of negative occurrences being forecasted as negative iii) false positive (FP) specifies the number of positive instances being anticipated as false and iv) false negative (FN) indicates the number of positive illustrations mistakenly anticipated as negative.

One of the widely used metric for evaluation of model perfection and performance of the classifier is Accuracy. Specificity also labeled as the rate of true negative, means it measures the proportion of undesirable (negative) classes which has been categorized as True. While sensitivity is the measure of the likelihood of desirable classes (True classes) being categorized as True. In order to calculate one-sidedness in a model, a statistical tool F-measure is used. It can be obtained by taking the weighted average of precision and recall. One of the utmost appreciated and a commendable statistical tool is receiver operating characteristics (ROC) curve [56], [57]. From the past several years it has been one of the most trustworthy measurers for performance and authenticity of a model. It can be styled as plotting sensitivity on the y-axis to specificity on the x-axis. Cohens-kappa statistics is another widely robust measure for evaluation. This approach measures the inter-observer agreement. If the resulting measured value is approaching 0, then it means an arbitrary guess or chance, and if the value approaches 1, it means a perfect agreement exists.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$f - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

$$Cohens - kappa(k) = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{13}$$

D. TRAINING AND CLASSIFICATION AT THE DE LEVEL

This segment presents the training and categorization of DE by means of two supervised ML techniques such as Bloom filter and k-NN. For detailed functioning explanation of proposed approach at DE, an HML-IDS algorithm (algorithm 2) is specified.

During the training and testing phase in algorithm 2, *FMs* are specified as input alongside class tag *CL* since supervised learning is implemented at the *DE* level. The size of *L* is equivalent to *m* and *CL* are binary (0 for regular and 1 for irregular). As can be perceived from the algorithm 2, line 1 initializes the process by creating a signature dictionary for normal behavior of the system. The loop on line number 4 check the signature of the incoming network packet in the signature dictionary, and if the package is perceived as non-anomalous then it is forwarded to the next classifier.

Knn is implemented due to the recognition of *FM* as a real feature matrix and accomplishing least cost of computation at the *DE* level. From algorithm 2, it can be noticed that each test s^{th} sample can be transmuted into *FMs* and categorized as normal or anomalous using Model-Training (see line 9 of HML-IDS algorithm 2).

Algorithm 2 HML-IDS

INPUT: *Balance Dataset*
OUTPUT: *Anomaly Prediction*
1: *For* $q = 1$ *to* m *do*//*generate Signatures*
2: *Generate signatures by Eq.* (3)
3: *End for*
4: *For r steps do* //*Classification level one* : *using Bloom filter*
5: *Classify* $\rightarrow$ *normal or abnormal* //*using the following equation* *Classify by Eq.* (5)
6: *End for*
7: $Model - Train = KNN(FM[training], CL)$ //*Classification level* 2 : *using instance* $-$ *based learner*
8: *For* $S = 1$ *to u do*
9: $Classify(Model - Train, FMu[test])) \rightarrow$ *FMs normal or anomalous*
10:*End for*

TABLE 4. Performance comparison of proposed approach with other anomaly detection methods on the similar dataset.

Approach	Accuracy	Precision	Recall	F1-score	Kappa
Proposed	**0.97**	**0.98**	**0.92**	**0.95**	**0.94**
BLF	0.89	**0.97**	0.67	0.78	0.71
RF	0.91	0.93	0.81	0.86	-
PCA-SVD	0.17	0.64	0.27	0.27	-
F. Cheng et al.	0.92	0.94	0.78	0.85	-
AutoMLP	0.95	0.96	0.90	-	-
GMM	0.45	0.79	0.44	0.59	-
MOLESTRA	0.93	-	-	-	0.74

E. RESULTS

We evaluate the classification results to check the performance of our combined anomaly detection approach for the test set. The metrics discussed in the prior section are used in our experiments to evaluate the effectiveness of our proposed anomaly detection approach. Salvaging experimental observations, all the programming work is implemented in

TABLE 5. Categorized results comparison of each attack type.

Approach	Detection rate						
	NMRI	CMRI	MSCI	MPCI	MFCI	DoS	Rconn
Proposed	**0.93**	**0.76**	**0.68**	**0.85**	**1.00**	**0.98**	**1.00**
BLF	0.82	0.67	0.53	0.74	**1.00**	0.94	**1.00**
RF	0.75	0.49	0.51	0.13	0.06	0.95	**1.00**
PCA-SVD	0.45	0.19	0.62	0.66	0.54	0.58	0.54
F. Cheng et al.	0.88	0.67	0.62	0.80	**1.00**	0.94	**1.00**
GMM	0.31	0.33	0.66	0.64	0.32	0.15	0.72

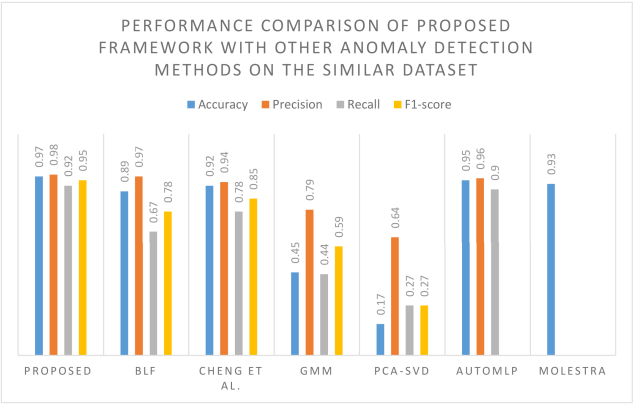


FIGURE 4. Performance comparison of proposed approach.

Anaconda (Python 3.7) using Sypder as IDE. The computer machine's specification comprises of Intel core i-5 processor @ 3.20GHz, 8 GB RAM and 64-bit Windows based OS. The average total time of identifying anomaly using our proposed approach is about 0.109 milliseconds. The memory required to pile the two level anomaly detection approach is 1040 KB. Both ought to be acceptable for present SCADA systems.

1) RQ1: PROPOSED APPROACH RESULTS

We demonstrate the results of our collective anomaly detection approach on the assessment metrics in Fig. 4 and table 4. It can be seen that our proposed approach has outperformed the existing models by a significant margin. Our proposed approach achieved as high as *97%* accuracy, *98%* precision, *92%* recall rate and *95%* F score. The accuracy of our model falls with the growth of k, which specifies some abnormal packets are somewhat adjacent to regular packets. But, we have tested our model with varying number of k, and it has been noticed that if we increase the k value up to 200, we still got accuracy rate of *95%*. This reveals that the selection of k plays a vital part for the effectiveness of our proposed approach. More significantly, it is also observed that our selection of $k = 3$, attained the utmost F1-score at the detection level, implicating that the parameters tuning in our proposed approach is effective.

We also illustrate the detection rate (recall) of abnormal packets in every type of attacks. Table 5 represent the results in detail. It can be noticed from table 5 and Fig. 5 that the proposed approach performs better in detecting anomalies in almost every attack situation.

We also note that the rate of detection for MSCI and CMRI types of attack are lesser than other types of attacks. One of the possible reason is that these types of attacks unveil noisy behavior as they are linked to the physical processes. As a consequence, some attacks may be viewed as regular behavior since the deviancy instigated by these attacks can be preserved as regular noise.

2) RQ2: COMPARISON WITH OTHER MODELS

In order to persuasively demonstrate the usefulness of our joined approach for detecting anomalies/intrusions, we also compare the performance of our proposed approach with other anomaly detection techniques for ICS. Since the authors of [52] used the same dataset in their work, so we also compare the outcomes of our approach with their two unsupervised methods. These models are Gaussian Mixture Model (GMM) and Principal Component Analysis with Singular Value Decomposition method (PCA-SVD). The numbers for these two models are directly taken from [52], Feng *et al.* [34], AutoMLP [54] and Demertzis *et al.* [53]. To the best of our knowledge, these are the latest methods that could aid to forecast the intrusions in SCADA and has substantial results. Hence, we handpicked these studies for the assessment with the proposed approach.

The thorough outcomes are presented in Table 4. It can be noticed that our approach demonstrates considerably greater performance matched to the other methods. The models showing neighboring efficiency to our approach are the Bloom filter model, the Random Forest model, the AutoMLP [54] and the Feng *et al.* [34] model. But, their anomaly detection ability is still noticeably inferior to ours. The other two models (GMM, PCA-SVD) have comparatively reduced performance primarily because they are not proficient enough to deal with such complex formats of data.

The columns of the table 4 denote the accuracy, precision, recall, and F1-score. While, the rows represent the efficiency of each compared methods. From the Table 4, we witness

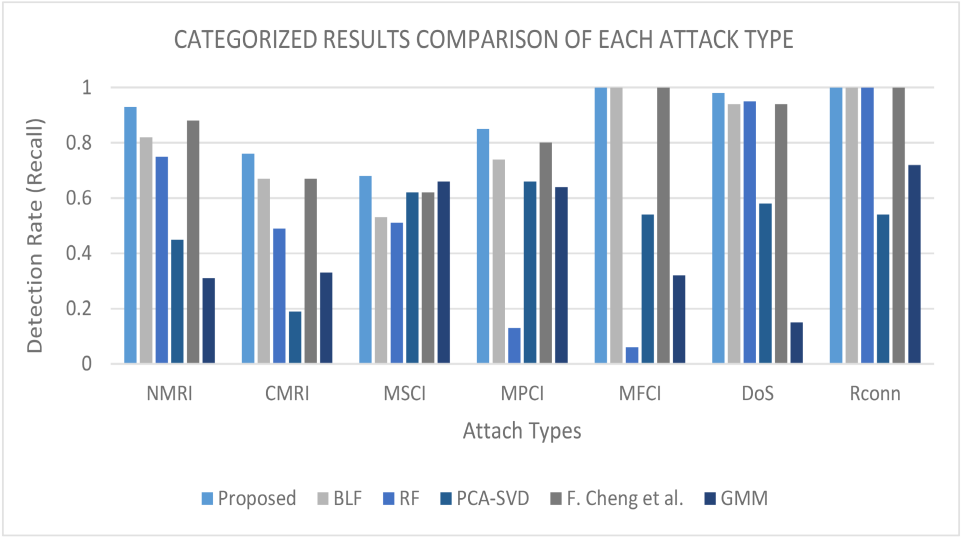


FIGURE 5. Categorized results comparison of each attack type.

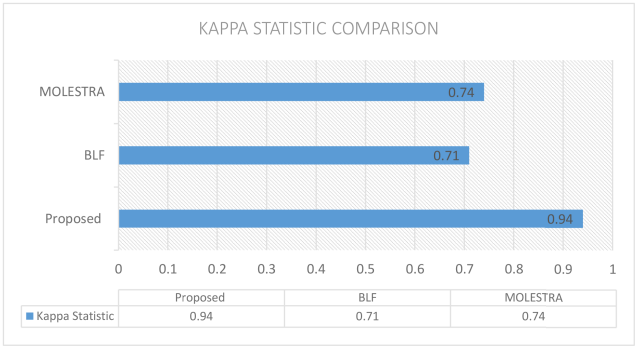


FIGURE 6. Kappa statistics comparison of the proposed approach.

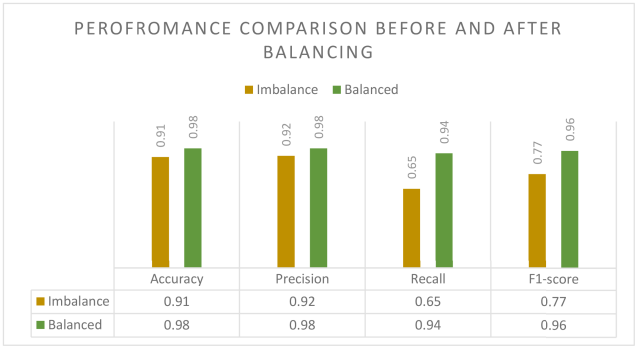


FIGURE 7. Performance comparison before and after balancing/re-sampling.

that the suggested approach outclasses the state-of-the-art methods.

From the Table 4, we mark the resulting interpretations:

- The suggested approach attains substantial enhancement in efficiency in terms of each assessment metrics. The enhancement in F1-score diverges from *9%* to *68%*.
- The suggested approach attains substantial enhancement in terms of kappa statistics with improvement diverges from *20%* to *23%*.

The results of kappa statistics are summarized in Fig. 6. The results show that our proposed approach is exhibiting kappa value that is closer to 1. It means in terms of kappa statistic evaluation; our proposed approach also beat other models.

3) RQ3: PERFORMANCE COMPARISON BEFORE AND AFTER BALANCING/RE-SAMPLING

A significant change of performance can be seen from Fig. 7.

Before the process of balancing/re-sampling, the classifier's accuracy is *91%* while the accuracy is raised to *97%* after re-sampling the dataset. Moreover, precision, recall and F score are also higher as compare to imbalance dataset. The results show that our balancing/re-sampling approach has explicitly increased the performance of our proposed approach.

The Fig. 8 show the ROC for the imbalanced dataset. As it can be seen from the ROC that the performance of the model is not so good for the imbalance dataset (the upper part of Fig. 8). While from the ROC of the balanced/re-sampled dataset (the lower part of Fig. 8), we can see that the true positive rate is better than imbalance dataset.

4) RQ4: COMPARISON WITH OTHER ML ALGORITHMS

Explicitly, we have applied several methods: a Bloom Filter (BF) method; a K-nearest neighbors method [58]; a Neural Net(MLP) method [59]; a AdaBoost method [60]; a Quadratic Discriminant Analysis method [61]; and a Random Forest (RF) method which is habitually reflected to be more appropriate for detecting outliers in mixture data [62] for abnormality discovery on the similar dataset. The motivation

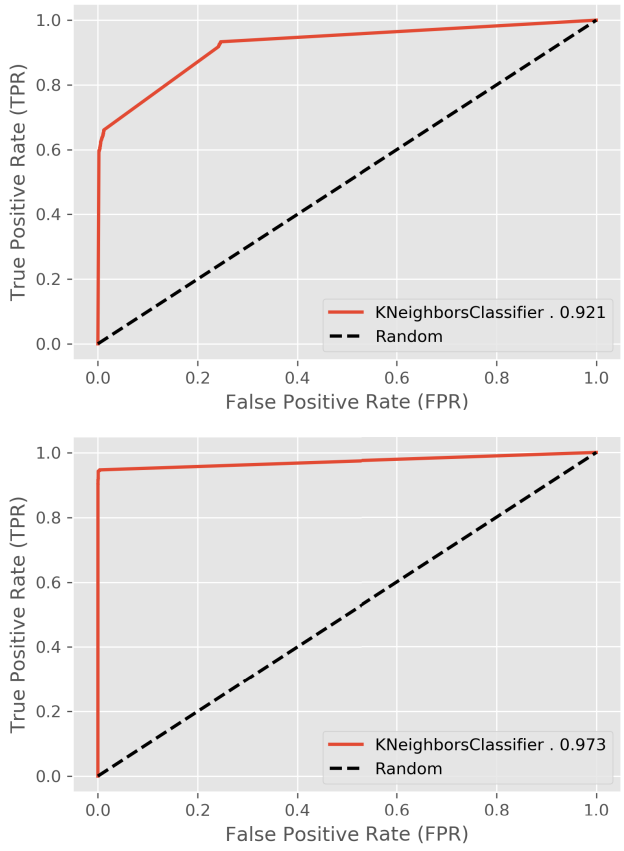
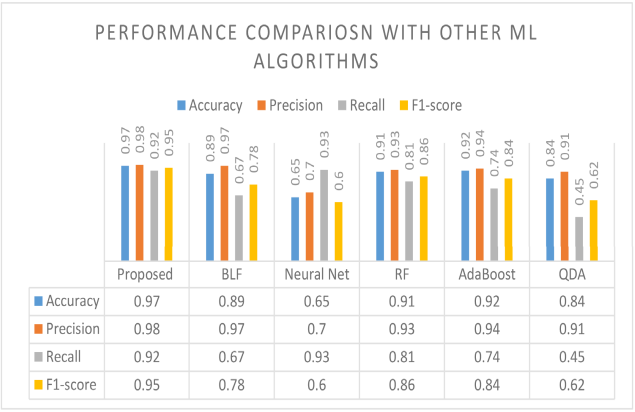


FIGURE 8. ROC comparison before and after balancing/re-sampling.



	Proposed	BLF	Neural Net	RF	AdaBoost	QDA
Accuracy	0.97	0.89	0.65	0.91	0.92	0.84
Precision	0.98	0.97	0.7	0.93	0.94	0.91
Recall	0.92	0.67	0.93	0.81	0.74	0.45
F1-score	0.95	0.78	0.6	0.86	0.84	0.62

FIGURE 9. Performance comparison with other ML methods.

to select these ML algorithms is their reputation with respect to usage practice and modest efficiency.

The performance outcomes of every classifier are shown in table data of Fig. 9. The rows of the table data in Fig. 9 signify the accuracy, precision, recall, and F1-score. While, the columns denote the performance of every matched approach. We also match their resultant efficiency in Fig. 9 that visualizes the variance.

The results in Fig. 9 shows that our proposed approach performed better as compared to other ML algorithms in terms of accuracy, precision, recall and F score with values *97%*, *98%*, *92%* and *95%* respectively. The other method that show good results near to our proposed approach is Random Forest with an accuracy of *91%*, precision *93%*, recall *81%* and F score of *86%*.

V. ADVANTAGES OF PROPOSED METHOD

We conclude the advantages of our proposed approach as follows:

- Due to the constant lookup time of bloom filter it is very fast to detect anomalies.
- It is capable to detect unseen (zero-day) attacks.
- It is competent to deal with the samples of data that are hybrid and complicated in nature.
- The training process of our model is simple and forthright since all the factors can be efficiently tuned to realistic standards by our suggested techniques.
- It shows great performance at the detection level, which is validated on an actual gas pipeline dataset matched with several other present anomaly prediction methods.
- The proposed approach achieved higher rate of accuracy and precision as compared to state-of-the-art one-class classification methods discussed in related work section II.

VI. THREATS

A. THREATS TO VALIDITY

A risk to build validity is the appropriateness of our metrics of assessment. In this work precision, recall, accuracy, and F1-score are used for the assessment of our proposed approach. Since, these metrics are customary standard and also embraced by numerous investigators [34], [52].

A threat to build validity is associated to the implementation of Allknn technique for balancing/re-sampling of the dataset. There are several data balancing methods, however, we select Allknn due to its relevance to our instance-based learner. Other balancing/re-sampling techniques may reduce the effectiveness of the suggested approach.

A threat to build interior validity is linked to the implementation of the methods. To soften the risks, the execution and outcomes are patterned. Nevertheless, there might be some unobserved bugs.

A risk to build exterior validity is interrelated to the generalization of our outcomes. We have only deliberated and explored the dataset that is linked to the gas pipeline of SCADA system. Therefore, the suggested approach may not do well for the dataset that is different than gas pipeline SCADA networks.

A hazard to exterior validity is a lesser amount of features. Hence, we use customary ML algorithms to assess the proposed approach. Deep learning methods can influence the performance of the proposed approach. But, they typically entail bulky training data and have a number of parameters to be tuned.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have suggested a multi-level approach for anomaly detection for ICS-specific systems, which articulately pools a Bloom filter based packet level anomaly finder and an instance-based time-series level abnormality detector. Like most other ML techniques, our proposed detection method entails a big dataset to let the models to be accurately trained. This paper has suggested a reliable feature extraction technique inspiring from data standardization at DFR. Moreover, our proposed approach also comprehend that classifiers accuracy can be improved by balancing/re-sampling the dataset. The extracted features which are the result of the advocated approach at DFR along with Bloom filter and Knn verified the trustworthy insights between regular and irregular behavior of the system.

Overall, the proposed IDS is capable to attain an acceptable efficiency while maintaining the cost of computation at a small level. In order to attain superior DR, it is planned in future to embrace more trustworthy or demonstrative feature mining approach at the DFR level. Moreover, we are also pursuing the means to accumulate more SCADA datasets to further investigate our approach.

For future work, we are planning to improve DR and overall performance of the proposed approach through deep learning methods.

REFERENCES

[1] S. Adepu and A. Mathur, "An investigation into the response of a water treatment system to cyber attacks," in *Proc. IEEE 17th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2016, pp. 141–148.

[2] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*. London, U.K.: Pearson, 2016.

[3] S. Kriaa, M. Bouissou, F. Colin, Y. Halgand, and L. Pietre-Cambacedes, "Safety and security interactions modeling using the BDMP formalism: Case study of a pipeline," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2014, pp. 326–341.

[4] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2012.

[5] *ICS-CERT Annual Vulnerability Coordination Report*, Dept. Homeland Secur., Washington, DC, USA, 2016.

[6] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Secur. Sci. Symp.*, vol. 46, 2007, pp. 1–12.

[7] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Comput. Secur.*, vol. 48, pp. 35–57, Feb. 2015.

[8] I. N. Fovino, A. Carcano, T. De Lacheze Murel, A. Trombetta, and M. Masera, "Modbus/DNP3 state-based intrusion detection system," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 729–736.

[9] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang, "Intrusion detection system for IEC 60870-5-104 based SCADA networks," in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2013, pp. 1–5.

[10] B. Kang, K. McLaughlin, and S. Sezer, "Towards a stateful analysis framework for smart grid network intrusion detection," in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res.*, 2016, pp. 1–8.

[11] T. H. Morris, Z. Thornton, and I. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," in *Proc. 7th Annu. Southeastern Cyber Secur. Summit*, 2015, pp. 3–4.

[12] K. Xu, K. Tian, D. Yao, and B. G. Ryder, "A sharper sense of self: Probabilistic reasoning of program behaviors for anomaly detection with context sensitivity," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, 2016, pp. 467–478.

[13] X. Shu, D. Yao, and N. Ramakrishnan, "Unearthing stealthy program attacks buried in extremely long execution paths," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 401–413.

[14] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. 17th Conf. Secur. Symp.* Berkeley, CA, USA: USENIX Association, Aug. 2008, pp. 139–154.

[15] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2661–2674, May 2013.

[16] A. Fielder, T. Li, and C. Hankin, "Modelling cost-effectiveness of defenses in industrial control systems," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2016, pp. 187–200.

[17] T. Li and C. Hankin, "Effective defence against zero-day exploits using Bayesian networks," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2016, pp. 123–136.

[18] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," *Symantec Secur. Response*, vol. 5, no. 6, p. 29, 2011.

[19] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Proc. Int. Conf. Crit. Infrastruct. Protection*. Springer, 2007, pp. 73–82.

[20] K. Poulsen. (2003). *Slammer Worm Crashed Ohio Nuke Plant Network*. [Online]. Available: http://www.securityfocus.com/news/6767

[21] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Ind. Control Syst.*, vol. 30, p. 22, Dec. 2014.

[22] D. Alert, "Cyber-attack against ukrainian critical infrastructure," Cybersecurity Infrastruct. Secur. Agency, Washington, DC, USA, Tech. Rep. ICS Alert (IR-ALERT-H-16-056-01), 2016.

[23] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: A survey and taxonomy," in *Proc. 1st Workshop Secure Control Syst. (SCS)*, vol. 11, 2010, p. 7.

[24] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surv.*, vol. 46, no. 4, p. 55, 2014.

[25] F. Skopik, I. Friedberg, and R. Fiedler, "Dealing with advanced persistent threats in smart grid ICT networks," in *Proc. ISGT*, Feb. 2014, pp. 1–5.

[26] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proc. 1st ACM Workshop Cyber-Phys. Syst. Secur.*, 2015, pp. 13–24.

[27] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 179–186, May 2011.

[28] P. Nader, P. Honeine, and P. Beauseroy, "LP-norms in one-class classification for intrusion detection in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2308–2317, Apr. 2014.

[29] J. Bigham, D. Gamez, and N. Lu, "Safeguarding SCADA systems with anomaly detection," in *Proc. Int. Workshop Math. Methods, Models, Archit. Comput. Netw. Secur.* Berlin, Germany: Springer-Verlag, 2003, pp. 171–182.

[30] S. Parthasarathy and D. Kundur, "Bloom filter based intrusion detection for smart grid SCADA," in *Proc. 25th IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, Apr. 2012, pp. 1–6.

[31] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 1827–1834.

[32] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA control system command and response injection and intrusion detection," in *Proc. eCrime Res. Summit*, Oct. 2010, pp. 1–9.

[33] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 195–200.

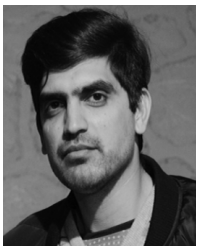
[34] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2017, pp. 261–272.

[35] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 339–344.

[36] S. Shitharth, "An enhanced optimization based algorithm for intrusion detection in SCADA network," *Comput. Secur.*, vol. 70, pp. 16–26, Sep. 2017.

[37] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[38] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[39] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.
[40] A. R. Revathi and D. Kumar, "An efficient system for anomaly detection using deep learning classifier," *Signal, Image Video Process.*, vol. 11, no. 2, pp. 291–299, 2017.
[41] B. Mohammadi and M. Sabokrou, "End-to-end adversarial learning for intrusion detection in computer networks," 2019, *arXiv:1904.11577*. [Online]. Available: https://arxiv.org/abs/1904.11577
[42] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.
[43] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Cham, Switzerland: Springer, 2017, pp. 146–157.
[44] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "AVID: Adversarial visual irregularity detection," 2018, *arXiv:1805.09521*. [Online]. Available: https://arxiv.org/abs/1805.09521
[45] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, Apr. 2017.
[46] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vis. Image Understand.*, vol. 172, pp. 88–97, Jul. 2018.
[47] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: A preprocessing engine," *J. Comput. Sci.*, vol. 2, no. 9, pp. 735–739, 2006.
[48] V. R. Patel and R. G. Mehta, "Impact of outlier removal and normalization approach in modified k-means clustering algorithm," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, p. 331, 2011.
[49] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 2004, pp. 39–50.
[50] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[51] I. Tomek, "Two modifications of CNN," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 11, pp. 769–772, Nov. 1976.
[52] S. N. Shirazi, A. Gouglidis, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis, and D. Hutchison, "Evaluation of anomaly detection techniques for SCADA communication resilience," in *Proc. IEEE Resilience Week (RWS)*, Aug. 2016, pp. 140–145.
[53] K. Demertzis, L. Iliadis, and V.-D. Anezakis, "MOLESTRA: A multi-task learning approach for real-time big data analytics," in *Proc. IEEE Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2018, pp. 1–8.
[54] A. Mansouri, B. Majidi, and A. Shamisa, "Metaheuristic neural networks for anomaly recognition in industrial sensor networks with packet latency and jitter for smart infrastructures," *Int. J. Comput. Appl.*, to be published.
[55] J. Brand and J. Balvanz, "Automation is a breeze with autoit," in *Proc. 33rd Annu. ACM SIGUCCS Conf. User Services*, 2005, pp. 12–15.
[56] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation," in *Proc. Australas. Joint Conf. Artif. Intell.* Berlin, Germany: Springer, 2006, pp. 1015–1021.
[57] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[58] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[59] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986.
[60] 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.
[61] H. Trevor, T. Robert, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
[62] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.



IZHAR AHMED KHAN received the B.Sc. degree from the University of Engineering and Technology, Pakistan, in 2008, and the master's degree in computer science from Mid Sweden University, Sweden, in 2011. He is currently pursuing the Ph.D. degree in computer science with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. His current research interests include machine learning, data mining, and anomaly detection systems.

DECHANG PI received the B.Eng. and M.Eng. degrees and the Ph.D. degree in mechatronic engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1994, 1997, and 2002, respectively, where he is currently a Professor and a Ph.D. Supervisor. He has authored over 100 journals and conference papers. His research interests include data mining and privacy, intelligent optimization methods, and security issues about moving objects. He presided over 30 research projects of the National Natural Science Foundation of China, the National 863 Program, the National Technical Foundation, the Civil Aerospace Foundation, and the Aviation Science Foundation.

ZAHEER ULLAH KHAN received the master's degree in computer science from the University of Peshawar, Pakistan, and the M.S. degree from Abdul Wali Khan University Mardan, Pakistan. He is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics, China. He has published many researcher papers in image processing and bioinformatics. His research interest includes predictive models for RNA/DNA sequences and generative models.

YASIR HUSSAIN received the B.Sc. degree from Bahauddin Zakariya University (BZU), Pakistan, in 2013, and the master's degree in computer science from the Virtual University of Pakistan, in 2015. He is currently pursuing the Ph.D. degree in computer science with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is particularly interested in machine learning, deep learning, data mining, recommender systems, and predictive models.

ASIF NAWAZ received the M.S. degree in software engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2010. He is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. His main interests include software engineering, machine learning, geographical information systems, data analysis, and decision support systems.