






Trustworthy and Reliable Deep-Learning-Based Cyberattack Detection in Industrial IoT

Fazlullah Khan , Senior Member, IEEE, Ryan Alturki , Senior Member, IEEE, Md Arafatur Rahman , Senior Member, IEEE, Spyridon Mastorakis , Member, IEEE, Imran Razzak , Senior Member, IEEE, and Syed Tauhidullah Shah

Abstract—A fundamental expectation of the stakeholders from the Industrial Internet of Things (IIoT) is its trustworthiness and sustainability to avoid the loss of human lives in performing a critical task. A trustworthy IIoT-enabled network encompasses fundamental security characteristics, such as trust, privacy, security, reliability, resilience, and safety. The traditional security mechanisms and procedures are insufficient to protect these networks owing to protocol differences, limited update options, and older adaptations of the security mechanisms. As a result, these networks require novel approaches to increase trust-level and enhance security and privacy mechanisms. Therefore, in this article, we propose a novel approach to improve the trustworthiness of IIoT-enabled networks. We propose an accurate and reliable supervisory control and data acquisition (SCADA) network-based cyberattack detection in these networks. The proposed scheme combines the deep-learning-based pyramidal recurrent units (PRU) and decision tree (DT) with SCADA-based IIoT networks. We also use an ensemble-learning method to detect cyberattacks in SCADA-based IIoT networks. The nonlinear learning ability of PRU and the ensemble DT address the sensitivity of

irrelevant features, allowing high detection rates. The proposed scheme is evaluated on 15 datasets generated from SCADA-based networks. The experimental results show that the proposed scheme outperforms traditional methods and machine learning-based detection approaches. The proposed scheme improves the security and associated measure of trustworthiness in IIoT-enabled networks.

Index Terms—Cybersecurity, data acquisition networks, deep learning, Industrial Internet of Things (IIoT), supervisory control, trustworthiness.

I. INTRODUCTION

THE Industrial Internet of Things (IIoT) is a pervasive network that connects a diverse set of smart appliances in the industrial environment to deliver various intelligent services. In IIoT networks, a significant amount of industrial control systems (ICSs) premised on supervisory control and data acquisition (SCADA) are linked to the corporate network through the Internet [1]. Typically, these SCADA-based IIoT networks consist of a large number of field devices [2], for instance, intelligent electronic devices, sensors, and actuators, connected to an enterprise network via heterogeneous communications [3]. This integration provides the industrial networks and systems with supervision and a lot of flexibility and agility [2]–[4], resulting in greater production and resource efficiency. On the other hand, this integration exposes SCADA-based IIoT networks to serious security threats and vulnerabilities, posing a significant danger to these networks and the trustworthiness of the systems [5]. The trustworthiness of an IIoT-enabled system ensures that it performs as expected while meeting a variety of security requirements, including trust, security, safety, reliability, resilience, and privacy [6]–[8]. Fig. 1 depicts the fundamental aspects of trustworthiness in an IIoT-enabled network. The basic goal of the IIoT-enabled system is to increase trustworthiness by safeguarding identities, data, and services, and therefore to secure SCADA-based IIoT networks from cybercriminals [8], [9].

Several protocol updates have been proposed to meet this purpose, including the distributed network protocol (DNP 3.0) [10]. However, it covers authentication and data integrity aspects only, leaving numerous holes for attackers to use known flaws like hash collision to carry out serious attacks [11]. Information Technology and Industrial Operational technology bodies build a typical risk management plan utilizing ISO 27005:2018 [10]

Manuscript received 7 December 2021; revised 17 March 2022, 19 May 2022, and 27 June 2022; accepted 2 July 2022. Date of publication 13 July 2022; date of current version 8 November 2022. This work was supported in part by the National Science Foundation under Awards CNS-2104700, CNS-2016714, and CBET-2124918, in part by the National Institutes of Health under Award NIGMS/P20GM109090, the University of Nebraska Collaboration Initiative, and in part by the Nebraska Tobacco Settlement Biomedical Research Development Funds. Paper no. TII-21-5431. (Corresponding authors: Fazlullah Khan; Ryan Alturki; Md Arafatur Rahman.)

Fazlullah Khan is with the Department of Computer Science, Abduli Wali Khan University Mardan, Mardan 23200, Pakistan (e-mail: fazlullah@awkum.edu.pk).

Ryan Alturki is with the Department of Information Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia (e-mail: rmturki@uqu.edu.sa).

Md Arafatur Rahman is with the School of Mathematics and Computer Science, University of Wolverhampton, WV1 1LY Wolverhampton, U.K. (e-mail: arafatur.rahman@ieee.org).

Spyridon Mastorakis is with the Department of Computer Science, University of Nebraska, Omaha, NE 68182 USA (e-mail: smastorakis@unomaha.edu).

Imran Razzak is with the School of Computer Science and Engineering, Faculty of Engineering, University of New South Wales Sydney, Sydney, NSW 2052, Australia (e-mail: imran.razzak@deakin.edu.au).

Syed Tauhidullah Shah is with the Department of Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: syed.tauhidullahshah@ucalgary.ca).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TII.2022.3190352>.

Digital Object Identifier 10.1109/TII.2022.3190352

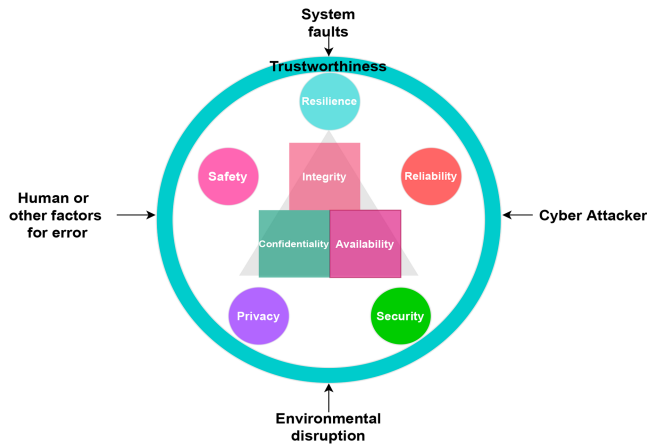


Fig. 1. Security and trustworthiness goals and CIA triad.

to recognize, rank, and implement alleviation techniques in automated or semiautomated enterprises. A comprehensive risk management plan and adequate preventive measures may not ensure absolute security against growing risks and attacks. This consequently offers a difficult research challenge for industrial and cybersecurity control researchers to 1) obtain the maximum degree of attack detection, 2) report malicious behavior as soon as it appears, and 3) isolate the afflicted subsystems as soon as possible. In recent years, there has been a surge toward the utility of artificial intelligence (AI) methods in evolving cybersecurity approaches, including attack prediction [12], privacy preservation [13], forensic exploration [14], and malware disclosure [15]. Deep learning (DL) is an AI approach that incorporates better learning models with considerable success in various disciplines [16]. However, designing a reliable and trustworthy AI, particularly a DL-based cyberattack detection model for the IIoT platforms, remains a research problem.

By considering the limitations of previous techniques, we employ network attributes of industrial protocols and propose a pyramidal recurrent unit (PRUs)- and decision tree (DT)-based ensemble detection mechanism. The proposed mechanism has the potential to detect cyberattacks in any extensive industrial network. The interoperability with other detection engines and expandability for a wider industrial network with multiple areas distinguishes the proposed mechanism from previous studies. The proposed detection method is disseminable across many IIoT domains. Furthermore, our model is straightforward to implement and deploy and can improve efficiency and accuracy while overcoming the shortcomings of previous efforts. The following capabilities can characterize the novelty and contribution of our article.

- 1) We propose a scalable and efficient DL- and DT-based ensemble cyber-attack detection framework to resolve trustworthiness issues in the SCADA-based IIoT networks.
- 2) We present an efficient probing approach by the SCADA-based network data to solve the protocol mismatch limitations of traditional security solutions for the IIoT platform.

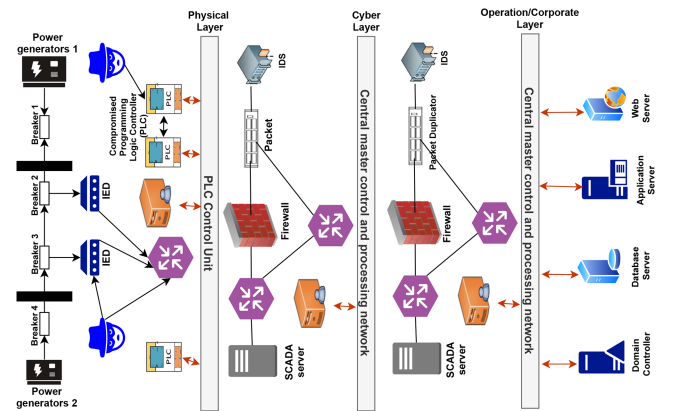


Fig. 2. SCADA-based industrial IoT network.

- 3) A statistical analytic approach for ensuring the trustworthiness and reliability of the proposed model for SCADA-based IIoT networks.

The rest of the article is organized as follows. In Section II, we have discussed the basics of problem formulation. In Section III, we have given details of our proposed work, followed by the results and discussion in Section IV. Finally, Section V concludes this article.

II. PRELIMINARIES AND METHODS

In this article, we follow the real-world settings [17] of cyberattacks on an ICS. Through these settings, we leverage the datasets from the power control system [18] for detecting industrial cyberattacks. Fig. 2 illustrates the overall architecture of a SCADA-based industrial control network. It is made up of various layers, including a processing and central master control layer, a physical layer, and a corporate layer, all of which are formed in a hierarchical order.

A. Datasets

The physical layer, as indicated in Fig. 2, contains various equipment such as breakers (BR1–BR4), intelligent electronic devices, power generators (G1, G2), and programmable logic controllers. The lowest physical layer collects sensor-based data and is used by the local control logic to make control decisions before transmitting it to the devices. They also get instructions from the top or master control/process layers, which also are responsible for managing and keeping track of the remote physical devices and local control layer devices. They are also equipped with intrusion detection systems (IDS). The corporate layer aids business operations and launches management declarations to the master control layer. In this article, we adopt the 15 benchmark datasets obtained from the SCADA power system¹ to identify and detect different kinds of attacks. The intrusion attacks on the SCADA system are detected using two separate classification events. The binary classification events, comprising 37 events, are divided into 28 attacks and 9 normal events. The other is the multiclass classification events, encompassing

¹<https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets>

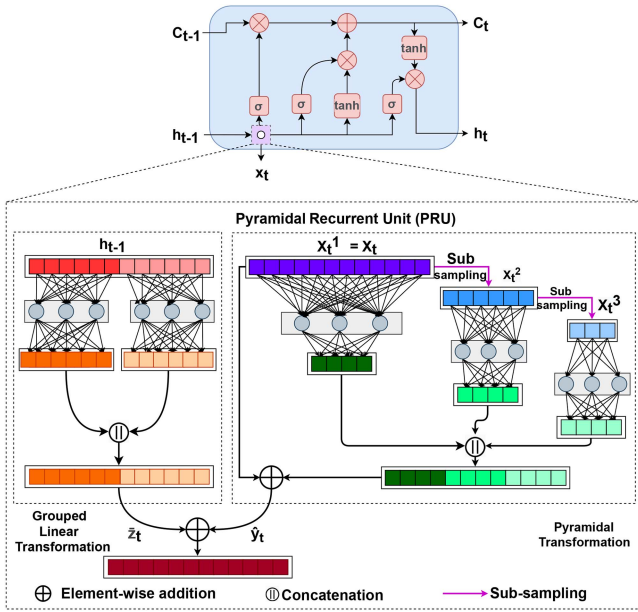


Fig. 3. PRU network.

37 different events, such as natural events, regular events, and attack events, each with its own set of class labels.

Each of the 15 datasets has thousands of distinct attacks. The datasets are randomly sampled at 1% to decrease the influence of a small sample size. Accordingly, there are 3711 attack-event samples, 1221 natural-event samples, and 294 no-event samples.

B. Problem Formulation

Assume a dataset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ with training examples, where x_i indicates a vector of real or discrete values. Further, these values represent the features of vector x_i , expressed as $\langle x_{i1}, x_{i2}, x_{i3}, \dots, x_{im} \rangle$. x_{ij} represents the j th feature of any given vector x_i . In contrast, the values of y_i are of dual nature. One type indicates binary classification, while the other consists of classes $\{1, \dots, K\}$, representing multiclassification. Different from that, the second type includes real values, representing regression. In a nutshell, given a training dataset D with E examples, the goal is to train a learning algorithm, which can produce a classifier output T . The classifier T indicates a hypothesis in the means of a true function, expressed as $f(x_i) = y_i$ that predicts new values for y_i every given value of x_i .

III. PROPOSED MODEL

A. PRU Models

Deep PRUs [19] are deep learning models used to manipulate sequential data. Fig. 3 provides an overview of the cell structure of a PRU cell. The PRU comprises several cells, each with three major layers: 1) the forget gate, 2) input gate, and 3) output gate. Also, PRU applies the pyramidal transformation to the input vector and uses a grouped linear transformation (GLT) to the context vector. Then, they combine them under the umbrella of PRU and feed it as input to the LSTM cell.

1) *Pyramidal Transformation (PR) for Input*: Instead of linearly transforming a given input vector x to an output vector y as $y = F_L(x) = \mathbf{W} \cdot x$, where $\mathbf{W} \in \mathbb{R}^{N \times M}$ is the weight matrix ($x \in \mathbb{R}^N$ to $y \in \mathbb{R}^M$), PR subsample it into K pyramidal levels to obtain various representations with different scales. The subsampling propagates K vectors as

$$\mathbf{x}^k \in \mathbb{R}^{\frac{N}{2^{k-1}}} \quad (1)$$

where 2^{k-1} denotes the sampling rate and $k = \{1, \dots, K\}$. For each $k = \{1, \dots, K\}$, the PR learns a scale-specific transformation as

$$\mathbf{W}^k \in \mathbb{R}^{\frac{N}{2^{k-1}} \times \frac{M}{K}}. \quad (2)$$

Then, PR concatenates the transformed subsamples to get the pyramidal output $\bar{\mathbf{y}} \in \mathbb{R}^M$ as

$$\bar{\mathbf{y}} = \mathcal{F}_P(\mathbf{x}) = [\mathbf{W}^1 \cdot \mathbf{x}^1, \dots, \mathbf{W}^K \cdot \mathbf{x}^K] \quad (3)$$

where $[\cdot, \cdot]$ denotes the concatenation operation. Given an input vector x , PR subsamples it using a kernel k with $2e + 1$ elements as

$$\mathbf{x}^k = \sum_{i=1}^{N/s} \sum_{j=-e}^e \mathbf{x}^{k-1}[si]\kappa[j] \quad (4)$$

where s denotes the stride operation while $k = \{2, \dots, K\}$.

2) *Grouped Linear Transformation*: GLT breaks down the traditional linear transformation through factoring in two parts. First, given the input vector $\mathbf{h} \in \mathbb{R}^N$, GLT split it into g smaller groups as

$$\mathbf{h} = \{\mathbf{h}^1, \dots, \mathbf{h}^g\} \quad \forall \mathbf{h}^i \in \mathbb{R}^{\frac{N}{g}}. \quad (5)$$

Then, through a linear transformation $\mathcal{F}_L : \mathbb{R}^{\frac{N}{g}} \rightarrow \mathbb{R}^{\frac{M}{g}}$, GLT transforms \mathbf{h}^i into $\mathbf{z}^i \in \mathbb{R}^{\frac{M}{g}}$ for each $i = \{1, \dots, g\}$. The final output vector is then formed by concatenating the resulting g output vectors \mathbf{z}^i as

$$\bar{\mathbf{z}} = \mathcal{F}_G(\mathbf{h}) = [\mathbf{W}^1 \cdot \mathbf{h}^1, \dots, \mathbf{W}^g \cdot \mathbf{h}^g]. \quad (6)$$

3) *Pyramidal Recurrent Unit*: PRU is created by extending the vanilla LSTM architecture using the pyramidal and the GLTs described above. At a given time t , PRU combines both input and context vectors through a transformation function using

$$\hat{\mathcal{G}}_v(\mathbf{x}_t, \mathbf{h}_{t-1}) = \hat{\mathcal{F}}_P(\mathbf{x}_t) + \mathcal{F}_G(\mathbf{h}_{t-1}) \quad (7)$$

where $v \in \{f, i, c, o\}$ indicates the forget, input, and output gates of the vanilla LSTM. $\hat{\mathcal{F}}_P(\cdot)$ denotes the pyramidal, whereas $\mathcal{F}_G(\cdot)$ represent the GLTs. The resultant $\hat{\mathcal{G}}_v$ is then fed to the vanilla LSTM architecture to model PRU. Specifically, a PRU cell takes $\mathbf{x}_t \in \mathbb{R}^N$, $\mathbf{h}_{t-1} \in \mathbb{R}^M$, and $\mathbf{c}_{t-1} \in \mathbb{R}^M$ at a given time t as input and generate the forget gate signal as

$$\mathbf{f}_t = \sigma(\hat{\mathcal{G}}_f(\mathbf{x}_t, \mathbf{h}_{t-1})) \quad (8)$$

The forget gate is in charge of removing each cell's prior information. The input and content gates, which update cell information is then calculated as

$$\mathbf{i}_t = \sigma(\hat{\mathcal{G}}_i(\mathbf{x}_t, \mathbf{h}_{t-1}))$$

TABLE I
PRUS SETTINGS FOR THE PROPOSED METHOD

PRU Model	Layer Size	Number of Layers
1	(100)	1
2	(200)	1
3	(100, 100)	2
4	(200, 100)	2
5	(100, 100, 100)	3
6	(100, 50, 20)	3

$$\hat{\mathbf{c}}_t = \tanh\left(\hat{\mathcal{G}}_c(\mathbf{x}_t, \mathbf{h}_{t-1})\right). \quad (9)$$

Similarly, the output gate is calculated as

$$\mathbf{o}_t = \sigma\left(\hat{\mathcal{G}}_o(\mathbf{x}_t, \mathbf{h}_{t-1})\right). \quad (10)$$

Context vector and cell state are then generated by combining the inputs with these gate signals as

$$\begin{aligned} \mathbf{c}_t &= \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \hat{\mathbf{c}}_t \\ \mathbf{h}_t &= \mathbf{o}_t \otimes \tanh(\mathbf{c}_t) \end{aligned} \quad (11)$$

where \otimes is the elementwise multiplication, σ represents the sigmoid while \tanh denotes the tangent activation function. In general, PRU cells are composed of only one layer. However, increasing the network depth enhances its efficiency and effectiveness when it comes to learning and recognizing complex sequential patterns [19]. Thus, we use a stack of PRUs with different configurations to better classify normal and attack events. The network size and number of layers are two of the most significant characteristics to consider while designing our PRUs. Table I lists the PRUs used in our method.

B. Ensemble of PRUs

To produce an aggregated outcome on the result of PRUs, we employ a DT unit. Suppose DT combines a set of different PRUs (denoted by L) over a subspace S for features $F_i \supseteq F$, indicated as $\{F_i(\cdot)\}_{j=1,\dots,S} \cdot \{y_i\}_{j=1,\dots,S}$ denotes the class label, which is acquired through distinct PRUs L . Each L can be independently classified for any given example $x \in F_i$ through its feature subspace F_i . The DT considers a set of confidence rates for each class in the dataset before deciding on the result. The DT module receives the input from L as

$$\begin{aligned} \text{Input of DT} &= \{L_{i,c} \text{ where } i \in \{\text{Number of } L\} \\ &\text{AND } c \in \text{Number of Classes}\} \end{aligned} \quad (12)$$

where $L_{i,c}$ indicates the confidence rate of i th trained model for class c . As an input, the DT takes these confidence rates and determines the association among the true label of network data and the L confidence rate in a hierarchical manner. Fig. 4 shows the schematic structure of the DT and its functions in our proposed scheme. Suppose a training set $D_M^{F_i}$, of M samples and F features, which each $i \in S$. In the same fashion, $D_N^{F_i}$ represents the test set with N samples and F features. DT

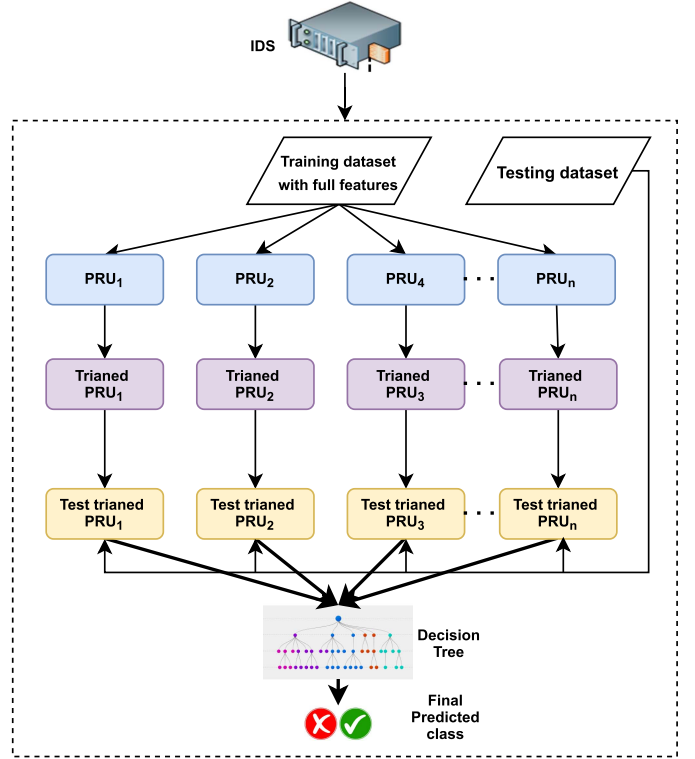


Fig. 4. Flowchart of the proposed method.

determines the PRU output space manifold and provides a model for classifying the output class label y_i . The proposed approach is efficient in training and testing, requires little memory, and is appropriate and scalable for intrusion detection in SCADA IIoT because of its ability to eliminate irrelevant features.

Theorem 1: Our method is trusted to detect SCADA-based IIoT cyberattacks through the ensemble of PRUs and DT.

Proof: Suppose S represents a group of training instances and a deep-learning model D can build a learner L . L can be considered a hypothesis around a true function f , which accepts an instance x and assigns a label y to it. The proposed model produces a collection of learners/hypotheses (L) and explores a space H for optimal hypotheses. The proposed learning process can discover various distinct hypotheses in H , where each provides identical or varying accuracy outcomes on training examples of distinct random feature sets. The proposed approach reduces the likelihood of selecting incorrect learners by generating a collection of accurate learners and combining them through a DT. Combining precise hypotheses can better statistically approximate the function f . Hence, the proposed model is trusted to identify intrusion attacks in SCADA-based IIoT networks.

IV. EVALUATIONS AND FINDINGS

We conducted a wide range of experiments with the benchmark datasets discussed in Section II-A. We implemented our proposed model using Python 3.7 and the popular deep learning

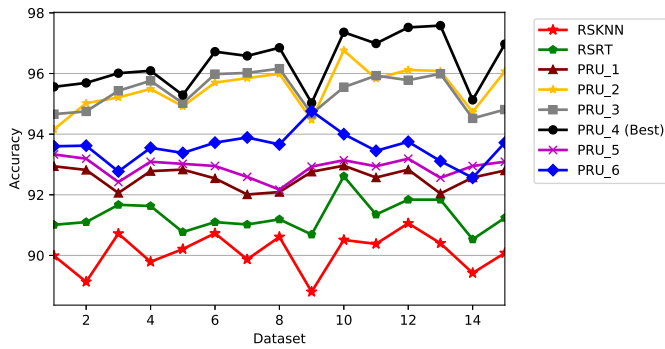


Fig. 5. Performance analysis of our proposed scheme for binary classification in terms of accuracy.

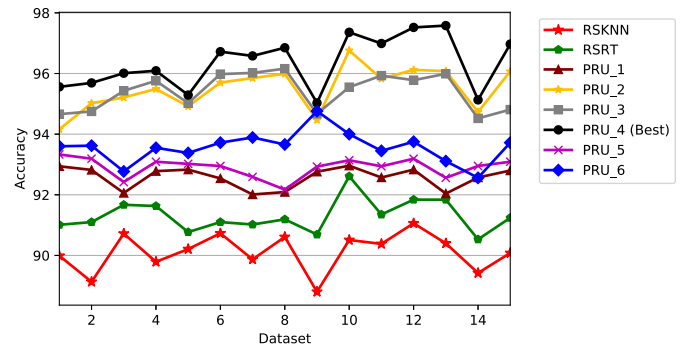


Fig. 6. Performance analysis of the proposed model for multiclassification in terms of accuracy.

framework PyTorch.² We ran all experiments on the NVIDIA GEFORCE GTX 1080 GPU for our proposed models and alternative baselines. We trained six distinct PRUs, each with a varied structure. We employ Adam [20], which delivers faster convergence than the SGD and avoids the challenge of adjusting the learning rate [16]. We selected 256 as the batch size, 200 as the epoch, 0.001 as the learning rate and determined the hyperparameters through experiments. We also employed a 10-fold cross-validation approach [21] for both training and testing, which breaks a dataset randomly into ten segments and takes one segment for testing and the remaining nine for training. However, we divided the dataset into three parts at random and utilized eight segments for training, one segment for testing, and one segment for validation. We use the following metrics and detection time to measure the effectiveness of our model:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (13)$$

$$\text{False positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (14)$$

where FP, FN, TP, and TN represent the false positive, false negative, true positive, and true negative, respectively. Accuracy measures the samples accurately detected by a classifier divided by total samples.

A. Results

Figs. 5–8 demonstrate the experimental outcomes of the baselines and our proposed model. Fig. 5 shows the accuracy, whereas Fig. 6 describes the false-positive rate for detecting both normal and abnormal events. In the same fashion, Fig. 7 shows the accuracy, whereas Fig. 8 illustrates the false-positive rate for classifying the normal and various attacks in traffic events.

B. Comparison With Benchmark Methods

We compare our method with RSKNN [10] and RSRT [14] models in terms of accuracy and computational time to illustrate its superior performance. We follow the same structure as reported in their work for a fair comparison. We compare the accuracy results for all of the 15 datasets, and in terms

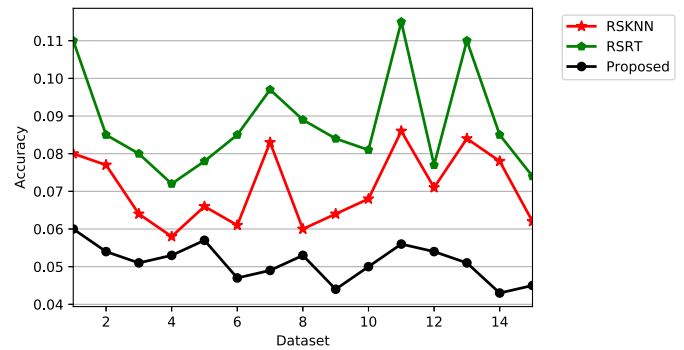


Fig. 7. Performance analysis of the proposed model for binary classification in terms of false-positive rates.

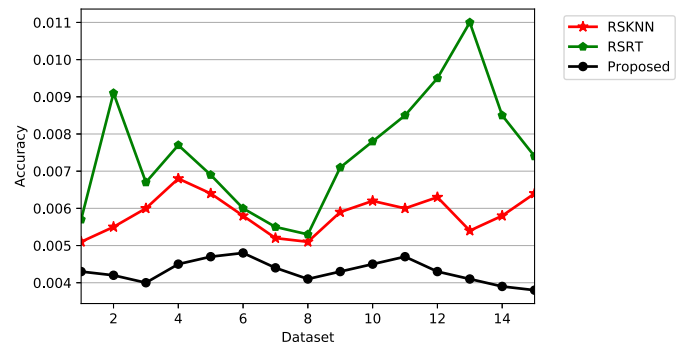


Fig. 8. Performance analysis of the proposed model for multiclassification in terms of false-positive rates.

of computational time costs, we only consider dataset 9. In addition, we also use a statistical analysis test to assess the statistical variations in accuracy results.

1) *Comparison of Accuracy Results:* We conducted experiments with each model on all 15 datasets. We conducted experiments with each model on all 15 datasets. Figs. 5 and 6 and Table II illustrate the accuracy results. As can be seen in both Figs. 5 and 6, PRU model 4 is the best model. Thus, for clarity, we only showcase the results of PRU 4. Table II shows how well our model detects both normal and abnormal events when compared to other baselines. Similarly, our model also outperforms the baseline models in the multiclassification attack settings. Also,

²<https://pytorch.org/>

TABLE II

COMPARISON RESULT OF OUR METHOD AND OTHER BASELINE METHODS IN TERMS OF ACCURACY FOR BINARY AND MULTICLASSIFICATION

Datasets	Binary classification			Multiclassification		
	RSKNN	RSRT	Proposed	RSKNN	RSRT	Proposed
1	95.87%	96.71%	98.89%	89.99%	91.01%	95.56%
2	95.12%	95.60%	98.21%	89.13%	91.10%	95.69%
3	95.91%	96.06%	98.49%	90.72%	91.67%	96.01%
4	95.17%	96.34%	98.64%	89.79%	91.63%	96.09%
5	96.55%	96.49%	98.75%	90.21%	90.77%	95.29%
6	95.73%	96.05%	98.46%	90.73%	91.10%	96.72%
7	95.26%	96.12%	98.53%	89.87%	91.02%	96.58%
8	95.59%	96.38%	98.61%	90.61%	91.19%	96.85%
9	95.16%	95.18%	98.57%	88.80%	90.69%	95.03%
10	96.13%	96.66%	98.82%	90.51%	92.61%	97.36%
11	95.77%	96.19%	98.59%	90.38%	91.35%	96.99%
12	95.94%	96.19%	98.62%	91.06%	91.84%	97.52%
13	96.73%	96.77%	99.03%	90.40%	91.84%	97.58%
14	95.48%	96.16%	98.52%	89.42%	90.53%	95.13%
15	95.10%	96.05%	98.45%	90.08%	91.24%	96.97%

it can be seen that our proposed model outperforms both RSRT and RSKNN for detecting both normal and abnormal events for binary and multiclass classification.

2) *Statistical Analysis of Accuracy Results:* We used the nonparametric Mann–Whitney *T*-test for a statistical analysis and looked at the implications of the accuracy results for RSRT and our proposed method. The nonparametric Mann–Whitney *T*-test is considered resilient against outliers, better for small sample sizes, and is independent of distributional assumptions [22]. The Mann–Whitney *T*-test compares the observations of two groups and uses their size for ranking them, and is computed as

$$T = R_1 - \frac{n_1(n_1 + 1)}{2} + R_2 - \frac{n_2(n_2 + 1)}{2} \quad (15)$$

where *R*1 and *R*2 imply the sum of rank in 1 and 2, respectively, and *n*1 and *n*1 represent sample sizes 1 and 2, respectively, by utilizing the sum of ranks and mean rank for every single group. The best group is ranked first, whereas the second-best is ranked second in this situation. The statistical analysis’s testing question can be stated as follows “*Is there a statistically significant difference between the accuracy results obtained by RSRT and the proposed models?*” We begin by presenting the hypothesis and classifying the assert in the following manner

- 1) *Alternate Hypothesis:* There are statistical variations for classifying normal and abnormal events (binary classification) or various kinds of attacks in traffic events (multiclassification) in the accuracy outcomes of the two models.
- 2) *Null Hypothesis:* There are no statistical variations for classifying normal and abnormal events (binary classification) or various kinds of attacks in traffic events (multiclassification) in the accuracy outcomes of the two models.

Fig. 9 depicts the standard error of standard deviation for classifying normal and abnormal attacks, whereas Fig. 10 illustrates the standard error of standard deviation in the multiclassification settings. We used the statistical SPSS tool to conduct the test. For binary classification, Tables III– V summarize the rank, test

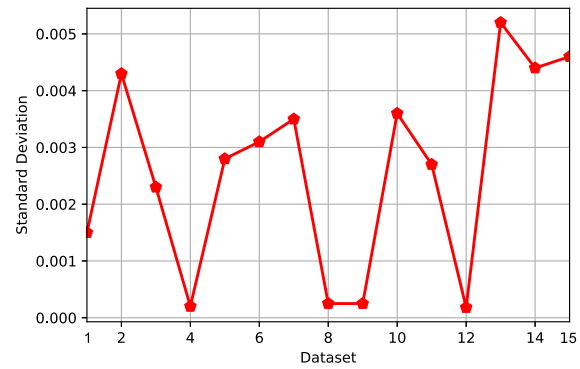


Fig. 9. Standard deviation of the proposed method for binary classification in terms of accuracy.

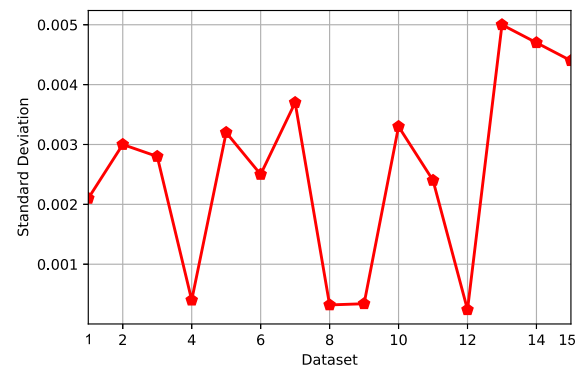


Fig. 10. Standard deviation of the proposed method for multiclassification in terms of accuracy.

TABLE III

DESCRIPTIVE STATISTICS OF OUR METHOD FOR BINARY AND MULTICLASSIFICATION IN TERMS OF ACCURACY RESULTS

<i>N</i>	Mode	Std. deviation	Mean	Max	Min
30	Binary classification	0.492	96.68	97.31	96.77
30	Multiclassification	0.73	92.22	93.96	90.51

TABLE IV

COMPARISON BETWEEN RSRT AND OUR PROPOSED METHOD FOR BINARY AND MULTICLASSIFICATION IN TERMS OF RANKS

Mode	<i>N</i>	Model	Sum of ranks	Mean rank
Binary Classification	15	RSRT	300.00	20.00
		PROPOSED	450.00	30.00
Multiclassification	15	RSRT	130.00	8.67
		PROPOSED	495.00	33.01

TABLE V

TEST STATISTICS OF OUR METHOD FOR BINARY AND MULTICLASSIFICATION IN TERMS OF ACCURACY RESULTS

Statistic	Binary classification	Multiclassification
Mann–Whitney U	48.25	15.00
Wilcoxon W	180.00	145.00
Z	−2.98	−3.40
Asymp. Sig. (2-tailed <i>p</i> -value)	0.0048	0.002
Exact Sig. [2*(1-tailed Sig.)]	0.0046	0.002

TABLE VI

COMPARATIVE RESULTS OF PROPOSED MODEL WITH RSRT IN TERMS OF AVERAGE TIME (SECONDS) AND TRAINING AND TESTING COST

Model	Task	Training (seconds)	Testing
RSRT	Binary class	0.22	0.03
	Multiclass	0.36	0.05
Proposed	Binary class	0.35	0.01
	Multiclass	0.47	0.02

statistics, and descriptive statistics in terms of accuracy results. The two-tailed p -value, as indicated in Table V, is below 0.05. Thus, with a confidence level of 95%, we refuse the null and adopt the alternative hypothesis. Consequently, we infer that the accuracy outcomes of the two models differ statistically. From Table IV, we may further deduce that these variations are for our proposed method, indicating its superiority over the RSRT, based on the sum of ranks and mean rank results. Likewise, we establish the following hypothesis for classifying normal and various other attacks in traffic events.

C. Comparison of Computational Time Costs:

We used dataset 9, which comprises 5340 different instances, to compare the time costs for both our proposed and RSRT methods. After preprocessing, the dataset contains 3738 and 1602 instances for both training and testing, respectively. We examine both binary and multiclass configurations to determine the time cost. On the specified dataset, for both training and testing, Table VI shows the average time cost. We can see that our proposed method requires somewhat more time to train than the RSRT. This is due to the fact that the RSRT model does not utilize a deep learning method. On the other hand, our proposed method takes substantially less time for testing than the RSRT model, making it more effective in real-world scenarios.

D. Reliability and Trustworthiness

To examine the reliability, assume that our method comprises 10 PRUs or learners. Because of the heterogeneous nature of ensemble learning, the errors that occur in these PRUs are uncorrelated. If some learners are inaccurate, the remaining learners may be accurate, enabling our method to properly categorize intrusion attacks in SCADA-based IIoT networks. Fig. 11 shows a simulated probability of error for 10 different learners. We can see that each learner has an error of less than or equal to 0.14, and 7 of them have an error of less than 0.09, making our method good enough to detect attacks in SCADA-based IIoT networks. We carry out experiments with dataset 1 to verify the trustworthiness of our proposed model by classifying attacks with various numbers of learners. Fig. 12 shows the accuracy results of the proposed model utilizing an ensemble of 10 base learners corresponding to a single learner. Also, we can observe how the accuracy of our proposed method increases by combining multiple learners.

We can also reveal the trustworthiness of our method by offering a mapping of the trusted computing base (TCB) model to the defense-in-depth model. This mapping can help explain

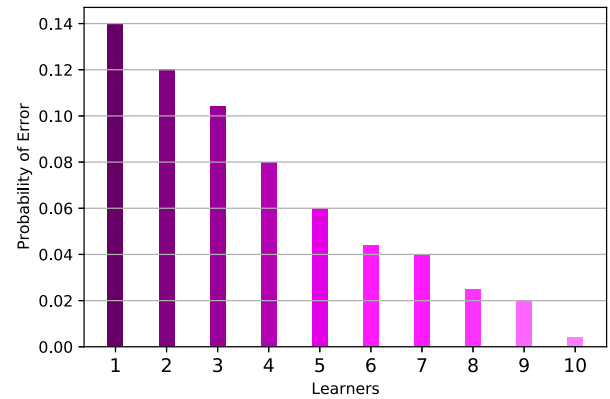


Fig. 11. Error probability for various learners.

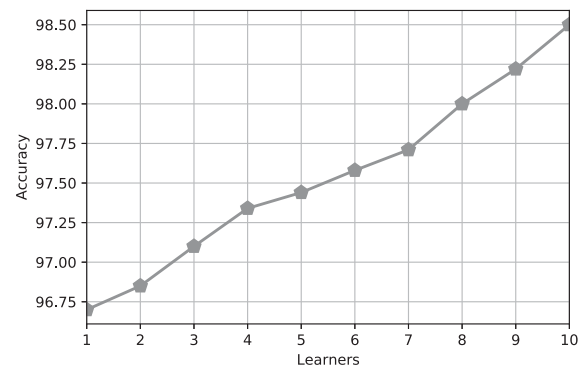


Fig. 12. Accuracy results of the proposed method for various learners.

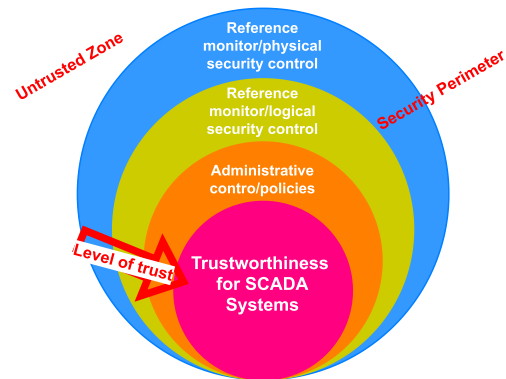


Fig. 13. TCB security paradigm employing a defense-in-depth method to ensure trustworthiness.

how confidentiality–integrity–availability (CIA) is sustained. Fig. 13 illustrates the TCB security paradigm, which is embedded in our proposed SCADA-based model. The elements of the trusted zone inside the security outline include security control, hardware and software, and policies, which are coupled to guarantee the maintenance of the CIA triad and the total security system adds to trustworthiness. The TCB/SCADA reference monitor/physical security control paradigm prevents and detects unwanted illegal actions to resources within the trusted zone’s boundary. This layer often includes automated physical access control systems (PACS), for instance, mantraps, CCTV cameras, and motion detectors. On the other hand, SCADA systems

and associated subsystems are typically positioned in remote locations, where PACS deployment is challenging. Hence, in this case, a defense-in-depth approach must be supplemented with extra measures, for example, establishing antimalware resources or IDSs in the logical control.

They are incompatible with the SCADA settings since they are dependent on application program interfaces or protocols. As a result, these classical detective or preventative security controls fail against blocking unauthorized access. Hence, accurate and reliable security control must be established to ensure a defense-in-depth approach and improve the trustworthiness of the SCADA system. We solved these shortcomings in our proposed model, formed a reliable cyber-attack detection method, and verified it with massive SCADA network traffic with various attacks targeting several vulnerabilities of SCADA components and the overall system.

V. CONCLUSION

The ability to protect SCADA-based IIoT networks against cyberattacks increases their trustworthiness. The existing security methods along with machine learning algorithms were inefficient and inaccurate for protecting IIoT networks. In this article, we proposed a cyberattacks detection mechanism using enhanced deep and ensemble learning in a SCADA-based IIoT network. The proposed mechanism is reliable and accurate because an ensemble detection model was built using a combination of the PRU and the DT. The proposed method was evaluated across 15 datasets generated from a SCADA-based network, and a considerable increase in terms of classification accuracy was obtained. Compared to state-of-the-art techniques, the obtained outcomes of our method exhibited a good balance between reliability, trustworthiness, classification accuracy, and model complexity, resulting in improved performance.

In the future, we will employ more powerful deep learning models to further improve trustworthiness by detecting cyberattacks accurately. In addition, we will try to formulate and assess its performance in real-world scenarios. Also, we will work on the selection of optimal features in scenarios when the features are not sufficient.

REFERENCES

- [1] Y. Luo, Y. Duan, W. Li, P. Pace, and G. Fortino, "A novel mobile and hierarchical data transmission architecture for smart factories," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3534–3546, Aug. 2018.
- [2] C. Gavriluta, C. Boudinet, F. Kupzog, A. Gomez-Exposito, and R. Caire, "Cyber-physical framework for emulating distributed control systems in smart grids," *Int. J. Elect. Power Energy Syst.*, vol. 114, 2020, Art. no. 105375.
- [3] M. S. Mahmoud, M. M. Hamdan, and U. A. Baroudi, "Modeling and control of cyber-physical systems subject to cyber attacks: A survey of recent advances and challenges," *Neurocomputing*, vol. 338, pp. 101–115, 2019.
- [4] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, and M. Xie, "A novel trust mechanism based on fog computing in sensor–cloud system," *Future Gener. Comput. Syst.*, vol. 109, pp. 573–582, 2020.
- [5] K. Guo et al., "MDMaaS: Medical-assisted diagnosis model as a service with artificial intelligence and trust," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2102–2114, Mar. 2020.
- [6] M. Al-Hawawreh and E. Sitnikova, "Developing a security testbed for industrial Internet of Things," *IEEE Internet of Things J.*, vol. 8, no. 7, pp. 5558–5573, Apr. 2021.

- [7] M. A. Shahriar et al., "Modelling attacks in blockchain systems using petri nets," in *Proc. IEEE 19th Int. Conf. Trust Secur. Privacy Comput. Commun.*, 2020, pp. 1069–1078.
- [8] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakraborty, and M. Ryan, "Deep-IFS: Intrusion detection approach for IIoT traffic in fog environment," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7704–7715, Nov. 2021.
- [9] S. Huda, J. Abawajy, B. Al-Rubaie, L. Pan, and M. M. Hassan, "Automatic extraction and integration of behavioural indicators of malware for protection of cyber–physical networks," *Future Gener. Comput. Syst.*, vol. 101, pp. 1247–1258, 2019.
- [10] Information Technology-Security Techniques-Information Security Risk Management, ISO/IEC 27005:2018, 2018.
- [11] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6182–6192, Sep. 2020.
- [12] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5244–5253, Aug. 2020.
- [13] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.
- [14] M. M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the trustworthiness in the industrial IoT networks through a reliable cyberattack detection model," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6154–6162, Sep. 2020.
- [15] A. N. Jahromi et al., "An improved two-hidden-layer extreme learning machine for malware hunting," *Comput. Secur.*, vol. 89, 2020, Art. no. 101655.
- [16] S. T. U. Shah, J. Li, Z. Guo, G. Li, and Q. Zhou, "DDFL: A deep dual function learning-based model for recommender systems," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2020, pp. 590–606.
- [17] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst.*, 2014, pp. 1–8.
- [18] A. Derhab et al., "Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security," *Sensors*, vol. 19, no. 14, 2019, Art. no. 3119.
- [19] S. Mehta, R. Koncel-Kedziorski, M. Rastegari, and H. Hajishirzi, "Pyramidal recurrent unit for language modeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4620–4630.
- [20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [21] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia Database Syst.*, vol. 5, pp. 532–538, 2009.
- [22] G. W. Zeoli and T. S. Fong, "Performance of a two-sample Mann-Whitney nonparametric detector in a radar application," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-7, no. 5, pp. 951–959, Sep. 1971.



Fazlullah Khan (Senior Member, IEEE) received the Ph.D. degree in computer science from Abdul Wali Khan University Mardan, Mardan, Pakistan, in 2020.

His research has been published in the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE INTERNET OF THINGS JOURNAL, IEEE ACCESS, Elsevier *Computer Networks*, Elsevier *Future Generation Computer Systems*, Elsevier *Journal of Network and Computer Applications*, Elsevier *Computers & Electrical Engineering*, Springer *Mobile Networks & Applications* (MoNET), and Springer *Neural Computing and Applications* (NCAA). His research interests include security and privacy, Internet of Things, machine learning, artificial intelligence, security and privacy issues in the Internet of Vehicles, SDN, fog/cloud computing, and big data analytics.

Dr. Khan was the Guest Editor of the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, Elsevier *Digital Communications and Networks*, Springer *Multimedia Technology and Applications*, Springer MoNET, and Springer NCAA. He has served more than 10 conferences in leadership capacities including General Chair, General Co-Chair, Program Co-Chair, Track Chair, and Session Chair.



Ryan Alturki (Senior Member, IEEE) received the Ph.D. degree in computer systems from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently an Assistant Professor with the Department of Information Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia. He authored or coauthored several publications in high-ranked international journals, conferences, and chapters of books. His research interests include eHealth, mobile technologies, the Internet of Things, artificial intelligence, cloud computing, and cybersecurity.



Md Arafatur Rahman (Senior Member, IEEE) received the Ph.D. degree in ETE from the University of Naples Federico II, Naples, Italy, in 2013.

He is currently a Senior Lecturer with the School of Engineering, Computing & Mathematical Sciences, University of Wolverhampton, Wolverhampton, U.K. His research interests include IoT, wireless communication networks, cognitive radio networks, 5G, vehicular communication, big data, cloud-fog-edge computing, machine learning, and security.



Spyridon Mastorakis (Member, IEEE) received the five-year diploma (equivalent to M.Eng.) in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2014, and the M.S. and the Ph.D. degrees in computer science from the University of California, Los Angeles, Los Angeles, CA, USA, in 2017 and 2019, respectively.

He is currently an Assistant Professor in computer science with the University of Nebraska Omaha, Omaha, Nebraska. His research interests include network systems and protocols, Internet architectures, IoT and edge computing, and security.



Imran Razzak (Senior Member, IEEE) received the Ph.D. degree from University of Technology Sydney Australian, Australia, in 2019. He is currently a Senior Lecturer in human-centered AI and machine learning with the School of Computer Science and Engineering, University of New South Wales, Sydney, Sydney, NSW, Australia. He is also an Associate Editors/Guest Editor of several journals such as IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, etc.

His research interests include machine learning and NLP with its application to a broad range of topics, particularly deep learning, big data analytics, healthcare, and cyber security, mainly focusing on the healthcare sector, and he is passionate about making the healthcare industry a better place through emerging technologies.



Syed Tauhidullah Shah received the B.S. degree in computer science from Abdul Wali Khan University Mardan, Mardan, Pakistan, and the M.S. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2017 and 2020. He is currently working toward the Ph.D. degree in machine learning and natural language processing for requirement elicitation with the Department of Software Engineering, University of Calgary, Calgary, AB, Canada.

His research interests include deep learning, recommender systems, Internet of Things, and natural language processing.