Cluster Computing (2022) 25:561–578
https://doi.org/10.1007/s10586-021-03426-w

A stacked deep learning approach to cyber-attacks detection in industrial systems: application to power system and gas pipeline systems

Wu Wang[1] · Fouzi Harrou[2] · Benamar Bouyeddou[3] · Sidi-Mohammed Senouci[4] · Ying Sun[2]

Received: 5 March 2021 / Revised: 9 September 2021 / Accepted: 17 September 2021 / Published online: 5 October 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
Presently, Supervisory Control and Data Acquisition (SCADA) systems are broadly adopted in remote monitoring large-scale production systems and modern power grids. However, SCADA systems are continuously exposed to various heterogeneous cyberattacks, making the detection task using the conventional intrusion detection systems (IDSs) very challenging. Furthermore, conventional security solutions, such as firewalls, and antivirus software, are not appropriate for fully protecting SCADA systems because they have distinct specifications. Thus, accurately detecting cyber-attacks in critical SCADA systems is undoubtedly indispensable to enhance their resilience, ensure safe operations, and avoid costly maintenance. The overarching goal of this paper is to detect malicious intrusions that already detoured traditional IDS and firewalls. In this paper, a stacked deep learning method is introduced to identify malicious attacks targeting SCADA systems. Specifically, we investigate the feasibility of a deep learning approach for intrusion detection in SCADA systems. Real data sets from two laboratory-scale SCADA systems, a two-line three-bus power transmission system and a gas pipeline are used to evaluate the proposed method's performance. The results of this investigation show the satisfying detection performance of the proposed stacked deep learning approach. This study also showed that the proposed approach outperformed the standalone deep learning models and the state-of-the-art algorithms, including Nearest neighbor, Random forests, Naive Bayes, Adaboost, Support Vector Machine, and oneR. Besides detecting the malicious attacks, we also investigate the feature importance of the cyber-attacks detection process using the Random Forest procedure, which helps design more parsimonious models.

Keywords Stacked deep learning · SCADA system · Intrusion detection · Critical infrastructure protection · Cyber-attacks

1 Introduction

Today, Supervisory Control and Data Acquisition (SCADA) systems have been widely deployed for online operation and monitoring of most critical infrastructures. Abnormal operating conditions can be sensed from a remote location by a SCADA system. Accordingly, the response time for correcting an abnormal condition is decreased, and the appropriate real-time controls can be applied. These systems are already on board for a wide range of applications, including electric power (generation, transmission, and distribution), water, transportation, telecommunication, pharmaceutical, and manufacturing industries, and are commonly involved in the constitutions of vital enterprises such as pipelines, manufacturing plants and building climate control [1]. Typical SCADA systems include components like computer workstations, Human Machine Interface (HMI), Programmable Logic Controllers (PLCs), Remote Terminal Units (RTU), sensors, and actuators [2]. Historically, early (i.e., monolithic) SCADA systems had private and dedicated networks. They were designed to run as isolated and independent systems without connecting to other systems [3]. The current SCADA systems are generally distributed, networked, and communicated over wide area network (WAN) systems, such as public IP networks (e.g., internet) and wireless cellular networks (e.g., 3G and 4G) using the Modicom Communication Bus (Modbus) TCP (Transmission Control

Extended author information available on the last page of the article

Protocol), Distributed Network Protocol (DNP3), and IEC 60870-5-104 protocols [4] principally. Therefore, the SCADA systems' infrastructure cost is significantly reduced by adopting the internet of things (IoT) technology, which involves the commercially available cloud computing services. For instance, many studies on embedded SCADA systems have been conducted [3, 5].

Although technology is advancing, SCADA systems are becoming increasingly vulnerable to various cyber-attacks. Different types of attacks, such as denial-of-service (DOS), data modifying, and packet injection, can seriously affect the SCADA system's components. For instance, several cyber-attacks targeting SCADA systems have been reported, including the distributed denial of service (DDOS) attacks that shut down Alabamas's Browns Ferry nuclear plant, attacks against water purification systems in Harrisburg, Pennsylvania, and malware attacks that paralyzed the train signal system at the CSX corporation [6]. Remarkably, the most prominent attack has targeted Stuxnet using a virus that infected hundreds of thousands of industrial controllers around the world [7]. Besides, the Ukrainian power system had seen colossal attacks conducted with the BlackEnergy malware, which leads to current interruption for more than 10,000 homes and facilities over several days [8]. Miller et al. [9] presented meticulous documentation that emphasized cyberattacks in SCADA critical infrastructures. Indeed, the existence of this bulk of cyber-attacks offers sufficient evidence to unearth the harshness of the security concerns in the SCADA systems and demands immediate attention from the cybersecurity and forensic science community to tackle such challenges.

This paper introduces a stacked deep learning-driven anomaly detection technique to detect and identify cyber-attacks in SCADA systems. The introduced stacked deep learning model consists of five deep learning models for deeply learning the malicious activities' features and discriminating them from nominal features. This choice is mainly motivated by deep learning models' high efficiency in discovering layer-by-layer complex nonlinearity in multivariate data, making them efficient to separate malicious from non-malicious and natural disturbances in SCADA systems. In other words, deep learning methods are efficient and flexible tools for modeling implicit relationships between process variables and enabling the recognition of complicated patterns. Note that the idea behind the proposed stacked deep learning model is inspired by the ensemble learning methods, such as Adaboost and random forest. It has been proved that Adaboost [10] and random forest [11] improves the classification accuracy of individual trees. Thus, classification accuracy can be improved using ensemble models combining multiple learners versus single learners. Importantly, exploiting the stacked deep learning model is advantageous in the sense that it has the potential to improve the detection of cyberattacks in SCADA systems. Real data sets from two laboratory-scale SCADA systems, a two-line three-bus power transmission system and a gas pipeline, are used to evaluate the proposed method's performance. Specifically, we investigate the capability of stacked and standalone deep learning-driven techniques in detecting different attacks in a modern power system and analyzing the remote terminal unit (RTU) serial communications in a gas pipeline system. These datasets are made publicly available by the Mississippi State University's Critical Infrastructure Protection Center [12]. To compare and check the models' performance detection quality, we use four common performance metrics: accuracy, precision, recall, and F-measure. To verify the proposed scheme's efficiency, we compare the obtained results to state-of-the-art approaches, including Nearest neighbor, Random forests, Naive Bayes, Adaboost, Support Vector Machine, Decision tree, One R, and J48. The results reveal promising performances of the stacked deep learning-driven scheme in detecting cyber-attacks in SCADA systems. Also, we investigated the variable importance by the Random Forest algorithm; more parsimonious models can be constructed based on important variables.

Section 2 highlights literature reviews on the related works and Sect. 3 introduces the proposed deep learning-based malicious attack detector. Sections 4 and 5 assess the proposed method and compare its performance using datasets for the Power system testbed and gas pipeline system, respectively. Finally, Sect. 6 concludes this study and sheds light on potential future research lines.

2 Related works

Modern SCADA systems are vulnerable to cyber-attacks because of their common usage of conventional communications protocols and extended mutual connection with corporate networks and the Internet [13]. Accordingly, cybersecurity has gained considerable attention across many research communities, and various intrusion detection systems (IDS) to detect attacks in SCADA systems and to protect their shared data have been developed [14]. In [15], a method for assessing oil and gas SCADA security has been introduced using causality analysis. This approach adopted the causality analysis evaluation method of fuzzy Mamdani reasoning for assessing factors neurons in the introduced method. It has been shown that the causality analysis-driven approach offers good ability in assessing SCADA information security. The authors in [16] introduced an intrusion detector, which is based on the concept of Context Awareness and Anomaly Behavior Analysis

(ABA), to identify and classify different types of attacks in Building Automation and Control network (BACnet). The performance of this detector is verified based on data from the Smart Building testbed designed at the University of Arizona Center for Cloud and Autonomic Computing. Results show the good detection ability of this detector to identify BACnet attacks. In [17], Linda et al. proposed an anomaly detection scheme based on neural networks, and they exploited the SCADA network and system information to handle the problem of bad packets. However, this solution can only handle external attacks; the internal attackers can still introduce malicious command packets to infect central equipment. In [18], Sayegh et al. used the network packets correlation and system behavior to detect injection attacks. Nevertheless, this proposed rule-based IDS does not detect novel or unidentified intrusions passed through traditional IDS in open access networks. The method in [19] first learns a whitelist of allowed communication flows based on the network traffic training set. Then, any non-whitelisted connections are flagged out as an alarm. Data mining and machine learning methods were recently largely exploited for designing effective IDS because of their flexibility to handle large-sized datasets, which is difficult to implement by a human agent manually [20]. However, SCADA data scarcity is considered one of the main issues for establishing efficient intrusion detection solutions for SCADA systems. For instance, in [21], SVM is considered to detect and classify malicious and infected data and potential cyber-attacks in future traffic streams. However, one of the primary problems of supervised techniques is that the learning process demands a large size of training observations. In [22], the fuzzy c-means-based method has been adopted to develop a network IDS, data are classified as normal and abnormal. Fovino et al. [23] suggested an IDS using system monitoring state evolution. However, the detection rules require prior knowledge of the physical process and its different critical states. In [24, 25], authors proposed an IDS based on moving average and Kalman filter. Unfortunately, such methods are appropriate when variables are linearly related through a predefined model-based system. Bayesian framework [26], graph theory [27] and equivalent line impedances [28] were used to deal with the false data injection attacks. The designed methods assume perfect protection of some PMU, even if such condition is typically invalid in practice. In [29], a method to detect suspicious activities in a power system has been proposed. This method employs the correlation coefficient-based procedure to select relevant data portions and applies Expectation Maximisation (EM) clustering algorithms to identify intrusive events in the inspected SCADA system. The detection efficiency of this approach has been verified using power system datasets for multiclass attacks and showed superior performance compared to Random Forests Nearest Neighbour and Naive Bayes classifiers.

Effective and efficient detection of malicious attacks in modern industrial systems is indispensable to maintain the desired specifications and continuous operation. It is worth pointing out that the cyber-attacks generally could have similar effects as some typical events, making the separation between malicious and natural events in complex systems challenging and infeasible for a human. Traditional machine learning methods for intrusion detection are generally not suited to reveal implicit and relevant information. Also, the methods mentioned above could not mine huge data [30, 31]. In recent years, deep learning has emerged as a promising tool in modeling time-dependent in time-series data and anomaly detection and used in a wide range of applications, including intelligent transportation systems [32], and health informatics [31, 33]. However, not much research is done to design the SCADA specific IDS based on deep learning. For instance, In [34], a deep learning scheme based on the Conditional Deep Belief Network (CDBN) is introduced to detect false data injection (FDI) that threatens the data integrity of SCADA systems. Specifically, a deep learning scheme is employed for recognizing the behavior features of FDI attacks based on historical data and then used the learned features for detecting the FDI attacks. It showed good detection performance compared to ANN-based and SVM-based methods when applied to four simulated scenarios using the IEEE 118-bus power test system and IEEE 300-bus system. The method in [35] used a convolutional neural network (CNN) model to characterize temporal patterns of SCADA traffic and then uncover network attacks. Based on the University of Arkansas's National Center for reliable electric power transmission testbed dataset, the results demonstrated that this method is effective against various anomalies. The authors in [36] proposed an Intrusion Detection and Prevention System (IDPS) called DIDEROT based on machine learning to detect and prevent malicious attacks and anomalies targeting the Distributed Network Protocol (DNP3) protocol. This strategy is performed in two complementary steps using supervised and unsupervised machine learning techniques. Specifically, a decision tree classifier is first employed to monitor DNP3 network flow and identify particular DNP3 cyberattacks. Then, an autoencoder-based anomaly detector is applied to detect DNP anomalies that could be caused by a potential security violation or electrical disturbances. Results based on DNP3 network traffic data generated using an emulator showed the promising detection performance of the DIDEROT strategy. In siniosoglou2021unified, an effective approach merging the benefits of Autoencoder and Generative Adversarial Network (GAN) models is introduced to detect anomalies and classify malicious Modbus/TCP and DNP3

attacks. This coupled approach has been designed to detect and classify anomalies in smart grid environments. Three datasets have been used to verify the efficiency of this approach, namely Modbus/TCP network flows, operational data, and DNP3 network flows, and showed its superior performance compared to other machine learning models. The authors in khan2019hml proposed a hybrid multilevel scheme to detect intrusion in SCADA systems. In this detection scheme, at first, dimensionality reduction approaches, including principal component analysis, are applied to extract the relevant features. Then, a Bloom filter is applied for generating a signature database and for anomaly detection. Lastly, an instance-based categorizer is trained and tested for predicting anomalies. Results based on actual data from the gas pipeline demonstrated the detection efficiency of this hybrid approach. In [37], intrusion detection and classification approach has been presented based on Intrusion Weighted Particle-based Cuckoo Search Optimization (IWP-CSO) and Hierarchical Neuron Architecture-based Neural Network (HNA-NN). Essentially, this approach is implemented in two complementary tasks; the dimensionality of input features is reduced IWP-CSO and NN, then HNA-NN is applied to the selected features to detect and classify cyberattacks. It has been shown that the amalgamation between IWP-CSO optimization with the HNA-NN classifier enables increasing the classification rate by 12%. Recently in [38], an ensemble deep learning strategy combining the benefits of feedforward neural network (FNN) and long-short term memory (LSTM) has been designed for detecting temporally uncorrelated and correlated attacks in SCADA networks. Results showed the outperformance of this ensemble model compared to the standalone FNN and LSTM-based IDSs. A semi-supervised deep learning autoencoder is employed in [39] to improve the detection of SCADA attacks in gas pipeline control systems. Specifically, this scheme learns the most relevant features based on attacks-free data; thus, malicious data could be easily flagged out because it leads to a high reconstruction error. Much research has been done in recent years on developing intrusion detection mechanisms for SCADA systems. For instance, see some relevant survey papers [40–44].

3 Methodology

Deep learning methods have shown its effectiveness in different areas including speech recognition [45], computer vision [46], and natural language processing [47]. The deep learning algorithm processes data through multiple layers of connected artificial neurons. It automatically extracts hierarchical information from the input data, and feature engineering can be largely avoided. The problem of discriminating cyber-attacks is well recognized as a pattern recognition problem, a field in which deep learning algorithm has been demonstrated as one of the most competitive methods.

3.1 The proposed stacked deep learning-driven method

To discriminate cyber attacks from normal operations, we propose a stacked deep learning model that ensembles the results of five forward neural networks with three fully connected hidden layers. Stack generalization [48] is a technique widely used in the machine learning community to boost performance of basic learners. Stack generalization learns a model based on the predictions from basic learners. In this paper, we use model averaging, which is a special case of stacked generalization, to boost the performance of deep learning models in cyber-attack detection in SCADA systems.

Before presenting the details, we will show theoretically that a stacked deep neural network, which averages the results of N individual deep learning models, can be arbitrarily accurate in a classification problem, and its prediction mean squared error (MSE) can be arbitrarily small compared to an individual deep learning model. Consider a binary cyber-attack detection problem, where $y = 1$ denotes an attack, $y = 0$ denotes a natural event. Suppose we already trained N deep learning models $m_l(x), l = 1, \ldots, N$, which are all better than random guessing, that is $\mathbb{P}(m_l(x) = y) = p > 0.5$. The stacked deep learning model is defined as by Eq. (1),

$$\begin{cases} m(x) = 1 & \text{if } \sum_{l=1}^{N} m_l(x)/N > 0.5 \\ m(x) = 0 & \text{otherwise.} \end{cases} \tag{1}$$

Assume that the N deep learning models are independent. Without loss of generality, assume that the new feature x_{new} corresponds to an attack, then the probability of the stacked deep learning model predicts an attack is

$$\begin{aligned} \mathbb{P}(m(x_{new}) = 1) &= \mathbb{P}\left(\frac{1}{N}\sum_{l=1}^{N} m_l(x_{new}) > \frac{1}{2}\right) \\ &\geq 1 - \exp\left(-2\left(p - \frac{1}{2}\right)^2 N\right), \end{aligned} \tag{2}$$

where the last inequality follows from Hoeffding's inequality for Bernoulli random variables. As N becomes large, the probability in (2) approaches 1, and this shows that the stacked deep learning model can be arbitrarily accurate as the number of averaged models increases. Next, we show that the prediction MSE of the stacked deep

learning model can be arbitrarily small. The prediction MSE of the stacked deep learning model is

$$\mathbb{E}(m(x_{new}) - 1)^2 = \mathbb{P}\left(\frac{1}{N}\sum_{l=1}^{N} m_l(x_{new}) < \frac{1}{2}\right) \leq \exp\left(-2\left(p - \frac{1}{2}\right)^2 N\right), \tag{3}$$

again, the last inequality follows from Hoeffding's inequality. As the number of averaged models N increases, the stacked deep learning model's prediction MSE can be arbitrarily close to 0 as shown in (3). Because the prediction MSE of an individual model is fixed, we proved that the stacked deep learning model's prediction MSE can be arbitrarily small compared to an individual deep learning model.

As proved above, it is necessary to employ stacked deep learning models because the stacked model increases the prediction accuracy while decreasing prediction MSE. The stacked deep learning model enables more accurate detection of cyber-attacks in SCADA systems, and it is less vulnerable than individual deep learning models because of its low prediction MSE. The theoretical analysis is inspired by the ensemble learning methods, e.g., Adaboost [10] and random forest [11]. It has been proved that Adaboost and random forest improves the classification accuracy of individual trees, where correlations between individual models are also discussed. In practice, the individual models are trained using the same dataset, and therefore individual models are correlated. A full analysis of the theoretical property of stacked model under correlated individual model is beyond the scope of the current paper. Practically, the stacked deep learning model indeed has higher accuracy compared to an individual deep learning model as shown in the empirical study.

3.2 The architecture of the proposed model and implementing details

Next, we introduce individual deep learning models' details and how we construct the stacked deep learning model. We construct five neural networks to demonstrate the robustness of the network's performance, that is, a slight change in the structure of the network will not deteriorate the detection ability. The structure of our network is motivated as follows. Networks with more hidden layers will generally enrich the represented features and achieve greater capability in solving real problems [49]. For example, successful plans for the famous ImageNet dataset all exploit huge neural networks with more than 30 layers and millions of parameters [49]. Researchers also benefit from deep neural networks when solving scientific problems such as protein structure prediction [50], solving the many-electron Schrödinger equation [51], and fighting the Coronavirus disease [52]. However, the depth of the neural network is restricted by the available samples. Otherwise, an overly large network will overfit the data. After some experiments, we found that a three-layer network can discriminate the malicious attacks in all the two classes, three classes, and multiple classes detection problems. Networks with more than three layers will generally overfit the data. The same neural network structure has been used for two classes, three classes, and multiple classes cyber-attack detection. The number of hidden neurons in the five networks are (80, 60, 60), (80, 80, 60), (100, 80, 80), (120, 100, 80) and (180, 120, 80), respectively, all the layers are fully connected. The activation function for the hidden layers is the rectified linear units (ReLU) function, while the activation function for the output layer is the sigmoid function and the softmax function for the two classes data and the multiple classes data, respectively. Table 1 shows the structure of the network for the two classes problem. The input layer has 122 units, i.e., 122 input features, and the output layer has 1 unit. There are in total 19,221 parameters in this example. The networks are implemented by Tensorflow 2.3.0 and Keras 2.4.3. The optimizer is RMSprop, all the networks are trained for 4000 epochs with a batch size 128.

We construct five networks to suit our computational resources, and the number of neurons are chosen to represent small (Network 1) to moderately large (Network 5) networks. The stacked network averages the outputs of the five neural networks. The accuracy of the stacked deep learning model for cyber-attack detection is higher than individual deep learning models. Algorithm 1 summarizes the steps to learn and predict with the stacked deep learning model.

Table 1 Structure of the used deep learning models

OPERATION		DATA DIMENSIONS	WEIGHTS(N)	WEIGHTS(%)
Input	#####	122		
Dense	XXXXX ----------		9840	51.2%
relu	#####	80		
BatchNormalization	μ|σ ----------		320	1.7%
	#####	80		
Dropout	| || ----------		0	0.0%
	#####	80		
Dense	XXXXX ----------		4860	25.3%
relu	#####	60		
BatchNormalization	μ|σ ----------		240	1.2%
	#####	60		
Dropout	| || ----------		0	0.0%
	#####	60		
Dense	XXXXX ----------		3660	19.0%
relu	#####	60		
BatchNormalization	μ|σ ----------		240	1.2%
	#####	60		
Dropout	| || ----------		0	0.0%
	#####	60		
Dense	XXXXX ----------		61	0.3%
sigmoid	#####	1		

Algorithm 1: The algorithm for the stacked deep learning model.

Input: Training data: $(x_i, y_i), i = 1, \ldots, n$; New data: x_{new}
Output: The predictions by the stacked deep learning model: y_{pred}
for $l = 1 : 5$ **do**
1. Train individual deep learning model m_l using the training data;
2. Obtain the output of the trained model for the new data: $\hat{y}_l = m_l(x_{new})$.

end
Calculate the prediction of the stacked model: $y_{pred} = (\hat{y}_1 + \ldots, \hat{y}_5)/5$.

3.3 The algorithm for training an individual deep learning model

In the following, we describe the details the training algorithm for a single deep leanring model. Table 1 summarizes the structure of the constructed neural network, the number of neurons, and the number of weights in each layer. Each densely connected layer is followed by a batch normalization layer and a dropout layer to prevent overfitting. The number of weights in the network depends on the number of input features and the network structure. For the network used for cyber-attack detection in the SCADA system, the number of weights is 19,221 in total. In each middle layer, the input features are first linearly transformed followed by a rectified linear units (ReLU) activation function. Denote the output of the lth layer as $\mathbf{a}^l = (a_1^l, \ldots, a_n^l)^\top$, where n is the number of neurons in the ith layer. For example, $n = 60$ in the second layer of our first network. The first layer will be the input layer, and the last layer is the output layer. Denote L as the number of layers. The output of the $(l+1)$th layer is

$$\mathbf{a}^{l+1} = \sigma(\mathbf{W}^{l+1}\mathbf{a}^l + \mathbf{b}^l),$$

where $\mathbf{W}^{l+1}$ is an $m \times n$ coefficient matrix for the $(l+1)th$ layer, m is the number of neurons in the $(l+1)$th layer, $\mathbf{b}^l$ is the bias vector, and $\sigma(x)$ is the activation function. For the hidden layers, $\sigma(x) = \max(0, x)$ is the ReLU function. For the output layer, $\sigma(x) = 1/(1 + \exp(x))$ is the sigmoid function for binary classification, and $\sigma(x) = exp(x_i)/(\exp(x_1) + \ldots + \exp(x_c)), i = 1, \ldots, c$, is the softmax function for multi-class classification where c is the number of events.

Denote (x, y) as one observed data, where x is the feature set, $y = (y_1, \ldots, y_c)$ is the one-hot encoding of the observed event. For binary classification, where we discriminate whether an event is a cyber-attack, we use the binary cross-entropy as the loss function. For multi-class classification, where we also determine the type of the attack, we use the multi-class cross-entropy as the loss function. More specifically, the loss function is

$$L(\mathbf{W}, \mathbf{b}) = \sum_{j=1}^{c} y_j \log(a_j^L). \tag{4}$$

To minimize the objective function and estimate the coefficients, we perform the stochastic gradient descent (SGD) procedure to train the constructed neural network. The workhorse of SGD is the backpropagation (BP) algorithm, which computes the gradient of the objective function with regard to the parameters of the network, i.e., the weights and biases. The BP algorithm computes the derivative of the lost function in an iterative way with regard to coefficients from the last layer back to the first layer; that is how the name came from. Denote $\mathbf{z}^l = \mathbf{W}^{l+1}\mathbf{a}^l + \mathbf{b}^l$, and $\delta^l = \partial L/\partial \mathbf{z}^l$. Let $\odot$ denote the Hadamard product of two vectors, i.e., element-wise products. The BP algorithm for one observed sample is shown in Algorithm 2. The SGD algorithm is presented in Algorithm 3.

Algorithm 2: The backpropagation algorithm.

Input: Data: (x, y); The number of layers: L.
Output: The partial derivatives:$\partial L/\partial \mathbf{W}$ and $\partial L/\partial \mathbf{b}$.
Calculates the activations $\mathbf{a}^2, \ldots, \mathbf{a}^L$.
for $l = L : 2$ **do**
1. $\delta^L = \frac{\partial L}{\partial \mathbf{a}^L} \odot \sigma'(\mathbf{z}^L)$.
2. $\delta^l = ((\mathbf{W}^{l+1})^\top \delta^{l+1}) \odot \sigma'(\mathbf{z}^l)$ for $l < L$.
3. $\frac{\partial L}{\partial \mathbf{b}^l} = \delta^l$.
4. $\frac{\partial L}{\partial \mathbf{W}^l} = \delta^l (\mathbf{a}^{l-1})^\top$.

end

Algorithm 3: The stochastic gradient descent algorithm.

Input: Data: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$; The number of epochs: E; The size of mini-batches: m.
Output: Network coefficients:$\mathbf{W}$ and $\mathbf{b}$.
Randomly initialize the coefficients $\mathbf{W}$ and $\mathbf{b}$.
for $i = 1 : E$ **do**
1. Randomly shuffule the input data and split it into blocks of size m.
2. For each mini-batch, calculate the partial derivatives by the BP algorithm and update the coefficients by

$$\mathbf{W} \to \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}$$

$$\mathbf{b} \to \mathbf{b} - \eta \frac{\partial L}{\partial \mathbf{b}}$$

end

The technique of dropouts is used to prevent overfitting [53], and the technique of batch normalization [54] is also adopted because it greatly accelerates training. The

network is trained for 4000 epochs, and in each epoch of training, the training data are shuffled and split into mini-batches of size 128. Figure 1 illustrates the flowchart of the proposed cyber-attack detection procedure. The experiment is divided into the training phase and the detection phase. In more detail, in the data preprocessing step, we remove missing values and standardize numeric features. In the training step, we train the neural network using the SGD algorithm. Finally, the trained model is verified using the testing data.

As an illustration of the convergence of the training step, Fig. 2 shows the history of training a neural network to discriminate the cyber attacks. In the long run, the loss decreases and the accuracy increases in the training process, meaning that the deep learning model gradually learned the structure of the data, and converged when the training completed. In each epoch, the computing cost of the stochastic gradient descent is approximately $O(nm)$, where n is the number of samples in the epoch, and m is the number of parameters in the network, e.g., the number of weights and biases in the network. Therefore, the computational burden of training increases as the number of epochs and samples, the number of neurons in each layer, and the network's depth increase. Training deep learning methods can be significantly accelerated by modern parallel computing devices and software. The hardware in our experiment is a cluster with 24 Intel Xeon E5-2650 CPUs, each with 64 gigabytes of memory. The model is implemented by Python programming language, and the deep neural networks are coded and trained by Python packages Keras and Tensorflow.

We experimented different setups for the neural network, such as different numbers of neurons in each hidden layer and different numbers of training epochs. All constructed neural networks demonstrate very similar performance.

3.4 Metrics of effectiveness

The confusion matrix (see Table 2) is used to evaluate and compare the IDS performance of the considered methods. True positives (TP) refers to the number of intrusions (attacks), False Positive (FP) represents the number of typical connections flagged out as attacks, True Negative (TN) is the number of typical observations declared as normal. False Negative (FN) is the number of attacks flagged out as typical observations.

To check the detection performance of investigated schemes for discriminating cyber-attacks, we adopted four common performance metrics:

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

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{8}$$

$$\text{F1} = 2\frac{\text{Precision.Sensitivity}}{\text{Precision} + \text{Sensitivity}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{9}$$

The accuracy (5) assesses the proportion of correct detections. A higher accuracy value indicates more satisfyingz overall intrusion detection. Sensitivity (6) refers to the capability to identify cyber-attacks correctly. Note that the recall is similar to sensitivity in binary classification. Meanwhile, specificity (7) gives the proportion of actual negatives that are correctly identified. Specificity points out the capacity to correctly discriminate typical observations. Precision (8) quantifies the relevance of the detected positives, and F1-score (9) denotes the harmonic average of precision and sensitivity.

4 SCADA system testbed description

To evaluate the performance of the deep-based approach for attack classification in SCADA systems, the dataset we use in this work was developed at the Mississippi State University SCADA Laboratory [55]. The dataset is generated using the testbed shown in Fig. 3. The power system includes two power generators (G1 and G2), four breakers (BR1, BR2, BR3, and BR4) that are controlled by four Intelligent Electronic Devices (IEDs) (R1, R2, R3, and R4), respectively, and two transmission lines, one relies BR1 to BR2 and the other is between BR3 and BR4. The whole networked system is then monitored through intrusion detection system SNORT and Syslog. The dataset provides measurements during typical activities and when the system runs under the following types of attacks [55]:

- *Short-circuit fault* The attacker can create a short-circuit everywhere in the targeted power line.
- *Line maintenance* In this scenario, the attacker turns off one or numerous breakers according to the line to be maintained.
- *Remote tripping command injection* Here, the attacker generates control commands to open one or more breakers.
- *Relay setting change* Attacker alters breakers setting to prevent them from responding to actual fault or a valid command.

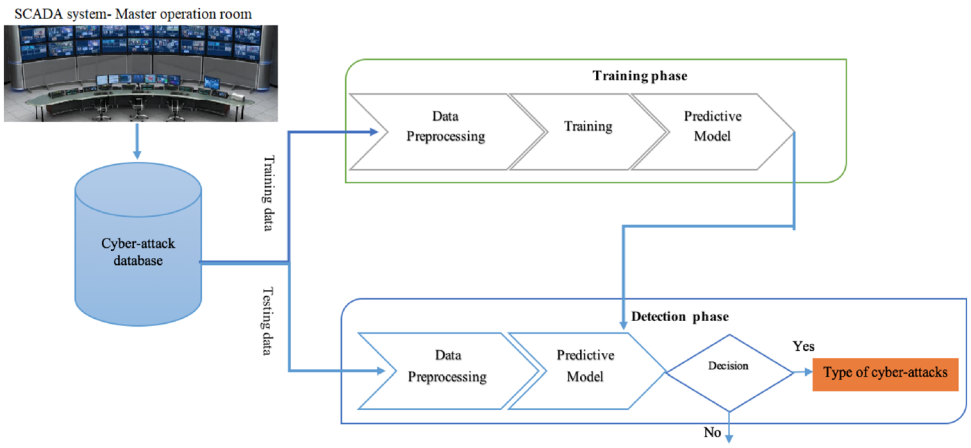

Fig. 1 The flowchart of the cyber-attack detection procedure

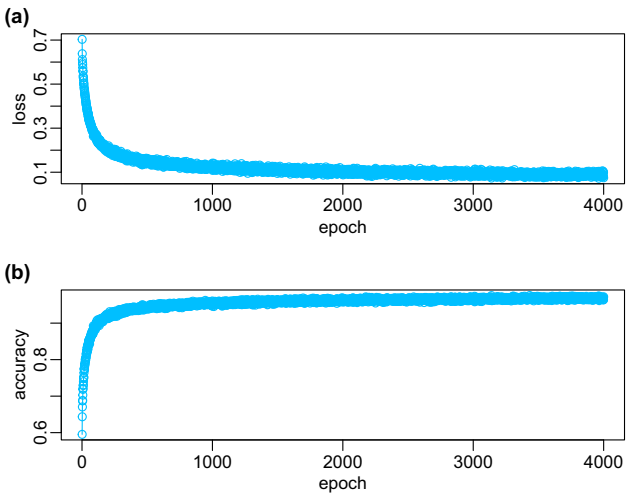

Fig. 2 The history for training a neural network to discriminate the cyber attacks. Upper panel (**a**), the history of the loss function. Lower panel (**b**), the history of the accuracy measure

– *Data injection* Attacker changes typical values of current, voltage, sequence components to create various faults in the system.

This study investigates the designed deep learning model's detection performance to detect intrusion on smart grid systems. Besides, we compare the proposed deep learning-driven approach with the state-of-the-state art machine learning methods (i.e., Nearest neighbor, Random forests, Naive Bayes, Adaboost, Support Vector Machine, oneR, and coupled Adaboost +JRipper) applied to the same data sets [55]. The comparison is performed using three different classification scenarios, namely multiclass, three-class, and binary classification.

– *Multiclass* The multiclass experiment contains 37 event scenarios, including attack events, normal operations, and natural events. In this experiment, each event scenario has its class and is discriminated independently by the models, which means that there are 37 classes in total.
– *Three-class* The 37 scenarios are classified into three classes: attack class containing 28 events, natural class containing 8 events, or No events class with 1 event.
– *Binary* There are two classes: attack class (28 events) and normal class (9 events).

The data is extracted from fifteen datasets, containing thousands of individual measurements throughout the power system for every type of event. The datasets have been randomly sampled at 1% for reducing the size and verifying the efficiency of small sample sizes. Essentially, there is an average of 294 "No event" data points, 3711 attack data points, and 1221 natural data points utilized over the classification schemes [55]. Here, we compared the proposed deep learning-driven attack detection scheme's detection performance with results obtained in [55] by using the baseline machine learning methods.

The heatmaps of the accuracy of the deep learning and machine learning classifiers over the 15 datasets when applied respectively to multiclass, three-class, and binary classification are depicted in Fig. 4a–c. Figure 6 displays the heatmap of the averaged accuracy over the 15 datasets for each method. It should be noted that tenfold cross-validation has been performed for each data set. Accuracy assesses the rate of correct classifications. A high accuracy value indicates a satisfying overall classification capability. We observe from Fig. 4a–c that both deep learning and shallow methods provide consistent results no matter what the used data set. Only relatively minor changes for each model, their performances are robust and consistent regardless of the data set. For example, the stacked deep learning model's accuracy fluctuates between 94.63 and 96.1 when applied to the 15 data sets for multiclass discrimination.

From Figs. 4 and 6, we observe that the deep learning models are robust to the shape of the networks, the

Table 2 Confusion matrix associated to intrusion detection

	IDS decision		
		Attack	No attack
Actual condition	Attack	True positive (TP)	False negative (FN)
	No attack	False positive (FP)	True negative (TN)

performance measures are close to each other for all the five constructed neural networks in all the classification tasks. For example, the average accuracy of Network 1 to Network 5 is 97.00%, 97.01%, 97.06%, 97.11%, and 97.03%, respectively, for binary intrusion detection over the 15 datasets. Larger networks, e.g., Network 5, do not necessarily outperform smaller networks, e.g., Network 3. Moreover, the stacked neural network, which pools the five networks' output, boosts the accuracy by a large margin compared to any single network; the average accuracy is 97.36% for binary classification tasks.

A paired t test is used to compare the accuracy of an individual deep learning model and the stacked deep learning model. The null hypothesis is that the compared accuracies are the same, the alternative hypothesis is that the accuracy of the stacked deep learning model is higher than that of an individual deep learning model. From Fig. 5, we can see that most of the p values are small, especially for the multiclass intrusion detection, and the test is rejected under a significance level of 0.05, meaning that the stacked deep learning model performs significantly better than individual deep learning models in terms of accuracy.

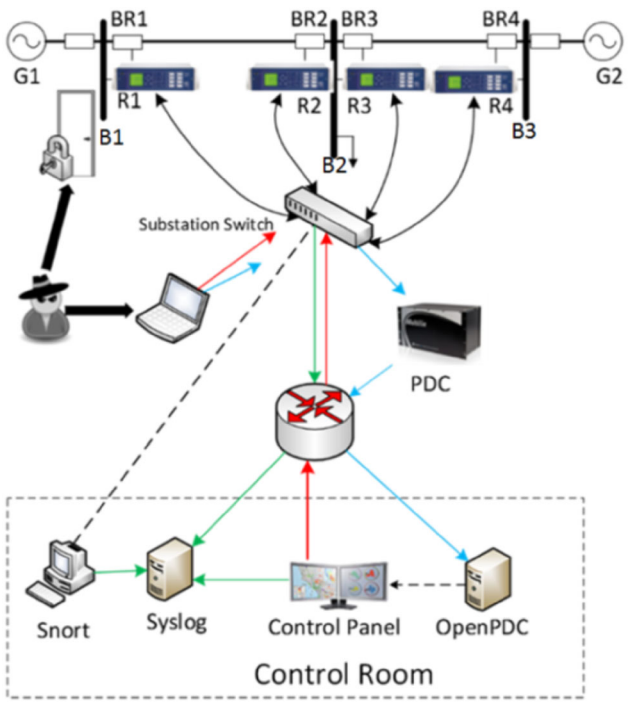

Fig. 3 Power system testbed [55]

As shown in Fig. 4, shallow machine learning models including oneR, NNge, RF, Naïve Bayes, and SVM algorithms achieved low accuracy values; these traditional methods do not discriminate the attack events very well, especially for multi-class detection. For example, Naïve Bayes only achieves an average accuracy of 11.3%, 35.35%, and 20.48% in multi-class, three-class, and binary classification, respectively. All the deep learning driven approaches exhibited superior performance compared to shallow methods over the 15 datasets for all classification types (i.e., binary, triple, and multiple). The highest overall accuracy is achieved by the stacked deep learning approach (Fig. 6). For multi-class, three-class, and binary classification, the deep learning approach dominates all baseline models in terms of accuracy by obtaining 95.52% (multi-class), 97.38% (three-class), and 97.36% (binary), respectively. Especially, the deep learning approach improves the detection accuracy by a large margin (more than 5%) in multi-class classification compared to traditional methods, signifying an exceptional capacity to discriminate different malicious and anomalous events. The primary reason may owe that deep learning models can extract relevant information from complex multivariate data. The Adaboost+JRipper approach follows the deep learning approach by achieving average accuracy values of 89.21% (multi-class), 95.26% (three-class), and 95.34% (binary).

Whereas accuracy gives a global indication of classifier performance, precision, recall, and F-measure are more exhaustive indicators of classifier errors. Recall measures the rate of a true positive number to the total numbers of samples in the positive class, while precision corresponds to the positive predictive value. F-measure is calculated using both recall and precision. It is the harmonic mean of precision and recall. Figure 7a–c depicts respectively, the heatmap results for averaged recall, precision, and F-measure over the 15 datasets. The stacked deep learning approach reaches the best precision for multiclass (0.9859) and binary (0.9781) classification (Fig. 7). It is followed by the Adaboost+JRipper approach with 0.8559 (multiclass) and 0.9489 (binary). Adaboost+JRipper achieved the best average precision value of 0.997 for the three-class case, followed by the stacked deep learning model with 0.9779.

The averaged recall by each approach for the three considered classification types are depicted in Fig. 7b.

Essentially, recall quantifies the true positive rate; that is, it indicates which method senses cyber-attacks most appropriately. Overall, the stacked deep learning method has the highest recall values than other traditional methods by achieving recall values of 0.9862 (multiclass), 0.9858 (three-class), 0.9846 (binary classification). Also, from Fig. 7b, one observes that some simple methods as Naïve Bayes and OneR achieve high averaged recall values (0.961 and 1, respectively), while RF can achieve moderate performance (i.e., 0.7943 (multiclass), 0.9255 (three-class), and 0.8865 (binary)). It should be noted that high recall values and low precision values of some classifiers (e.g., OneR and Naïve Bayes) is an indication of the classifier's bias towards the positive (malicious attack) class. In other words, these classifiers enable good classification of malicious attacks, but with an expense of false-positive values. On the other hand, the proposed deep learning-based detection approach can consistently detect malicious attacks with high recall and precision values. Figure 7c summarizes the averaged F-measure values obtained by each method. It can be seen clearly that the deep learning approach overall and the stacked deep learning approach especially performed better than the other shallow learning methods in terms of F-measure in all the cases by reaching the highest overall values of 0.9861 (multiclass), 0.9818 (three-class), and 0.9813 (binary).

In summary, the deep learning approach, especially, the stacked deep learning approach, demonstrated a promising detection performance. The comparison results recommend that the deep model significantly outperforms the shallow machine learning methods in reliably discriminating abnormal events in power systems. This could be attributed to the extended capacity and the deep learning model's flexibility in extracting relevant information from multivariate data.

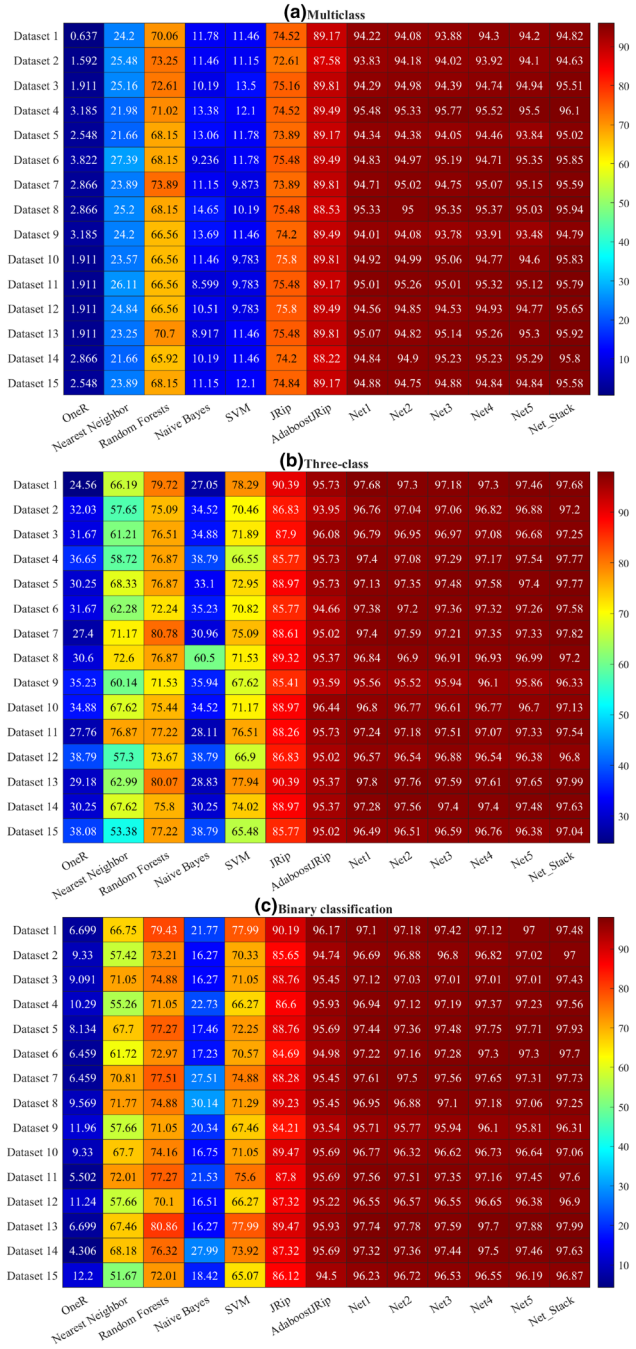

Fig. 4 Accuracy **a** multiclass, **b** Three-class, and **c** Binary

4.1 Features importance identification

The considered dataset contains 128 features (Table 3). There are 4 phasor measurement units (PMUs) or synchrophasors, which measure the electrical waves on an electricity grid, records 29 features each for a total of 116 PMU measurements.

Still within the cyber-attacks detection framework, it is worth noticing that some features generally have a low contribution to the modeling and cyber-attacks detection, but they are used in the modeling and can increase the complexity of the model. Thus, identifying essential features is undoubtedly an essential step in designing parsimonious modeling by eliminating useless features. The identification of important features to cyber-attacks detection is achieved by the random forest algorithm, as depicted in Fig. 8. We describe the permutation importance of a feature as follows. The out-of-bag accuracy of the trained Random Forest is denoted as A_{base}. To calculate the importance of a feature, we permute its values and then pass all the out-of-bag samples back through the Random Forest. The resulting accuracy is denoted as A_{perm}. The variable importance of that feature is the drop in overall accuracy caused by permutation, that is

$$variable\ importance = A_{base} - A_{perm}.$$

Fig. 5 The P value for testing the hypothesis that the stacked deep learning model has higher accuracy than individual deep learning models. **a** multiclass, **b** Three-class, and **c** Binary

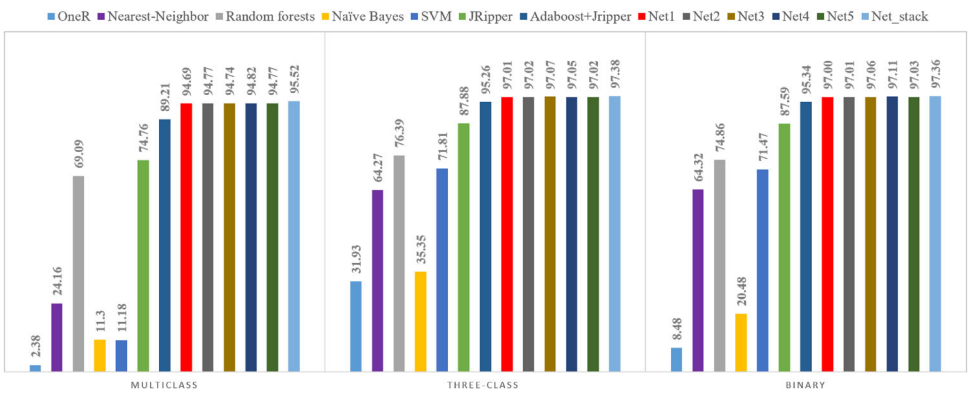

Fig. 6 Binary class accuracy over Fifteen Datasets

Features were ordered according to their importance. We observe the absence of notable contributions of specific features to the detection process. The largest contribution is obtained by the feature called R4.PM2.V (i.e., Phase A-C Voltage Phase Magnitude measured by PMU R4) with 2.59%. The second observation is that 69 features are contributing to the learning value with 95.24% (Fig. 8). These features include Voltage Phase Magnitudes Voltage Phase Angles, Current Phase Magnitudes, Zero Voltage Phase Angles, Zero Current Phase Magnitudes, and Appearance Impedance measurements. Using only the features with 95.24% contribution may enable dimensionality reduction and make a model less complicated than including all features.

5 Gas pipeline system

The dataset we use in this work was created at the SCADA Laboratory of Mississippi State University [56] using a real-world gas pipeline system. It includes two actuators to monitor the system state and maintain the pressure level; the network's interface serial Modbus RTU and the supervisory controls (MTU and the iFIX HMI). The dataset represents the MODBUS traffic, which was collected by a network logger through an RS-232 connection. The dataset provides measurements during normal operating activities and when the system runs under different types of attacks. There are three groups of attacks:

- *Reconnaissance Attacks* include different scanning operations (e.g., address scan, function code scan, device identification attack, and points scan.
- *Command Injection Attacks* consist of injecting spoofed (fake, forged) control and administration commands to modify system behavior. Three types were implemented, including malicious state command injection (MSCI), malicious parameter command injection (MPCI), and malicious function code command injection (MFCI) attacks.
- *Denial-of-Service Attacks* through resources exhausting, such attacks make an effort to shut down a part or the whole SCADA system. Two examples of DOS are available: the invalid cyclic redundancy code (CRC) and jamming attack.

From a total of 274,627 records that make the dataset, normal traffic is about 214,580 instances (i.e., 78.13%), and attacks appear in 60,048 instances (i.e., 21.87%). The overall details of such attacks are in [56]. Table 4 summarizes the list and the distribution of normal and attacks records considered in this study.

5.1 Command injection results

SCADA systems have an essential role in monitoring modern plants. With the increase of cyber-attacks, the security of these systems becomes indispensable to avoid serious problems. This section investigates the capability of the proposed deep learning methods in identifying and discriminating malicious intrusions in the gas pipeline system testbed.

We assess the performance of the deep learning-driven attack detection method when applied to discriminate the data injection attack types and normal (attacks-free) RTU transactions. The corresponding validation metrics are calculated and listed in Table 5. Multiclass discrimination results using the deep learning-driven method implies that almost all attacks are well detected, except the address scan attacks. Also, we observe that the proposed method achieved perfect detection of the normal (attacks-free) RTU transactions. As expected, high detection performances are obtained using the deep learning approach due to the simplicity of the command injection attacks compared to data/response injection. The low detection performance for address scan attacks can be attributed to the meager amount of the exemplar data for this type of attack (only 2 data points). The availability of only two instances of address scan attacks compared to the total amount of data makes discrimination difficult.In this experiment, after organizing the data as a binary classification problem (attack-free and malicious data), we applied the deep learning-driven attack detection scheme for further assessment. The corresponding validation metrics are computed and tabulated in Table 6. Similarly to data/response injection, by merging the malicious RTU data points, the two classes' discrimination becomes an easy task, and we obtain a powerful detector. Moreover, the proposed deep learning scheme achieves perfect detection performance when applied to detect DOS attacks (Table 7).

5.2 Features importance identification

Now, we identify the most important variables that contribute to the cyber-detection output. The feature importance calculated using the random forest is shown in Fig. 9. The features which contributed the most discriminating power to detect cyber-attacks are data length and setpoint, followed by control mode and control scheme. While, we can observe from Fig. 9 that six features (i.e., command, invalid data length, invalid function code, PSI, pump state, and solenoid state) barely contribute to cyber-attack detection and be ignored when designing a cyber-attack detector.

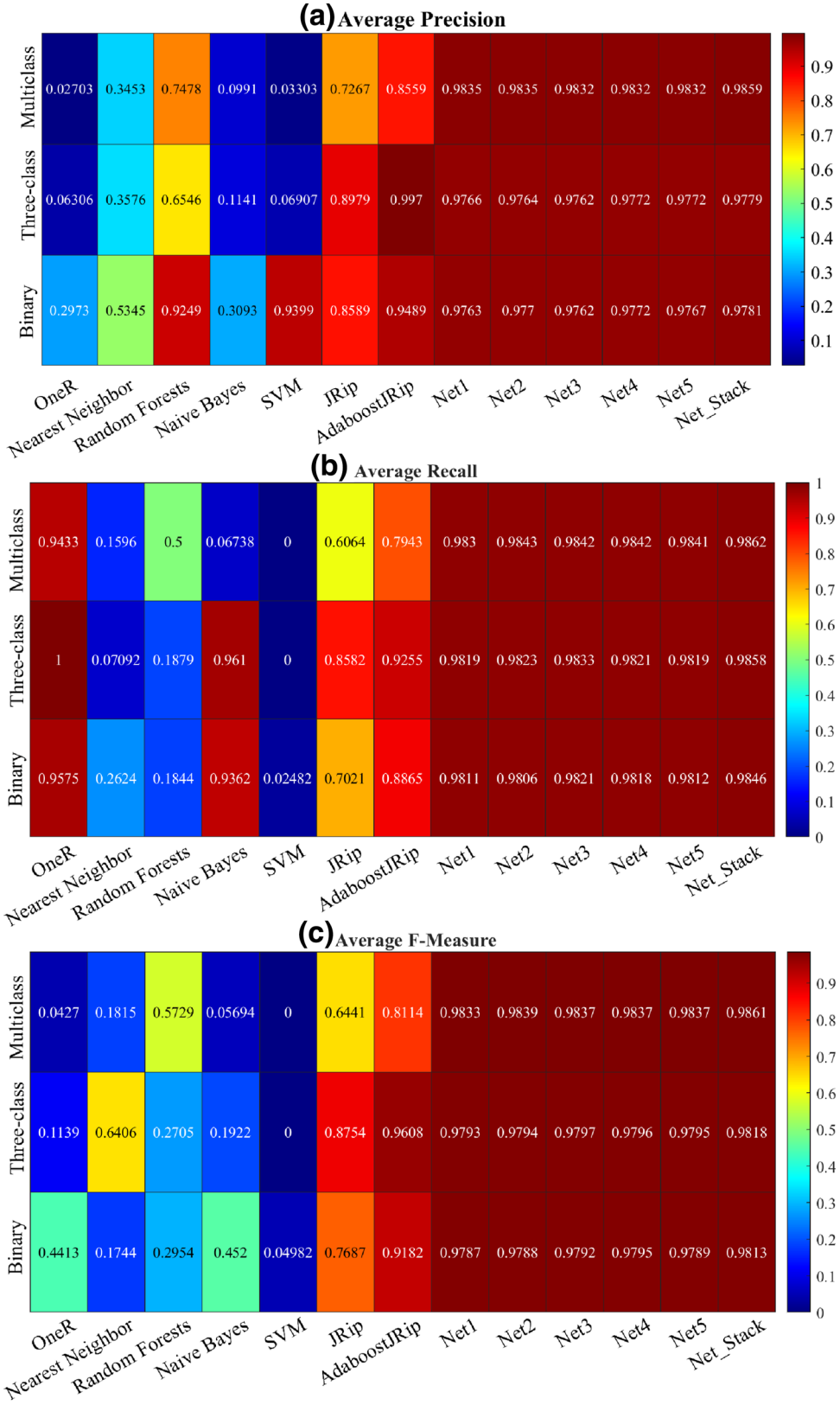

Fig. 7 Averaged Binary class performance over the fifteen datasets

Table 3 Variables definition

Feature	Description
PA1:VH—PA3:VH	Phase A–C Voltage phase angle
PM1:V—PM3:V	Phase AC Voltage phase magnitude
PA4:IH—PA6:IH	Phase A–C Current phase angle
PM4:I—PM6:I	Phase A–C Current phase magnitude
PA7:VH—PA9:VH	Pos.–Neg.–Zero Voltage phase angle
PM7:V—PM9:V	Pos.–Neg.–Zero Voltage phase magnitude
PA10:VH—PA12:VH	Pos.–Neg.–Zero Current phase angle
PM10:V—PM12:V	Pos.–Neg.–Zero Current phase magnitude
F	Frequency for relays
DF	Frequency delta (dF/dt) for relays
PA:Z	Appearance impedance for relays
PA:ZH	Appearance impedance angle for relays
S	Status flag for relays

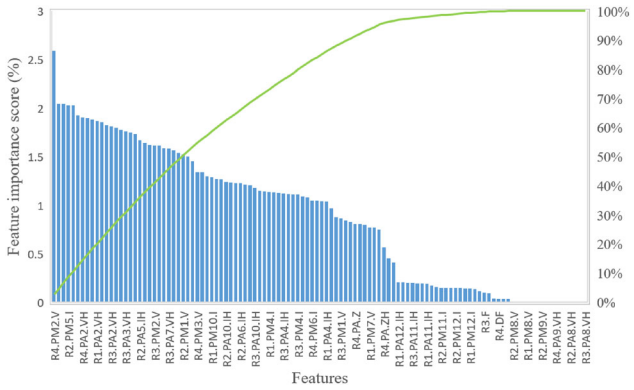

Fig. 8 Feature importance ranked using the RF algorithm

Table 4 Distribution of training data instances

Command injection	
Dataset	Instances
Normal	28,086
Address scan	2
Func code scan	9
Illegal setpoint	197
Illegal PID	49

We repeated the experiment based on the deep learning model using only the four important variables (i.e., control mode, control scheme, data length, and setpoint), which permit reduce the dimensionality of the dataset (Tables 8, 9).

In summary, this study demonstrated the promising performance of a stacked deep learning-driven approach for improving intrusion detection in industrial systems. This approach also exhibited a suitable capacity in detecting broad classes of attacks in SCADA systems using a very basic set of features. Results revealed that the stacked deep learning approach exhibits superior detection performance in comparison to the baseline machine learning methods and also to standalone deep learning models. It has also been shown that by using feature importance the data dimensionality can be reduced, and a more parsimonious deep learning model can be designed.

6 Conclusion

Accurate cyber-attacks detection in modern industrial systems is undoubtedly indispensable to enhance their resilience and guarantee continuous production with the desired specifications. However, traditional intrusion detection systems based on shallow machine learning methods are generally limited for appropriately detecting malicious attacks in modern industrial systems. As shown in the literature, deep learning technologies are promising for intrusion detection in SCADA systems because their ability to tackle the non-linear, dynamic SCADA data. Towards this purpose, this paper introduces a stacked deep learning-driven approach for cyber-attacks detection. Results show that the proposed stacked deep learning model can deeply learn the suspicious activities' relevant features and recognize them from normal activities. Thus, the stacked deep learning-based intrusion detection method outperforms various state-of-the-art shallow methods, including the standalone deep learning models, Nearest neighbor, Random forests, Naive Bayes, Adaboost, Support Vector Machine, and oneR. Besides detecting the malicious attacks in the two considered SCADA systems, we also provide the feature importance on the cyber-attacks

Table 5 Detection performance of the proposed detector for command injection attacks (Multiclass)

	Accuracy	Sensitivity	Specificity	Precision	Recall	F-measure
Address scan	0.9998	0	0.9999	0	0	0
Function code scan	0.9999	0.9	0.9999	0.8333	0.9	0.85
Good	0.9999	0.9999	1	1	0.9999	0.9999
Illegal setpoint	1	0.995	1	1	0.995	0.9974
PID modification	1	1	1	0.9833	1	0.9909

Table 6 Detection performance of the proposed detector for command injection attacks (Binary classification)

	Accuracy	Sensitivity	Specificity	Precision	Recall	F-measure
Normal	0.9999	1	0.9960	1	1	1
Malicious	0.9999	0.9960	1	0.9963	0.9960	0.9961

Table 7 Detection performance of the proposed detector for DOS attacks detection (Binary classification)

	Accuracy	Sensitivity	Specificity	Precision	Recall	F-measure
Normal	1	1	1	1	1	1
Malicious	1	1	1	1	1	1

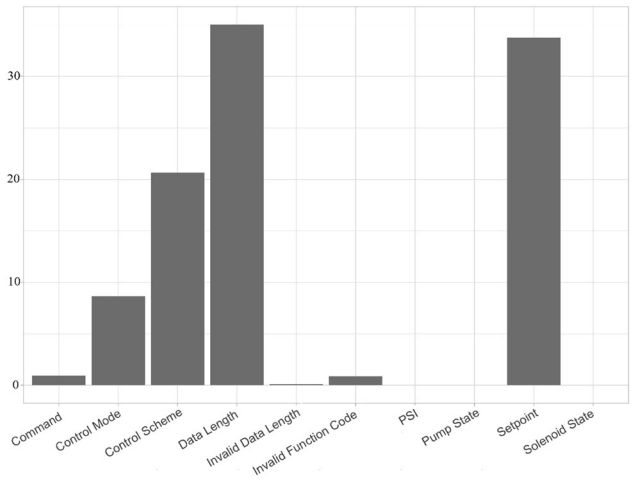

Fig. 9 The feature importance for detecting cyber-attacks in the gas pipeline system

detection process based on the Random Forest procedure. Feature importance identification enables dimensionality reduction and designing parsimonious and less complicated models.

As the time-series data from two investigated SCADA systems are multiresolution in nature and contain significant temporal noises, it would be attractive to build multiscale deep learning models involving wavelet-based presentations to improve cyber-attack detection. Another important direction of improvement is using the developed stacked deep learning models to design an intrusion detection system for the internet of things (IoT) applications [57].

Acknowledgements This publication is based upon work supported by King Abdullah University of Science and Technology (KAUST), Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7-3800. Wu Wang would like to thank the High-performance Computing Platform of Renmin University of China for the computing resources.

Author contributions All authors contributed to the study conception and design. All authors read and approved the final manuscript.

Data availability The datasets generated during and/or analysed during the current study are available in https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

Table 8 Variable selection Command Injection Binary class

	Accuracy	Sensitivity	Specificity	Precision	Recall	F-measure
Good	0.9995	0.9999	0.9575	0.9996	0.9999	0.9998
Malicious	0.9995	0.9575	0.9999	0.9923	0.9575	0.9742

Table 9 Variable selection Command Injection multiple class

	Accuracy	Sensitivity	Specificity	Precision	Recall	F-measure
Address scan	0.9999	0	1	0	0	0
Function code scan	0.9997	0	1	0	0	0
Good	0.9995	0.9999	0.9578	0.9996	0.9999	0.9998
Illegal setpoint	0.9999	0.9950	0.9999	0.9852	0.9950	0.9899
PID modification	1	1	1	1	1	1

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhu, B., Joseph, A., Sastry, S.: A taxonomy of cyber attacks on SCADA systems. In: International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, pp. 380–388. IEEE (2011)
2. Papić, M., Bundalo, Z., Bundalo, D., Stojanović, R., Kovačević, Ž, Pašalić, D., Cvijić, B.: Microcomputer based embedded SCADA and RFID systems implemented on LINUX platform. Microprocess. Microsyst. **63**, 116–127 (2018)
3. East, S., Butts, J., Papa, M., Shenoi, S.: A taxonomy of attacks on the DNP3 protocol. In: International Conference on Critical Infrastructure Protection, pp. 67–81. Springer, New York (2009)
4. Perez, R.L., Adamsky, F., Soua, R., Engel, T.: Machine learning for reliable network attack detection in scada systems. In: 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 633–638. IEEE (2018)
5. Sajid, A., Abbas, H., Saleem, K.: Cloud-assisted IoT-based scada systems security: a review of the state of the art and future challenges. IEEE Access **4**, 1375–1384 (2016)
6. Kamal, P., Abuhussein, A., Shiva, S.: Identifying and scoring vulnerability in scada environments. In: Future Technologies Conference (FTC), pp. 845–857 (2017)
7. Chen, T.M., Abu-Nimeh, S.: Lessons from stuxnet. Computer **44**(4), 91–93 (2011)
8. Assante, M.J.: Confirmation of a coordinated attack on the Ukrainian power grid. In: SANS Industrial Control Systems Security Blog, vol. 207 (2016)
9. Miller, B., Rowe, D.: A survey SCADA of and critical infrastructure incidents. In: Proceedings of the 1st Annual Conference on Research in Information Technology, pp. 51–56 (2012)
10. Bartlett, P., Freund, Y., Lee, W.S., Schapire, R.E.: Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Statist. **26**(5), 1651–1686 (1998)
11. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
12. Tommy, M.: Industrial control system (ICS) cyber attack datasets. https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
13. Bouyeddou, B., Harrou, F., Kadri, B., Sun, Y.: Detecting network cyber-attacks using an integrated statistical approach. Clust. Comput. **24**(2), 1435–1453 (2021)
14. Almalawi, A., Fahad, A., Tari, Z., Alamri, A., AlGhamdi, R., Zomaya, A.Y.: An efficient data-driven clustering technique to detect attacks in scada systems. IEEE Trans. Inf. Forensics Secur. **11**(5), 893–906 (2015)
15. Yang, L., Cao, X., Geng, X.: A novel intelligent assessment method for scada information security risk based on causality analysis. Clust. Comput. **22**(3), 5491–5503 (2019)
16. Pan, Z., Pacheco, J., Hariri, S., Chen, Y., Liu, B.: Context aware anomaly behavior analysis for smart home systems. Int. J. Inf. Commun. Eng. **13**(5), 261–274 (2019)
17. Linda, O., Vollmer, T., Manic, M., Neural network based intrusion detection system for critical infrastructures. In: International Joint Conference on Neural Networks, pp. 1827–1834. IEEE (2009)
18. Sayegh, N., Elhajj, I.H., Kayssi, A., Chehab, A.: SCADA intrusion detection system based on temporal behavior of frequent patterns. In: MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference, pp. 432–438. IEEE (2014)
19. Barbosa, R.R.R., Sadre, R., Pras, A.: Flow whitelisting in scada networks. Int. J. Crit. Infrastruct. Protect. **6**(3–4), 150–158 (2013)
20. Mitchell, R., Chen, I.-R.: A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. **46**(4), 1–29 (2014)
21. Maglaras, L.A., Jiang, J., Cruz, T.: Integrated OCSVM mechanism for intrusion detection in SCADA systems. Electron. Lett. **50**(25), 1935–1936 (2014)
22. Ren, W., Cao, J., Wu, X.: Application of network intrusion detection based on fuzzy c-means clustering algorithm. In: Third International Symposium on Intelligent Information Technology Application, vol. 3, pp. 19–22. IEEE (2009)
23. Fovino, I.N,. Carcano, A., Murel, T.D.L., Trombetta, A., Masera, M.: Modbus/DNP3 state-based intrusion detection system. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 729–736. IEEE (2010)
24. Knorn, F., Leith, D.J.: Adaptive kalman filtering for anomaly detection in software appliances. In: IEEE INFOCOM Workshops, pp. 1–6. IEEE (2008)
25. Ye, N., Chen, Q., Borror, C.M.: EWMA forecast of normal system activity for computer intrusion detection. IEEE Trans. Reliab. **53**(4), 557–566 (2004)
26. Kosut, O., Jia, L., Thomas, R.J., Tong, L.: Malicious data attacks on smart grid state estimation: attack strategies and countermeasures. In: First IEEE International Conference on Smart Grid Communications, pp. 220–225 . IEEE (2010)
27. Giani, A., Bent, R., Hinrichs, M., McQueen, M., Poolla, K.: Metrics for assessment of smart grid data integrity attacks. In: IEEE Power and Energy Society General Meeting, pp. 1–8. IEEE (2012)
28. Pal, S., Sikdar, B., Chow, J.H.: Detecting malicious manipulation of synchrophasor data. In: 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 145–150. IEEE (2015)
29. Keshk, M., Moustafa, N., Sitnikova, E., Creech, G.: Privacy preservation intrusion detection technique for scada systems. In: Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE (2017)
30. Harrou, F., Sun, Y., Hering, A.S., Madakyaru, M., et al.: Statistical Process Monitoring Using Advanced Data-Driven and Deep Learning Approaches: Theory and Practical Applications. Elsevier, Amsterdam (2020)
31. Wang, W., Lee, J., Harrou, F., Sun, Y.: Early detection of Parkinson's disease using deep learning and machine learning. IEEE Access **8**, 147635–147646 (2020)
32. Dairi, A., Harrou, F., Sun, Y., Senouci, M.: Obstacle detection for intelligent transportation systems using deep stacked autoencoder and *k*-nearest neighbor scheme. IEEE Sens. J. **18**(12), 5122–5132 (2018)
33. Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.-Z.: Deep learning for health informatics. IEEE J. Biomed. Health Inform. **21**(1), 4–21 (2016)
34. He, Y., Mendis, G.J., Wei, J.: Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans. Smart Grid **8**(5), 2505–2516 (2017)
35. Yang, H., Cheng, L., Chuah, M.C.: Deep-learning-based network intrusion detection for scada systems. In: 2019 IEEE Conference on Communications and Network Security (CNS), pp. 1–7. IEEE (2019)
36. Radoglou-Grammatikis, P., Sarigiannidis, P., Efstathopoulos, G., Karypidis, P.-A., Sarigiannidis, A.: Diderot: an intrusion detection and prevention system for dnp3-based scada systems. In: Proceedings of the 15th International Conference on Availability, Reliability and Security, pp. 1–8 (2020)

37. Shitharth, S., et al.: An enhanced optimization based algorithm for intrusion detection in scada network. Comput. Secur. **70**, 16–26 (2017)
38. Gao, J., Gan, L., Buschendorf, F., Zhang, L., Liu, H., Li, P., Dong, X., Lu, T.: Omni scada intrusion detection using deep learning algorithms. IEEE Internet Things J. **8**(2), 951–961 (2020)
39. Joshi, C., Khochare, J., Rathod, J., Kazi, F., A semi-supervised approach for detection of scada attacks in gas pipeline control systems. In: IEEE-HYDCON, pp. 1–8. IEEE (2020)
40. Radoglou-Grammatikis, P.I., Sarigiannidis, P.G.: Securing the smart grid: a comprehensive compilation of intrusion detection and prevention systems. IEEE Access **7**, 46595–46620 (2019)
41. Zeng, P., Zhou, P.: Intrusion detection in scada system: a survey. In: Intelligent Computing and Internet of Things, pp. 342–351. Springer (2018)
42. Rakas, S.V.B., Stojanović, M.D., Marković-Petrović, J.D.: A review of research work on network-based scada intrusion detection systems. IEEE Access **8**, 93083–93108 (2020)
43. Quincozes, S.E., Albuquerque, C., Passos, D., Mossé, D.: A survey on intrusion detection and prevention systems in digital substations. Comput. Netw. **184**, 107679 (2021)
44. Cui, L., Qu, Y., Gao, L., Xie, G., Yu, S.: Detecting false data attacks using machine learning techniques in smart grid: a survey. J. Netw. Comput. Appl. 102808 (2020)
45. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
47. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167 (2008)
48. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
49. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
50. Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A.W., Bridgland, A., et al.: Improved protein structure prediction using potentials from deep learning. Nature **577**(7792), 706–710 (2020)
51. Pfau, D., Spencer, J.S., Matthews, A.G., Foulkes, W.M.C.: Ab initio solution of the many-electron schrödinger equation with deep neural networks. Phys. Rev. Res. **2**(3), 033429 (2020)
52. Jamshidi, M., Lalbakhsh, A., Talla, J., Peroutka, Z., Hadjilooei, F., Lalbakhsh, P., Jamshidi, M., La Spada, L., Mirmozafari, M., Dehghani, M., et al.: Artificial intelligence and covid-19: deep learning approaches for diagnosis and treatment. IEEE Access **8**, 109581–109595 (2020)
53. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
54. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
55. Hink, R.C.B., Beaver, J.M., Buckner, M.A., Morris, T., Adhikari, U., Pan, S., Machine learning for power system disturbance and cyber-attack discrimination. In: 7th International symposium on resilient control systems (ISRCS), pp. 1–8. IEEE (2014)
56. Morris, T., Gao, W.: Industrial control system traffic data sets for intrusion detection research. In: International Conference on Critical Infrastructure Protection, pp. 65–78. Springer (2014)
57. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: Ton iot telemetry dataset: a new generation dataset of iot and iiot for data-driven intrusion detection systems. IEEE Access **8**, 165130–165150 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Wu Wang received his B.S. degree in statistics from East China Normal University, Shanghai, China, in2008, and the M.S. and Ph.D. degrees in probability and mathematical statistics from Fudan University,Shanghai, China, in 2011 and 2017, respectively. From 2014 to 2015, he was a Visiting Scholar with theUniversity of Michigan, MI. From 2017 to 2020, he was a Postdoctoral Fellow with the Statistics Program,King Abdullah University of Science and Technology. He joined School of Statistics, Renmin Universityof China as an assistant professor in 2020. Since 2018, he has been a member of the American StatisticalAssociation (ASA) and the International Chinese Statistical Association (ICSA).

Fouzi Harrou received the M.Sc. degree in telecommunications and networking from the University ofParis VI, France, and the Ph.D. degree in systems optimization and security from the University ofTechnology of Troyes (UTT), France. He was an Assistant Professor with UTT for one year and with theInstitute of Automotive and Transport Engineering, Nevers, France, for one year. He was also aPostdoctoral Research Associate with the Systems Modeling and Dependability Laboratory, UTT, forone year. He was a Research Scientist with the Chemical Engineering Department, Texas A&M Universityat Qatar, Doha, Qatar, for three years. He is actually a Research Scientist with the Division of Computer,Electrical, and Mathematical Sciences and Engineering, King Abdullah University of Science andTechnology. He is co-author of the book "Statistical Process Monitoring Using Advanced Data-Drivenand Deep Learning Approaches: Theory and Practical Applications" (Elsevier, 2020). His currentresearch interests include statistical decision theory and its applications, fault detection and diagnosis, anddeep learning.

Benamar Bouyeddou received the Dipl.-Ing in Telecommunications and the Master degree inSystems and Networks of Telecommunications from Abou Bekr Belkaid University (UABB),Algeria, in 2004 and2007, respectively. Currently heis a Phd. student in Telecommunications atthe Abou Bekr Belkaid Tlemcen University and Member of STIC laboratory. He is currently anassistant professor of computer networks and Telecommunications Systems at Dr. Tahar MoulaySaida University, Algeria. His research interests include anomaly detection, computer networksand security, and Internet of things.

Sidi-Mohammed Senouci received his Ph.D. in Computer Science in October 2003 from the Universityof Paris 6 and his HDR from INP Toulouse, France. From December 2004 to August 2010, he wasresearcher in France Telecom R&D (Orange Labs) Lannion. Since September 2010, he is professor at ISAT,a major French postgraduate school located in Nevers, France, and part of the University of Bourgogne.Since October 2017, he is director the laboratory DRIVE EA 1859 collocated in ISAT Nevers. Heparticipated or still participates to several national and European-wide research projects. Among them FP7FOTsis, ITEA CarCoDe, ITEA FUSE-IT and FUI PARFAIT. He holds 7 international patents on thesetopics and published his work in major IEEE conferences and renowned journals. He was co-chair inVehiCom2009, IEEE Globecom2010, IEEE WCMC2010, IEEE Globecom 2011, IEEE ICC'2012 andIEEE ICC'2017. He also acted or still acts as TPC member of different IFIP, ACM or IEEE conferencesand workshops. He was the Chair of IEEE ComSoc IIN Technical Committee, TCIIN (2014–2016). He ispart of the editorial board of IEEE Network Magazine and serves as Guest Editor of premium journals, suchADHOC journal, IEEE Network Magazine, IEEE Access, IEEE Vehicular Technology Magazine, IEEEAHSN TC Newsletter and the French journal REE. He is also a Member of IEEE and the CommunicationsSociety and Expert Senior of the French society SEE (Society of Electricity and Electronics).

Ying Sun received the Ph.D. degree in statistics from Texas A&M, in 2011. She held a two-yearpostdoctoral research position at the Statistical and Applied Mathematical Sciences Institute and theUniversity of Chicago. She was an Assistant Professor with Ohio State University for a year before joiningKAUST, in 2014. At KAUST, she established and leads the Environmental Statistics research group, whichworks on developing statistical models and methods for complex data to address important environmentalproblems. She has made original contributions to environmental statistics, in particular in the areas ofspatiotemporal statistics, functional data analysis, visualization, computational statistics, with anexceptionally broad array of applications. She received two prestigious awards: The Early InvestigatorAward in Environmental Statistics presented by the American Statistical Association and the Abdel El-Shaarawi Young Research Award from The International Environmetrics Society.

Authors and Affiliations

Wu Wang[1] · Fouzi Harrou[2] · Benamar Bouyeddou[3] · Sidi-Mohammed Senouci[4] · Ying Sun[2]

✉ Fouzi Harrou
fouzi.harrou@kaust.edu.sa

Wu Wang
wu.wang@ruc.edu.cn

Benamar Bouyeddou
benamar.bouyeddou@univ-saida.dz

Sidi-Mohammed Senouci
Sidi-Mohammed.Senouci@u-bourgogne.fr

Ying Sun
ying.sun@kaust.edu.sa

[1] Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing 100872, China

[2] CEMSE Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

[3] STIC Lab, Department of Telecommunications, Abou Bekr Belkaid University, Tlemcen, Algeria

[4] DRIVE Laboratory, University of Burgundy, Nevers, France