IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2020, accepted July 15, 2020, date of publication July 28, 2020, date of current version August 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012411

Replay Attack Detection in Smart Cities Using Deep Learning

ASMAA A. ELSAEIDY[1], (Graduate Student Member, IEEE), NISHANT JAGANNATH[1], ADRIAN GARRIDO SANCHIS[2], ABBAS JAMALIPOUR[3], (Fellow, IEEE), AND KUMUDU S. MUNASINGHE[1], (Senior Member, IEEE)

[1]Faculty of Science and Technology, University of Canberra, Canberra, ACT 2601, Australia
[2]School of Science, UNSW Canberra, Canberra, ACT 2601, Australia
[3]School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia

Corresponding author: Asmaa A. Elsaeidy (asmaa.elsaeidy@canberra.edu.au)

ABSTRACT Intrusion detection is an important and challenging problem that has a major impact on quality and reliability of smart city services. To this extent, replay attacks have been one of the most common threats on smart city infrastructure, which compromises authentication in a smart city network. For example, a replay attack may physically damage smart city infrastructure resulting in loss of sensitive data, incurring considerable financial damages. Therefore, towards securing smart cities from reply attacks, intrusion detection systems and frameworks based on deep learning have been proposed in the recent literature. However, the absence of the time dimension of these proposals is a major limitation. Therefore, we have developed a deep learning-based model for replay attack detection in smart cities. The novelty of the proposed methodology resides in the adoption of deep learning based models as an application for detecting replay attacks to improve detection accuracy. The performance of this model is evaluated by applying it to a real life smart city dataset, where replay attacks were simulated. Our results show that the proposed model is capable of distinguishing between normal and attack behaviours with relatively high accuracy. In addition, according to the results, our proposed model outperforms traditional classification and deep learning models. Last but not least, as an additional contribution, this paper presents a real life smart city data set with simulated replay attacks for future research.

INDEX TERMS Smart cities, intrusion detection, replay attack, deep learning, convolutional neural networks.

I. INTRODUCTION

Internet of things (IoTs) is based on the concept of connecting any device to the Internet. This sort of technology has lead to the creation of smart cities [1], in which basic infrastructure components, such as electricity, health, traffic, agriculture and water resources are monitored and controlled through the Internet. It has been shown that this can reduce costs, make life easier and more comfortable [2]. Smart city data have unique characteristics as they are collected from a wide range of different connected smart devices. The data is collected using different network structures and technologies [3], [4].

The integrity, availability and privacy of data plays a critical role in the success of implementing IoT technologies in smart cities [5]. Smart city data play a major role in people's day to day life; for example, people with respiratory conditions may access environmental data (e.g., air pollution and CO_2 levels) or use this to plan their daily activities (e.g., walk to work/school vs. taking the car). Unfortunately, these smart city systems have become valnerable targets for intruders [6]. Such cyber-attacks represent a significant threat to smart cities and the lives of their residents. The two fundamental components for securing smart city infrastructure are through improved authorization and authentication, where replay attacks violate the authentication component. A replay attack targets a system's authentication by hacking its configuration, which generates unreliable and misleading data [7], [8]. There are possible harms that could be caused by such attacks. In smart vehciles, traffic jams and accidents could happen by sending false information between vehicles [9], and the disruption of smart meters, which results in sending incorrect information to the system [10]. These

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

serious effects show the importance of developing an intrusion detection system. The application of machine learning approaches, specifically deep learning, to intrusion detection in smart cities has proven to be efficient at detecting cyber-attacks. However, the absence of a time dimension, which was not considered in these deep learning-based detection approaches is considered a limitation.

In this paper, our proposed model for attack detection is a deep convolutional neural network with four layers, designed to detect replay attacks by considering the time domain in their identification of an attack. The novelty of the proposed methodology resides in adopting deep learning models as an application to detect replay attacks in smart cities to improve detection accuracy. The performance of our proposed methodology is evaluated based on the accuracy of classifying the behaviour of either normal or attacked dataset. In addition, the performance of our proposed methodology is compared to other typical machine learning and deep learning models from the literature. The experimental results showed that our proposed model outperforms traditional classification and deep learning models.

The dataset used to evaluate the proposed model is generated by simulating replay attack over normal generated data from a real smart city platform in the city of Queanbeyan, Australia. This smart city infrastructure is deployed based on a collaboration between the University of Canberra (UC), and the Queanbeyan-Palerang Regional Council (QPRC) under the Commonwealth Government's smart cities and Suburbs Program in 2017. The QPRC invested the resources to turn the Queanbeyan city into a smart city, by installing range of smart sensors and meters to monitor parking areas, lighting, traffic, weather, water river quality and soil moisture and temperature.

The main contributions of the work introduced in this paper can be summarized as below:

- Proposing a deep learning model for replay attack detection in smart cities by treating the smart city data as a multivariate time series alongside attack classification.
- Introducing a smart city benchmark dataset that is generated by simulating replay attacks on top of real smart city data.

The remainder of this paper is organized as follows: Section II discusses the related work. Section III describes the proposed methodology. Section IV includes the description of the smart city platform. Section V describes the experimental setups and how the synthetic replay attack datasets are generated. Section VI explains the results and discussion. Section VII concludes the paper and Section VIII highlights future directions.

II. RELATED WORK

The potential for cyber-attacks to threaten various systems and smart city infrastructures, as well as intrusion detection systems and frameworks, have been disscussed in the literature. Intrusion detection systems that have been proposed recently, in the context of protecting smart city infrastructures, include: a proposed security protocol for replay attack detection in telecare medicine information systems and smart campuses [11]; the proposal of an intrusion detection system based on an anonymous lightweight authentication protocol to authenticate legitimate vehicles based on smart cards [12]; and the payload mutual authentication scheme that is added to the constrained application protocol (COAP) to improve replay attacks detection [13].

The application of machine learning approaches, specifically deep learning, for intrusion detection in smart cities have proven to be efficient at detecting cyber-attacks. A deep dense random neural network-based approach has been applied for online detection of network attacks that could be launched against IoT gateways [14]. However, a simple threshold detector approach is able to achieve the same results. In [15], a distributed deep learning approach was applied for attack detection in IoT and fog networks. This approach showed better performance than shallow learning models in detecting accuracy. An intrusion detection approach was proposed in [16], which applied the stacked deep polynomial network for classifying the dataset into either normal or attack. For selecting the optimal features to be used by the deep model, the spider monkey optimization algorithm was applied. This approach resulted in better performance than typical machine learning approaches.

Deep belief networks were applied in [17] to detect real-world intrusions effectively. Another application of deep belief networks was reported in [18], where a grid search method was applied to obtain the best hyper-parameters for the deep network. The results showed the ability of deep belief networks to detect the attacks effectively. In [19], a deep restricted Boltzmann machines model was combined with threshold detector for distributed Denial of service attack detection. Applying the deep component for extracting high level features from the raw dataset records contributes to the efficiency of discriminating between normal and attack traffic.

As with previous works, which combined different models, deep belief networks were combined with decision tree models for attack detection [20] and deep restricted Boltzmann machines were combined with multi-layer perceptron of different types, in addition to support vector machines and random forest models, for the detection of distributed Denial of service attack [21]. In [20], a deep belief network was used for dimensionality reduction, selecting required features and detecting if there was an attack, while a decision tree model was subsequently applied to classify the attacks and signal alerts. In [21], deep restricted Boltzmann machines were used to learn higher features that are used by the different classification models to discriminate between normal and different types of attacks.

In summary, most deep learning models report efficient performance in attack detection either by using deep models that feature both learning and classification, or combining these deep learning models with other machine learning

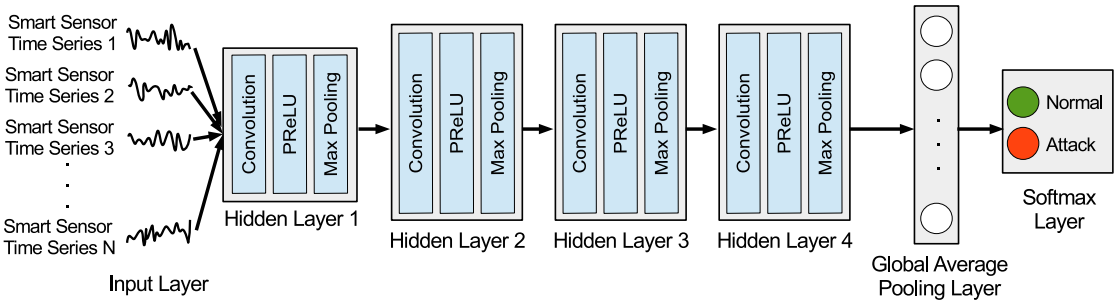


FIGURE 1. The proposed deep learning architecture for replay attack detection.

approaches to perform the classification step. However, there are two main limitations in the previous reported models; the absence of a time dimension, which was not considered in these deep learning-based detection approaches, and the lack of a benchmark dataset for smart cities. The next section will introduce the proposed methodology, which was hypothetically designed to fill these gaps.

III. THE PROPOSED MODEL

In this section, our proposed model for replay attack detection in smart cities is described. The proposed model is a deep convolutional neural network (CNN) architecture, which consists of an input layer, four hidden layers, a global average pooling layer and an output layer, as shown in Fig. 1.

In smart cities, smart sensors and meters are used to collect data streams over time, sending them to the centralized IoT data collection platform. The data collected by the smart meters is the input to the proposed network architecture as time series data. Smart city datasets that will be explained in Section IV have small number of features. A deep-based architecture is preferred over existing typical machine learning approaches. The reason behind this is that the time domain is considered alongside the classification task to discriminate the attack from normal behaviour. Modelling time domain and performing a classification is a complex and challenging task for typical machine learning approaches. Even with small number of features, deep models outperform typical machine learning models such as multi-layer perceptron [21], [22]. Due to the small number of features, the proposed network structure starts with a smaller hidden layer compared to the remaining hidden layers, as shown in Fig. 1. The reason behind this structure is to learn the high level features in an incremental way, which starts from a small number of input features to a large number of learned features through convolution layers. Handling the feature learning this way could help to have stable and robust performance during model training process [23]. The selection of four hidden layers for the proposed architecture is based on an incremental approach, starting with one hidden layer and evaluating the performance. Subsequently, another hidden layer is added, and the performance compared. If the performance has improved, more hidden layers are added until no further improvement is detected. In our trails, adding a fifth layer did not improve the performance, so we settled on a total of four hidden layers.

The input layer has $N \times k$ neurons, where N is the number of variates for input time series and k is the length of each variate. Each hidden layer applies three operations in sequence: convolution, activation function, and max pooling. A convolution operation applies one dimensional filters over the time series data. Convolution for time stamped data is applied as follows:

$$C_t = \mathrm{f}(\omega * X_{t-l/2:t+l/2} + b) \tag{1}$$

where $t \in [1, T]$ and T is the length of one variant time series X. The C_t is the result of convolution operation of applying filter ω of length l to X using the non-linear activation function f() and b is the bias. The parametric rectified linear unit (PReLU) is the activation function used in the proposed architecture. The PReLU is a rectified linear unit (ReLU) activation function with adaptive learned parameters of the rectifies [24]. The ReLU activation function is calculated as follows:

$$\mathrm{f}(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{2}$$

while PReLU is calculated as follows:

$$\mathrm{f}(x) = \begin{cases} \alpha x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{3}$$

where α is a parameter that controls the activation function slope of the negative part. In our proposed architecture, the α parameter of PReLU activation functions is trained jointly with network weights towards having specialized activations that enhance the overall model accuracy.

The last operation applied in each of the hidden layer blocks is the max pooling operation. This local pooling introduces invariance to small disturbances in the activation result by taking the maximum value in a local pooling window [25]. After applying the previously explained operations through the four hidden layers, the outputs are passed to a global average pooling layer. This layer performs global average pooling on the convolution feature map generated from the fourth hidden layer block to produce the final features, which

are fully connected to the output classification layer. Another benefit of applying a global average pooling layer, instead of just a flatten layer, is that it acts as a structural regularizer, which prevents overfitting during training [26]. The features generated from the global average pooling layer are passed to a softmax layer with two neurons representing the normal and attack classes. Even for a binary classification problem softmax function could be used. The advantage of using the softmax activation function over the sigmoid activation function is that the softmax ensures the sum of the outputs is one, while the sigmoid just makes the output between zero and one. Furthermore, in order to increase the probability of a particular class, the model correspondingly decreases the probability of the other class, which is not the case if the sigmoid is used. The softmax outperforms the sigmoid and other activation functions when it is applied on wide range of benchmark datasets in different domains [27]–[30]. This activation function assign a probability distribution for each class where their summation is equal to one. The softmax activation function is calculated as follow:

$$f(x) = \frac{e^x}{\sum_{i=1}^{K} e^{x_i}} \tag{4}$$

where K is the number of output neurons in the output classification layer, and x is the output of one of the output neurons in the output layer.

The proposed CNN architecture is trained using the back-propagation algorithm where the Adam optimizer is used with cross entropy loss function [31]. The Adam algorithm starts by initializing the learning rate, exponential decays for moment estimates and a small constant used for numerical stabilization. The network parameters including filter weights, biases and PReLU activation function coefficients are initialized. The algorithm runs for a certain number of epochs by sampling from the training dataset in a minibatch way. The gradients are then calculated and used to estimate first and second biased moments. Next, the estimated moments are corrected and used to compute the step size, which will be used to update the network parameters. The Adam algorithm has the advantages of being able with sparse gradients and non-stationary objectives, and requires little memory. In the next section, the smart city test platform is explained in details. This platform is used to generate the datasets applied to evaluate the performance of the proposed methodology.

IV. THE SMART CITY TEST PLATFORM

A smart city benchmark dataset is characterized by real time collected periodical data. It is collected based on sensors and a communication network installed to transmit data into the cloud to monitor and control daily operations. Examples of these infrastructures include a deployed smart city infrastructure in city of Aarhus in Denmark [32]; the smart city model introduced by IBM, which collects and monitors data from water resources, traffic, buildings, public safety agents and energy [33]; a smart city application in [34], which deploys

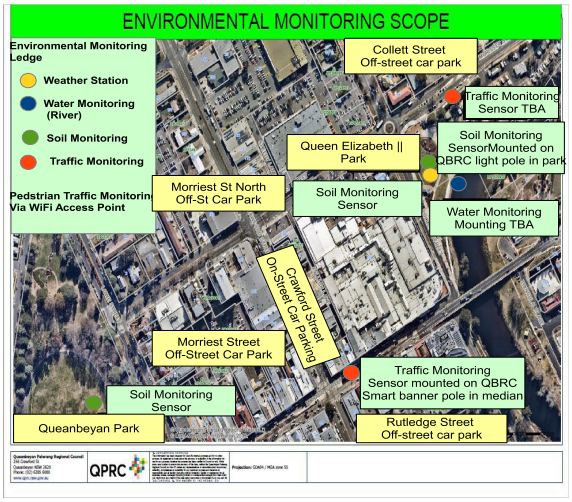


FIGURE 2. The Queanbeyan smart city map.

air pollution sensors to analyze and predict air quality; and the smart city Pulse project, which collects data from air and traffic sensors to monitor air pollution levels [35].

The smart city generated data used for research purposes should be reliable. Since it must satisfy a number of qualification criteria, the installed network infrastructure technology deployed to transmit information, and the consistency and robustness of the cloud repositories, which store and manage the data [36]. Furthermore, the quality of smart sensors and meters used to collect the data should be considered. The dataset used to evaluate the proposed methodology in detecting replay attacks in smart cities is generated from real smart city infrastructure. The IoT Research Group at UC developed this for the City of Queanbeyan under the Commonwealth Government's Smart Cities and Suburbs Program from 2017-19, as shown in Fig. 2.

The environmental monitoring, weather station, and river monitoring nodes use a 4G platform to directly upload the sensor data to the cloud server (Fig. 3). Next sections will explain in detail the deployed nodes for data collection, which will form the baseline for our synthetic dataset to evaluate our proposed model.

A. SOIL MANAGEMENT NODE

A state-of-the-art smart soil management platform is being designed and installed to provide information to guide the management of the city's irrigation system. The sensor board shown in Fig. 4 is designed to automate data recording of soil volumetric water content (VWC) at 30 cm and 60 cm depth, soil temperature at 30 cm and battery levels.

The Therm200 is an analogical sensor probe that can measure temperatures from −40°C to 85°C. It provides a lineal response to the temperature where the voltage for −40°C is 0V and the voltage for 85°C is 3V [37].

The echo EC-5 is an analogical sensor that measures volumetric water content (VWC) at a frequency of 70 MHz. Volumetric water content, which is the ratio of water vol-

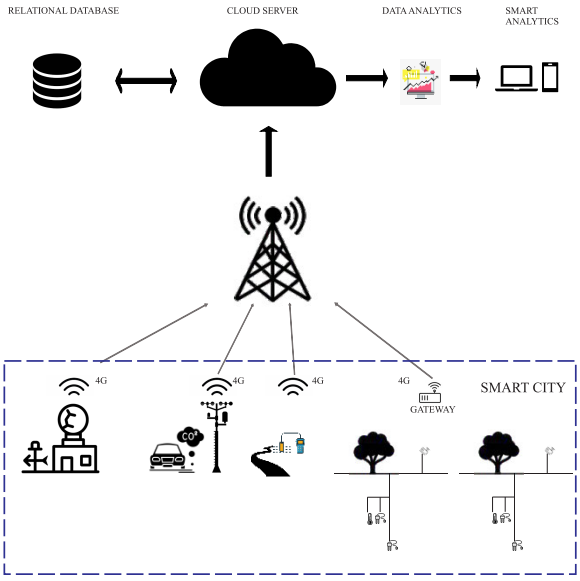


FIGURE 3. Queanbeyan smart city platform architecture.

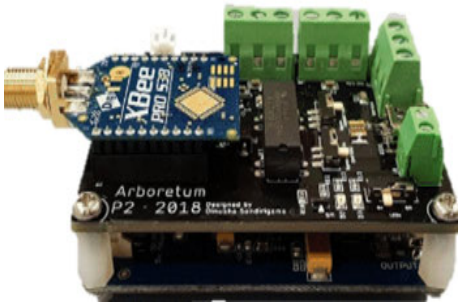


FIGURE 4. Custom built sensor board node.

ume to soil volume, is widely used in agriculture. In this work the following calibration equation was used for each EC-5 sensor:

$$\mathrm{VWC} = 0.00119V_{\mathrm{out}} - 0.401 \tag{5}$$

where VWC is the volumetric soil water content in m^3/m^3 and V_{out} is the sensor output when excited at 2500 mV [38].

B. ENVIRONMENTAL MONITORING NODE

To assess the impact of traffic on the city environment, two environmental monitoring nodes have been installed in streets with a high traffic volume. The sensor board is designed to automate continuous data recording of noise levels and CO_2 concentration. Environmental noise levels are measured with an outdoor omnidirectional microphone that operates across the 10 Hz to 20 kHz frequency range. Measurements of environmental noise are recorded as decibels (dBA) with a range of 50 dBA to 100 dBA. Carbon dioxide concentration, expressed as parts per million (ppm), provides an early warning of rising CO_2 levels due to the traffic. To measure these levels, a Non-Dispersive Infrared (NDIR) gas sensor (NE20-CO2P-NCVSP), which has a nominal range of 0 to 5000 ppm with a resolution of 25 ppm was used [39].

C. WEATHER STATION NODE

The weather station node provides data on air temperature, humidity, atmospheric pressure, solar radiation, ultraviolet (UV) radiation, wind speed, wind direction, and rain. Temperature, humidity, and pressure are measured with a combined digital sensor board, the BME280. This board operates in a range of 30 to 110 kilopascal (kPa) for the atmospheric pressure sensor, 0 to 100% of relative humidity for the humidity sensor and -40°C to 85°C for the temperature sensor. Solar radiation is measured with the SQ-110 sensor, which operates in the spectral range of 410 to 655 nanometers (nm). This analogical sensor provides an output voltage propotional to the intensity of the light. The SQ-100 measures UV radiation in the spectral range of 250 to 400 nm. The sensor provides an output voltage proportional to the light intensity in the ultraviolet range [40]. The WS-3000 station integrates an anemometer, a pluviometer, and a wind vane. The anemometer measures wind speed with a sensitivity of 2.4 km/h per turn in a range of 0 to 240 km/h. The wind vane measures wind direction with a resolution of 22.5°. The pluviometer comprises a 0.28 mm bucket that triggers the interruption of the microcontroller once it is filled. After that it is emptied automatically. The amount of rain is calculated from the number of bucket emptying events [40].

D. RIVER MONITORING NODE

This node continuously monitors number of water quality parameters including electrical conductivity, water acidity (pH), water temperature, and turbidity. Water temperature is a key in the health of fish and other temperature-sensitive aquatic microorganisms. Water temperature is monitored by the Pt-1000 sensor, which measures changes in conductivity induced by temperature differences ranging from 0°C to 100°C [41]. Electrical conductivity (EC) measures concentration of dissolved ions or salts in water. A rise in EC indicates an increase in pollutants, fertilizers, and other ions in rivers [42]. A magnetic field between the two electrodes of the conductivity sensor provides EC readings in Siemens per centimeter (S/cm) from the Queanbeyan River. Water pH is a measure of accidity, ranging from 0 and 14, where most of the rivers and lakes have a pH range of 6.5 to 8.5. The biological availability of nutrients in the water or chemical changes of water can be detected through pH variation. The gravity pH is an analogical sensor that provides an output voltage proportional to the pH of the water. For example at pH 7 this sensor provides a 2.048V reading [41]. Turbidity is an indicator of the clarity of the water. An increase in water turbidity reveals the presence of pollutants or sediments such as waste water discharge, soil erosion, agricultural pesticides, and algae growth. The turbidity probe is an infrared (IR) optical sensor that measures the light scattered at a 90° angle from an IR light. This sensor reports the measurements in

Nephelometric Turbidity Units (NTU) with an accuracy of 1 NTU for a range from 0 to 4000 NTU.

E. NODE RADIO INTERFACES

The soil management node integrates the XBEE Pro line, the Series 3B (900HP) that utilizes the protocol DigiMesh. This protocol supports a synchronous-sleeping mesh network. It can provide synchronous readings and dynamically route and exchange data between other sensor nodes, and relay these data to the gateway in an application programming interface (API) frame. The XBEE Pro S3B has its own internal 4-channel 10-bit Analog-to-Digital Chip (ADC) [43]. Both soil sensor nodes in the network are synchronized, meaning the ''wake up'' and ''sleep'' cycle sequences follow the same time pattern to reduce sensor node power consumption [44]. A Raspberry Pi and the XBEE coordinator constitute the gateway, which is responsible for establishing the network and receiving data from the two soil sensor nodes. Data from nodes is then uploaded to the cloud server using a 4G modem, stored in a relational database. The environmental monitoring, weather station, and river monitoring nodes are integrated through a 4G module that supports LTE and HSPA+, thus enabling high-speed connectivity to the Amazon Web Services (AWS) cloud servers [39]. It also incorporates protocols such as SSL for secure connection with the cloud server and FTP for managing the files on the nodes. The nodes are integrated with dual antenna equipped with MIMO technology that provides excellent signal strength for maximum performance.

F. DASHBOARDS

The API is built using a node express server package and is hosted on AWS. When the data is uploaded by the gateway, the server is notified and stores the data in a relational database. The platform's API uses tier level access control, allowing only the administrator and other pre-defined high-level permission clients to modify the node's data and request API keys to upload and retrieve data from the server. The relational database is the master database that keeps the records of all sensor data, battery levels, timestamps, tier-level of clients, and other relevant information with regards to the sensor node. A real-time database is used to authenticate clients and update the dashboard in real-time. The dashboard has been designed and built using React JS framework to graphically present data from all nodes. The dashboard can be used to modify node configurations, add new nodes into the network, and change sampling frequency intervals depending on tier-level of the client.

V. EXPERIMENTAL EVALUATION

A. SYNTHETIC REPLAY ATTACK DATASET

In this section, we describe the generation of synthetic replay attack datasets based on the normal behaviour of the smart city test platform. These datasets will be used for evaluating the proposed model's performance. Smart city benchmark datasets that contain actual and attack data cannot be easily found in the literature. Moreover, when such datasets are available, permission to experimentally run attacks is not easily obtained, the process is costly, and there are complex security and privacy issues. Consequently, most of the work reported in the literature is based on artificially generated datasets [36], [45]. In this paper, we have the advantage of having access to a real dataset. In a real-world scenario, replay attacks violate data authentication in order to make physical damage. This is accomplished by generating misleading data based on the normal behaviour. Therefore, in this paper replay attack behaviour is mimicked to generate the synthetic datasets based on the estimated probability distributions of the normal data [9].

Two platforms are selected, generating two different datasets that have been used one by one for experimental evaluation; the soil management node and environmental monitoring node, as described in Sections IV-A and IV-B, respectively. These two nodes were selected for the reliability of the generated data and the low error rate associated with reading and uploading the data to the cloud. The time series data collected from the soil management node range from 09/07/2018 - 00:37 to 01/05/2019 - 00:37 (dd/mm/yyyy - hh:mm format) with readings per minute. This soil node dataset has five features: soil temperature, soil moisture at 30 cm depth, soil moisture at 60cm depth, battery levels and samples, which indicates the number of packets reached the gateway successfully. The time series data collected from the environmental monitoring node range from 03/09/2019 - 11:26 to 04/03/2020 - 12:23 (dd/mm/yyyy - hh:mm format) with readings per minute. This environmental monitoring node dataset has five features: noise amplitude, air temperature, humidity, air pressure, and CO_2 levels.

The datasets were cleaned by removing all NULL or zero values, as these indicate an error in sensor reading. The probability distributions for each feature were then estimated and used to generate the synthetic replay attack data. All features for the soil management dataset were drawn from a normal distribution. For the environmental dataset, noise amplitude and CO_2 levels were drawn from a normal distribution, air temperature was drawn from a Weibull max distribution, humidity was drawn from Weibull min distribution and air pressure was drawn from a genextreme distribution. Given these estimated distributions, we started from the start date-time and incremented the time by minutes. If the next date-time does exist in the normal dataset, we skipped it to the next one. If the current date-time does not exist in the normal dataset, it was considered as a candidate entry for an attack filed. Based on a comparison between a random number generated and a pre-defined threshold, it was determined whether or not this candidate will be added as an attack field to the dataset. This threshold value controls the percentage of attack instances to the number of normal instances. Figs. 5 and 7 show the time series observations and the probability distributions for the soil management node synthetic dataset features, respectively. While Figs. 6 and 8

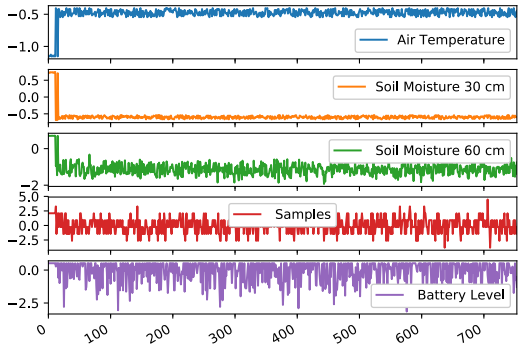


FIGURE 5. A sample of a multivariate time series from the soil management node dataset after generating synthetic replay attacks with 5 sub-plots corresponding to the dataset features: air temperature, soil moisture 30 cm, soil moisture 60 cm, samples and battery level.

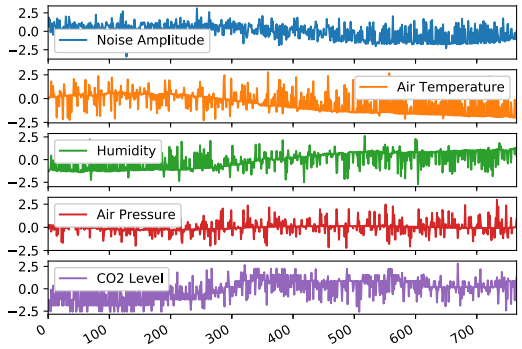


FIGURE 6. A sample of a multivariate time series from the environmental monitoring node dataset after generating synthetic replay attacks with 5 sub-plots corresponding to the dataset features: noise, air temperature, humidity, air pressure and CO_2 level.

show the time series observations and the probability distributions for the environmental node synthetic dataset features, respectively.

B. EXPERIMENTAL SETUPS

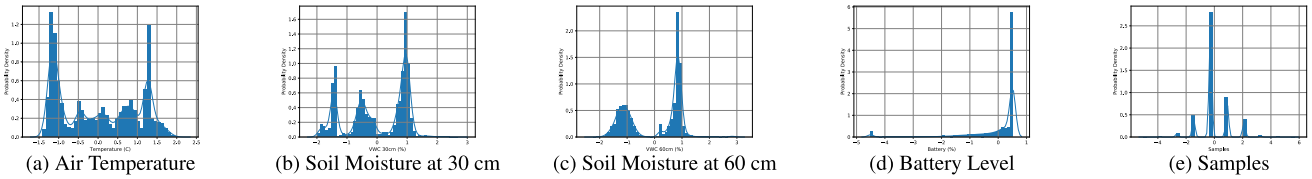
For comparative evaluation, we selected five state-of-the-art methods including: multi-layer perceptron (MLP) [46], deep multi-layer perceptron (DMLP) [47], deep residual network (DRN) [47], time LeNet (TLN) model [48], and echo state network for classification (ESNC) [49]. The MLP is a fully connected feedforward model with an input layer, one hidden layer an and output layer, where dropout operations are applied on hidden and output layers. The DMLP model is a fully connected feed forward neural network with an input layer, three hidden layers and an output layer, where the hidden and output layers are preceded by a dropout operation. The DRN is a deep CNN model with 9 convolutional layers followed by a batch normalization operation for each layer, and a global average pooling layer, which is fully connected to the output layer. The TLN model has two convolutional layers with a max pooling operation, followed by a fully connected layer that is connected to the output layer.

The ESNC model applies typical echo state network model to map the input features into a higher dimension, and the results of the dynamical reservoir are passed into the fully connected layer that is connected to the output layer. The ESN model is a special type of recurrent neural networks with a straightforward design and training procedure. It overcomes the issues of exploding and vanishing gradients when training typical recurrent neural networks (RNNs) using backpropagation algorithm [50]. Typical ESN architecture contains an input layer, a hidden layer called reservoir, and an output layer. In ESN model, only output weights are trained in strightforward way, while the remaining weights are fixed. The weights between the input layer and dynamical reservoir are initialized with random values within a predefined scale. The dynamical reservoir weights are designed to maintain an echo state property by generating initial random sparse weights within the $[-1, 1]$ range. Finally, the reservoir weights matrix is multiplied by a chosen value called spectral radius [51]. The ReLU activation function is used for all these five models' hidden layers and the softmax activation function is used for the output layer. The Adam algorithm is used as the optimization algorithm with cross entropy as the loss function.

The generated synthetic datasets were normalized to have zero mean and unit variance. In addition, the outliers were detected and removed. The soil management dataset had a total of 89,566 instances, while the environmental dataset had 178,211. Each dataset was divided into training and testing subsets where 20% of the dataset was used for the testing. The training subset was further divided into train and validation subsets with 20% for the validation subset. This training/validation split was used to search for the best combination of hyper-parameters for each model, including the proposed model. Once the best parameters were found for each model, we trained each model using the whole training subset and evaluated its performance on the testing subset. This evaluation process using the training/testing split was applied with 30 runs with different random seeds and 1000 epochs. The reported result for each model was the average performance over the 30 random runs.

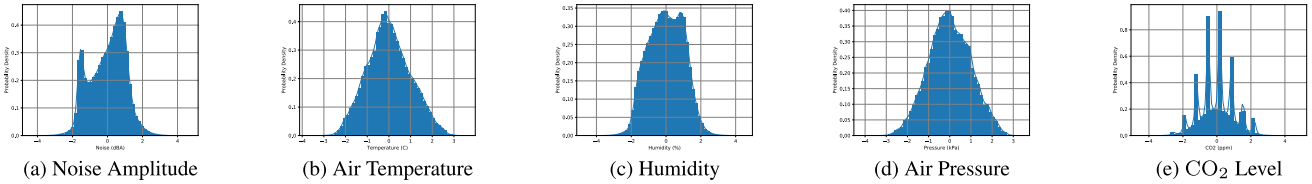
To find the best hyper-parameters for each model, a grid search method was applied [52]. For each model we identified a range for each model hyper-parameter and searched for the combination that provided the greatest accuracy on the validation dataset. For all models, the range for learning rate was [1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001] and for batch size the range was [8, 15, 32, 64, 128]. For the proposed model, the size of the filters for convolution layers range was [8,16,32,64]. For MLP and DMLP models, hidden layer size range was [250, 500, 750], hidden layer dropout range was [0.1, 0.2, 0.3, 0.4] and output layer dropout range was [0.2, 0.3, 0.4, 0.5]. For the DRN model, the size of the filters for convolution layers range was [32, 64, 128, 256]. For the TLN model, size of the filters for convolution layers range was [5, 10, 15, 20, 25, 30] and flatten layer size range was [250, 500, 750].

For the ESNC model, reservoir size range was [50, 100, 150, 200], spectral radius range was [0.3, 0.59, 0.6, 0.85],



(a) Air Temperature (b) Soil Moisture at 30 cm (c) Soil Moisture at 60 cm (d) Battery Level (e) Samples

FIGURE 7. Soil management node dataset features probability distributions after adding the synthetic replay attacks. All features are drawn from normal probability distribution, which provides the best fitting.



(a) Noise Amplitude (b) Air Temperature (c) Humidity (d) Air Pressure (e) CO_2 Level

FIGURE 8. Environmental monitoring node dataset features probability distributions after applying the synthetic replay attacks. Noise amplitude and CO_2 level are drawn from normal distribution, air temperature is drawn from weibull max distribution, humidity is drawn from weibull min distribution and air pressure is drawn from genextreme distribution, where all these distributions provide the best fit.

reservoir connectivity range was [0.25, 0.5, 0.75, 0.9], input weights scale range was [0.1, 0.3, 0.5, 0.8], and hidden layer connected to the output layer range was [250, 500, 750]. After applying the grid search method for tuning the models' hyper-parameters, the best combinations for each model found were as follows: for the proposed model, learning rate was 0.001, batch size was 16, 16 filters for first convolution layer, and 32 filters for the the remaining three convolution layers; for MLP, learning rate was 0.001, number hidden units for the hidden layer was 250, dropout rate for the hidden layer was 0.1 and dropout rate for the output layer was 0.2; for DMPL, the learning rate was 1.0, number of hidden units for each of the three hidden layers was 750, dropout rate for first hidden layer was 0.3, dropout rate for the second and third hidden layers was 0.4 and dropout rate for the output layer was 0.5; for DRN, the learning rate was 0.01 and the number of filters for the 11 convolution layers was 32; for TLN, the learning rate was 0.001, the number of filters for the first layer was 10, the number of filters for the second convolution layer was 25 and the number of units for the dense layer was 500; and for ENSC, the learning rate was 0.01, the reservoir size was 100, the reservoir spectral radius was 0.6, the reservoir connectivity was 0.25, the input weights scale was 0.1, and number of units for the fully connected layer was 500.

All models are all implemented using Python 3.7 and Keras library with Tensorflow framework as the backend [53]. The experiments are run on an Intel(R) Xeon(R) Silver 4116 CPU 2.10GHz 2.10GHz (2 processors), with 128 GB of RAM and Windows 10 (64 bit) machine.

VI. RESULTS

This section presents the results obtained from applying the proposed methodology and the other deep models for replay attack detection using the synthetic datasets. First, we visualized the average accuracy measure averaged over the 30 runs,

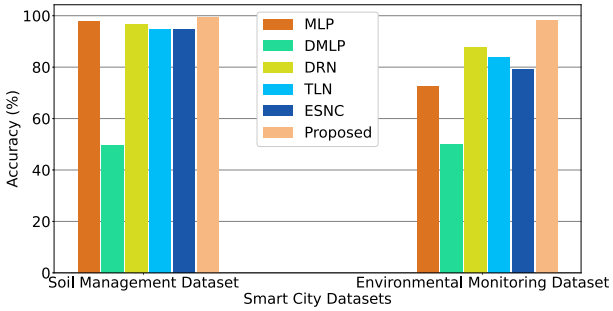


FIGURE 9. Average accuracy measure for proposed model compared with state-of-the-art models in both soil and environmental datasets.

as shown in Fig. 9. The mean and standard deviation for accuracy, false positive rate, sensitivity, specificity, and precision measures were calculated for all learning models averaged over the 30 runs for each dataset, as shown in Tables 1 and 2. These measures used were calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{6}$$

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

$$\text{Sensetivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

where FPR is false positive rate, and TP, TN, FP and FN are the true positives, true negatives, false positives, and false negatives, respectively.

Each one of these five measures quantify a different aspect of the models performance. Accuracy measures show how accurate the model is by calculating the ratio of number of correct predictions to the total number of input samples.

TABLE 1. Reported performance measures (accuracy, false positive rate, sensitivity, specificity and precision) for learned models compared with proposed model for soil management node dataset. The mean and standard deviation is calculated for each measure (mean±std). Bold indicates the best reported model performance for each measure.

	Accuracy %	False Positive Rate %	Sensitivity %	Specificity %	Precision %
MLP	97.886(±6.436)	7.719(±6.322)	97.886(±6.322)	97.886(±6.322)	97.886(±6.322)
DMLP	49.452(±0.421)	50.1167(±0.1142)	49.451(±0.411)	49.451(±0.411)	49.451(±0.411)
DRN	96.773(±3.006)	3.5412(±2.7216)	96.773(±2.956)	96.773(±2.956)	96.773(±2.956)
TLN	94.876(±4.171)	5.1875(±3.999)	94.876(±4.101)	94.876(±4.101)	94.876(±4.101)
ESNC	94.894(±0.026)	5.2114(±3.234)	94.854(±0.126)	94.854(±0.126)	94.854(±0.126)
Proposed	**99.301(±2.274)**	**2.0749(±2.006)**	**99.3013(±2.2361)**	**99.3013(±2.2361)**	**99.3013(±2.2361)**

The FPR measure is the ratio of negative predictions that are incorrectly identified as positive predictions. Sensitivity measures the portion of actual positives that are correctly predicted. Specificity measures the portion of actual negatives that are correctly predicted. Precision measures the classifier exactness by calculating the ratio of positive predictions to the total number of positive classes.

The Wilcoxon signed-rank test [54] is applied with alpha 0.05 correction to determine whether there is a statistically significant difference between the learning models, as shown in Table 3 for the soil dataset and in Table 4 for the environmental dataset, where the accuracy performance measure is used. The null hypothesis (H0) assumes that there is no significant difference between two machine learning algorithms, so both performance samples are drawn from the same distribution. In Tables 3 and 4, the cells where the H0 is rejected, shown in bold, indicate a significant difference between the performance of two compared algorithms. In all other cells, the null hypothesis was accepted. In addition, we plot the correlation between the testing accuracy measures for the learning models for both smart soil dataset (Fig. 10) and smart traffic dataset (Fig. 11).

The reported performance accuracy in Fig. 9, and Tables 1 and 2 show that the proposed model is the best model with the highest accuracy values in both datasets. There is a significant difference in performance between the proposed model and other models in both datasets, as shown in Tables 3 and 4. However, the performance reported for the proposed model in the soil dataset is better than the performance reported in environmental dataset. This could be related to the probability distributions for the features of environmental dataset, which contains three features out of five drawn from generalized extreme value distribution. The DMLP model reports the worst performance over all models in both datasets, as shown in Fig. 9 and Tables 1 and 2. This was not expected since this model has more hidden layers than the MLP model. However, it seems that increasing the hidden layers in the typical MLP model overfits the training dataset and reports poor results in the testing subset. When comparing the other CNN models, DRN and TLN, DRN outperforms the TLN model in both datasets. The DRN model has 11 layers and global average pooling layers, while the TLN model has only two layers with smaller filters and a flatten layer. The global average pooling layer acts as a generalized factor to prevent overfitting, and the large number

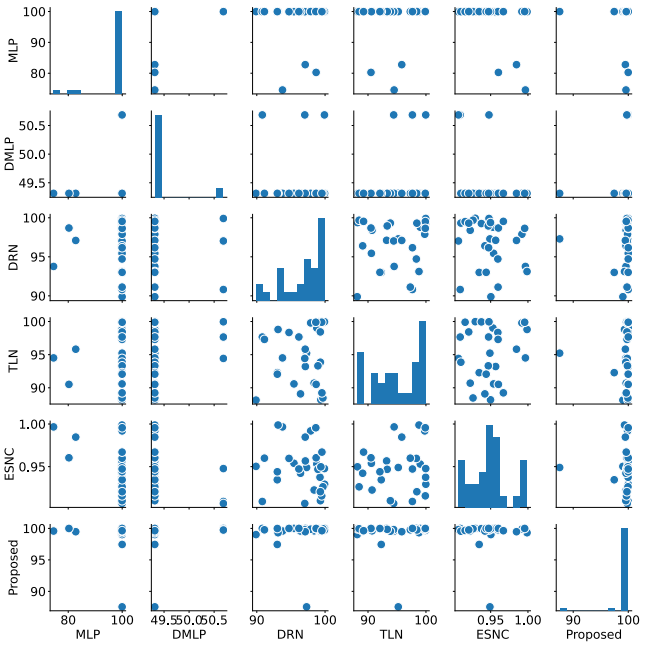


FIGURE 10. Correlation between the learned models and the proposed model based on the average accuracy measure for the soil management node dataset.

of layers in DRN compared to just two layers with small filters in TLN could be the reasons for DRN performing better than TLN. Both DRN and the proposed model have a global average pooling layer, but the proposed model has four layers with gradual increase in number of filters compared to fixed filter sizes over the 11 layers in the DRN model. This large number of layers in DRN could affect the performance of the DRN model by increasing the overfit factor compared to the proposed model. The accuracy of the ESNC model is very close to the TLN model, but it is still lower than the DRN and MLP models. The ESN model is a type of recurrent neural network that learns higher complex time-domain relations between input features by mapping the input time series into the reservoir sparse dimension. A possible explanation for why typical MLP outperforms the ESNC model is the large difference between the number of features and the reservoir size. This difference could have an opposite affect by mapping the input features to a complex high dimensional space, which makes the classification task hard. Same behaviour for the ESNC performance was observed for the environmental node dataset, where the CNN models outperforms the ESNC

TABLE 2. Reported performance measures (accuracy, false positive rate, sensitivity, specificity and precision) for learned models compared with proposed model for environmental node dataset. The mean and standard deviation is calculated for each measure (mean±std). Bold indicates the best reported model performance for each measure.

	Accuracy %	False Positive Rate %	Sensitivity %	Specificity %	Precision %
MLP	72.652(±0.792)	35.2208(±0.4933)	72.5901(±0.778)	72.5901(±0.778)	72.5901(±0.778)
DMLP	49.929(±0.353)	50.032(±0.078)	49.9239(±0.3724)	49.9239(±0.3724)	49.9239(±0.3724)
DRN	87.554(±3.008)	16.3461(±2.889)	87.474(±2.946)	87.474(±2.946)	87.474(±2.946)
TLN	83.693(±2.621)	17.6315(±2.4491)	83.6152(±2.5733)	83.6152(±2.5733)	83.6152(±2.5733)
ESNC	79.218(±0.982)	20.2208(±0.4933)	79.2389(±0.8321)	79.2389(±0.8321)	79.2389(±0.8321)
Proposed	**98.0416(±0.651)**	**5.6633(±0.998)**	**97.6422(±0.811)**	**97.6422(±0.811)**	**97.6422(±0.811)**

TABLE 3. The results of Wilcoxon tests between pairwise combinations of the models for the soil management node dataset, represented by W statistic and the alpha value: W statistic(p-value). Cells in bold indicate that the null hypothesis H0 is rejected. Accuracy measure is used for the statistical analysis.

	MLP	DMLP	DRN	TLN	ESNC
MLP					
DMLP	0.0(0.0)				
DRN	**87.0(0.003)**	**0.0(0.0)**			
TLN	**85.0(0.002)**	**0.0(0.0)**	144.0(0.069)		
ESNC	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	
Proposed	**142.0(0.103)**	**0.0(0.0)**	**34.0(0.0)**	**32.0(0.0)**	**0.0(0.0)**

TABLE 4. The results of Wilcoxon tests between pairwise combinations of the models for the environmental monitoring node dataset, represented by W statistic and the alpha value: W statistic(p-value). Cells in bold indicate that the null hypothesis H0 is rejected. Accuracy measure is used for the statistical analysis.

	MLP	DMLP	DRN	TLN	ESNC
MLP					
DMLP	**0.0(0.0)**				
DRN	**0.0(0.0)**	**0.0(0.0)**			
TLN	**0.0(0.0)**	**0.0(0.0)**	**55.0(0.0)**		
ESNC	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	
Proposed	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**	**0.0(0.0)**

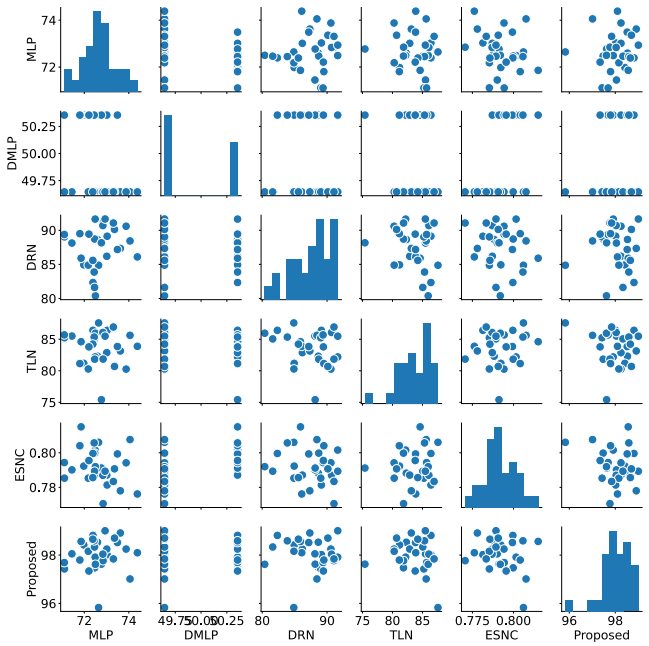


FIGURE 11. Correlation between the learned models and the proposed model based on the average accuracy measure for the environmental monitoring node dataset.

model. However, in this dataset the ESNC outperforms the accuracy reported by typical MLP model. As mentioned previously, this could be related to the complex probability distributions for the environmental dataset.

At the level of the datasets, the MLP model performs better in the soil dataset than the other models, except for the proposed model, as shown in Tables 1 and 2. However, MLP performance is still close to both the DRN and TLN models. The MLP is a simple model with just one hidden layer, which results in fewer hyper-parameters to tune, and this reduces the chances of overfitting. Another reason why the MLP model performs better is that all features in the soil dataset are drawn from a normal probability distribution, which makes the task of modeling this dataset easier. In the environmental dataset, the MLP performed less well than the DRN and TLN models, while even DRN, TLN and ESNC model's performance dropped compared to their performance in the soil dataset. As mentioned previously, the environmental dataset has three features out of five drawn from a generalized extreme value distribution, which is a more complex distribution that combines the Gumbel, Frechet and Weibull maximum extreme value distributions [55]. Given the nature of this dataset, MLP in this case cannot not handle this complexity with one hidden layer. This is why the DRN, TLN, and ESNC models outperform MLP in the environmental dataset compared to the reported performances in the soil dataset. The reported performance measures (Tables 1 and 2) showed a stability in the performance of all models. The differences were minor between the accuracy and other measures that concentrates in measuring portions of positive and negative predictions to the total number of positive and negative targets. No correlation was detected between the testing accuracy reported by the learning models in both datasets, as shown in Figs. 10 and 11; however, in the environmental dataset, most of the correlated plots were scattered to indicate no sign of correlation at all.

VII. CONCLUSION

The proposed model introduced in this paper for replay attack detection is a part of our ongoing research toward securing smart city infrastructure and services. The proposed model is a deep convolutional neural network with four hidden layers, a global average pooling layer, and an output layer.

Previous works have evaluated intrusion detection models by creating test beds in small scale and controlled conditions. The performance of the proposed methodology in this paper was evaluated by synthetically generating replay attacks on real-life normal behaviour generated from Queanbeyan smart city infrastructure in Australia. Two data nodes, soil management and environmental monitoring, were selected to be used to evaluate the performance of the proposed methodology. These datasets are multivariate time series with replay and normal behaviours. The performance of the proposed methodology was compared with typical neural network models, typical convolutional neural networks models, and a well known recurrent neural network model called echo state networks. The experimental results showed that our proposed model outperforms all other models with high accuracy. However, modelling the environmental monitoring node dataset was more challenging than the soil management node dataset as the former is drawn from more complex probability distributions. A typical neural network model was able to outperform other convolutional neural networks in the soil management dataset only. In addition, the performance of echo state network was expected to perform better than observed here, and it appears that applying this type of recurrent neural network for time series classification is not a straightforward task.

VIII. FUTURE WORK

Modelling smart city data is a challenging task and different aspects need to be considered such as data complexity, time domain, and heterogeneity in the data types generated from different sensors and meters. The work in this paper provides valuable insights into how deep learning models might contribute to the detection of replay attacks on smart cities. The work proposed in this paper will be extended in the following future directions:

- Including more node data, either in separate evaluation or combined with large complex datasets.
- Combining the deep learning model into an ensemble learning paradigm aimed at enhancing the overall detection performance.
- Integrating applied machine learning approaches in security frameworks for reliable attack detection in smart cities.
- Generating real replay attacks in real smart city infrastructure rather than generating synthetic attacks.

REFERENCES

[1] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, and P. Siano, "Iot-based smart cities: A survey," in *Proc. 16th Int. Environ. Electr. Eng. Conf. (EEEIC)*, Florence, Italy, Jun. 2016, pp. 1–6.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[3] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of Things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[4] M. Banerjee, J. Lee, and K.-K.-R. Choo, "A blockchain future for Internet of Things security: A position paper," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 149–160, Aug. 2018.

[5] Z. A. Baig, P. Szewczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, K. Sansurooah, N. Syed, and M. Peacock, "Future challenges for smart cities: Cyber-security and digital forensics," *Digit. Invest.*, vol. 22, pp. 3–13, Sep. 2017.

[6] E. O'Dwyer, I. Pan, S. Acha, and N. Shah, "Smart energy systems for sustainable smart cities: Current developments, trends and future directions," *Appl. Energy*, vol. 237, pp. 581–597, Mar. 2019.

[7] Y. Lu and L. D. Xu, "Internet of Things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2103–2115, Apr. 2019.

[8] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. 6th Mobile Comput. Netw. Annu. Int. Conf.*, Boston MA, USA, Aug. 2000, pp. 275–283.

[9] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[10] T.-T. Tran, O.-S. Shin, and J.-H. Lee, "Detection of replay attacks in smart grid systems," in *Proc. Int. Conf. Comput., Manage. Telecommun. (ComManTel)*, Jan. 2013, pp. 298–302.

[11] M. Safkhani and A. Vasilakos, "A new secure authentication protocol for telecare medicine information system and smart campus," *IEEE Access*, vol. 7, pp. 23514–23526, 2019.

[12] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10626–10636, Dec. 2017.

[13] M. A. Jan, F. Khan, M. Alam, and M. Usman, "A payload-based mutual authentication scheme for Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 1028–1039, Mar. 2019.

[14] O. Brun, Y. Yin, and E. Gelenbe, "Deep learning with dense random neural network for detecting attacks against IoT-connected home environments," *Procedia Comput. Sci.*, vol. 134, pp. 458–463, Oct. 2018.

[15] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.

[16] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning—Based intrusion detection framework for securing IoT," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 11, p. 3803, Nov. 2019.

[17] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, p. 1977, Apr. 2019.

[18] B. A. Tama and K.-H. Rhee, "Attack classification analysis of IoT network via deep learning approach," *Res. Briefs Inf. Commun. Technol. Evol.*, vol. 3, pp. 1–9, Nov. 2017.

[19] A. Elsaeidy, I. Elgendi, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "A smart city cyber security platform for narrowband networks," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6.

[20] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101842.

[21] A. Elsaeidy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "Intrusion detection in smart cities using restricted Boltzmann machines," *J. Netw. Comput. Appl.*, vol. 135, pp. 76–83, Jun. 2019.

[22] I. Kok, M. U. Simsek, and S. Ozdemir, "A deep learning model for air quality prediction in smart cities," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1983–1990.

[23] S. Han, Z. Meng, Z. Li, J. O'Reilly, J. Cai, X. Wang, and Y. Tong, "Optimizing filter size in convolutional neural networks for facial action unit recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5070–5078.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[25] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[27] P. T. Krishnan, P. Balasubramanian, and S. Umapathy, "Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network," *Phys. Eng. Sci. Med.*, vol. 43, no. 2, pp. 505–515, Feb. 2020.
[28] A. Arami, A. Poulakakis-Daktylidis, Y. F. Tai, and E. Burdet, "Prediction of gait freezing in parkinsonian patients: A binary classification augmented with time series prediction," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1909–1919, Sep. 2019.
[29] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[30] R. Memisevic, C. Zach, M. Pollefeys, and G. E. Hinton, "Gated softmax classification," in *Proc. Adv. Neural Info. Process. Syst. Conf.*, Vancouver, BC, Canada, Dec. 2010, pp. 1603–1611.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Learn. Represent. Conf.*, San Diego, CA, USA, May 2015, pp. 1–15.
[32] M. I. Ali, F. Gao, and A. Mileo, "Citybench: A configurable benchmark to evaluate RSP engines using smart city datasets," in *Proc. 14th Int. Semantic Web Conf. (ISWC)*, Bethlehem, PA, USA, Oct. 2015, pp. 374–389.
[33] S. Schaefer, "Smarter cities series: A foundation for understanding IBM smarter cities," IBM Redbooks, Armonk, NY, USA, Tech. Rep., 2011.
[34] A. R. Honarvar and A. Sami, "Towards sustainable smart city by particulate matter prediction using urban big data, excluding expensive air pollution infrastructures," *Big Data Res.*, vol. 17, pp. 56–65, Sep. 2019.
[35] P. Barnaghi, R. Tönjes, J. Höller, M. Hauswirth, A. Sheth, and P. Anantharam, "CityPulse: Real-time IoT stream processing and large-scale data analytics for smart city applications," in *Proc. Eur. Semantic Web Conf.*, Trentino, Italy, Oct. 2014, pp. 1–2.
[36] S. Mallapuram, N. Ngwum, F. Yuan, C. Lu, and W. Yu, "Smart city: The state of the art, datasets, and evaluation platforms," in *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, May 2017, pp. 447–452.
[37] J. A. Pardiñas-Mir, L. Rizo-Dominguez, and L. E. Pérez-Bernal, "A wireless sensor network as a living lab for the development of solutions for IoT and smart cities," in *Proc. 15th Int. Joint Conf. e-Bus. Telecommun.*, 2018, pp. 469–474.
[38] *EC-5 Sensor User Manual*, Decagon Devices Inc., Pullman, WA, USA, 2019.
[39] *4G Networking Guide*, Libelium Comunicaciones Distribuidas SL, Zaragoza, Spain, 2020, p. 94.
[40] *Smart Agricult. 3.0. Tech. Guide*, Libelium Comunicaciones Distribuidas SL, Zaragoza, Spain, 2020, p. 88.
[41] *Smart Water Tech. Guide*, Libelium Comunicaciones Distribuidas SL, Zaragoza, Spain, 2019, p. 94.
[42] R. Das, N. R. Samal, P. K. Roy, and D. Mitra, "Role of electrical conductivity as an indicator of pollution in shallow lakes," *Asian J. Water Environ. Pollut.*, vol. 3, no. 1, pp. 143–146, Dec. 2006.
[43] *User Guide-XBee-PRO 900HP/XSC RF Modules*, Digi. International, Hopkins, Minnesota, 2017.
[44] C. Bell, *Beginning Sensor Networks With Arduino and Raspberry Pi*. New York, NY, USA: Spinger/Sci-Tech/Trade, 2013.
[45] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.
[46] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, nos. 14–15, pp. 2627–2636, Aug. 1998.
[47] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.
[48] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in *Proc. Adv. Anal. Learn. Temporal Data Workshop*, Porto, Portugal, Sep. 2016, pp. 1–9.
[49] Q. Ma, L. Shen, W. Chen, J. Wang, J. Wei, and Z. Yu, "Functional echo state network for time series classification," *Inf. Sci.*, vol. 373, pp. 1–20, Dec. 2016.
[50] H. Jaeger, *Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the' Echo State Network' Approach*, GMD-Forschungszentrum Informationstechnik Bonn, Sankt Augustin, Germany, 2002.
[51] H. Abdelbari and K. Shafi, "Learning structures of conceptual models from observed dynamics using evolutionary echo state networks," *J. Artif. Intell. Soft Comput. Res.*, vol. 8, no. 2, pp. 133–154, Apr. 2018.
[52] R. Liu, E. Liu, J. Yang, M. Li, and F. Wang, "Optimizing the hyperparameters for SVM by combining evolution strategies with a grid search," in *Proc. Int. Intell. Comput. Conf.*, Kunming, China, Aug. 2006, pp. 712–721.
[53] *Keras: The Python Deep Learning Library*, Astrophysics Source Code Library, USA, 2018.
[54] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[55] S. Kotz and S. Nadarajah, *Extreme Value Distributions: Theory and Applications*, 1st ed. Singapore: World Scientific, 2000.

ASMAA A. ELSAEIDY (Graduate Student Member, IEEE) received the M.Sc. degree in information systems and technology from the Faculty of Computers and Informatics, Zagazig University, Egypt. She is currently pursuing the Ph.D. degree with the Faculty of Science and Technology, University of Canberra, Canberra, Australia. Her Ph.D. research focuses on developing security frameworks and intrusion detection systems for smart cities applications using machine learning approaches. Her research interests include smart cities, the Internet of Things, cybersecurity, machine learning, and deep learning. She received the Best Paper Award for a 2017 published conference paper from her Ph.D. work.

NISHANT JAGANNATH received the master's degree in network engineering from the Faculty of Science and Technology, University of Canberra, where he is currently pursuing the Ph.D. degree with the Science and Technology Department. His doctoral research investigates the current issues in adopting blockchain technology, given the diverse nature of its implementations from financial markets to supply chain. He is also a part-time Faculty Member with the Science and Technology Department, University of Canberra. His research interest includes developing solutions for interoperability issues that are hindering standardization of blockchain technology for a variety of applications. He received the prestigious Dean's Excellence Award for his outstanding academic performance from the Faculty of Science and Technology, University of Canberra.

ADRIAN GARRIDO SANCHIS is currently a Chemical Engineer that works as an Associate Lecturer at the School of Science, UNSW Canberra, Canberra. He specializes on developing new technologies for environmental monitoring and the purification and sterilization of wastewater from bench to commercial-scale applications. He has pioneered the development of a novel sterilization process, which has been adopted by Australian Pork Ltd. (APL), who have recently funded his research and have funded the development and construction of a small-scale water treatment pilot unit, which has been trialed at a piggeries water treatment plant in NSW. He submitted a provisional patent application via UNSW to seek protection for this new process. He has substantial experience in commercial research and development and a unique range of abilities and skill set in water treatment and automation/environmental monitoring through the IoT devices.

ABBAS JAMALIPOUR (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Nagoya University. He is currently a Professor of ubiquitous mobile networking at The University of Sydney. He has authored nine technical books, 11 book chapters, and over 550 technical articles and holds five patents, all in the area of wireless communications. He is a Fellow of the Institute of Electrical, Information, and Communication Engineers (IEICE) and the Institution of Engineers Australia, an ACM Professional Member, and an IEEE Distinguished Speaker. He is the President of the IEEE Vehicular Technology Society. He held the positions of the Executive Vice-President and the Editor-in-Chief of *VTS Mobile World* and has been an elected member of the Board of Governors of the IEEE Vehicular Technology Society, since 2014. He was a recipient of a number of prestigious awards, such as the 2019 IEEE ComSoc Distinguished Technical Achievement Award in Green Communications, the 2016 IEEE ComSoc Distinguished Technical Achievement Award in Communications Switching and Routing, the 2010 IEEE ComSoc Harold Sobol Award, the 2006 IEEE ComSoc Best Tutorial Paper Award, and 15 Best Paper Awards. He has been the General Chair or the Technical Program Chair for a number of conferences, including the IEEE ICC, GLOBECOM, WCNC, and PIMRC. He was the Editor-in-Chief of the IEEE Wireless Communications, the Vice President-Conferences, and a member of the Board of Governors of the IEEE Communications Society. He serves as an Editor for IEEE Access, the IEEE Transactions on Vehicular Technology, and several other journals.

KUMUDU S. MUNASINGHE (Senior Member, IEEE) received the Ph.D. degree in telecommunications engineering from The University of Sydney. He is currently an Associate Professor in network engineering and the Leader of the IoT Research Group, Human Centred Research Centre, University of Canberra. He has over 100 refereed publications with over 860 citations (H-index 17) in highly prestigious journals, conference proceedings, and two books to his credit. His research interests include next-generation mobile and wireless networks, the Internet of Things, green communication, smart grid communications, and cyber-physical security. He has secured over $1.6 Million dollars in competitive research funding by winning grants from the Australian Research Council (ARC), the Commonwealth and State Governments, the Department of Defence, and the industry. He has also received the highly prestigious ARC Australian Postdoctoral Fellowship, served as the Co-Chair for many international conferences, and served as an editorial board member for a number of journals. His research has been highly commended through many research awards, including two VC's Research Awards and three IEEE Best Paper Awards. He is a Charted Professional Engineer, an Engineering Executive, and a Companion (Fellow Status) of Engineers Australia.