

Reservoir Computing Meets Smart Grids: Attack Detection Using Delayed Feedback Networks

Kian Hamedani¹, Lingjia Liu², *Senior Member, IEEE*, Rachad Atat,
Jinsong Wu³, *Senior Member, IEEE*, and Yang Yi, *Senior Member, IEEE*

Abstract—A new method for attack detection of smart grids with wind power generators using reservoir computing (RC) is introduced in this paper. RC is an energy-efficient computing paradigm within the field of neuromorphic computing and the delayed feedback networks (DFNs) implementation of RC has shown superior performance in many classification tasks. The combination of temporal encoding, DFN, and a multilayer perceptron (MLP) as the output read-out layer is shown to yield performance improvement over existing attack detection methods such as MLPs, support vector machines (SVM), and conventional state vector estimation (SVE) in terms of attack detection in smart grids. The proposed algorithms are shown to be more robust than MLP and SVE in dealing with different variables such as the amplitude of the attack, attack types, and the number of compromised measurements in smart grids. The attack detection rate for the proposed RC-based system is higher than 99%, based on the accuracy metric for the average of 10 000 simulations.

Index Terms—Attack detection, delayed feedback networks (DFNs), neuromorphic computing, reservoir computing (RC), smart grids, state vector estimation (SVE), temporal encoder.

I. INTRODUCTION

ENERGY harvesting from renewable resources, such as solar and wind, is gaining lots of attention from both academia and industry, especially with the ongoing increase in the world's power demand and the recent advancements in this field. Energy harvesting technologies are foreseen to power smart grid elements by up to 80%, including smart meters and sensors, which will significantly reduce battery replacement costs and the ongoing maintenance costs of smart grids. Furthermore, renewable energy will significantly reduce the fossil fuel power generation leading to a greener and sustainable

environment. A solar panel of size 121 centimeters (cm) by 53.6 cm or a wind turbine with a rotor of 1 meter (m) in diameter under an 8 m/s wind speed can generate 100 watt (W) of electric power [1]. Even though energy harvesting and renewable energy seem appealing for smart grids, they have several drawbacks and complications that unless addressed, very limited benefits can be gained from them [2]. Both wind and solar harvesting are unreliable as primary sources of power generation [3]. Energy harvesting should only be used as a supplementary source of power, where it can assist in reducing the power plant generation costs, carbon emissions, and fossil-based systems [2], [4], [5].

In this paper, we use wind turbines as a major source of electrical power generation for smart grids. The first wind turbine dates back to 1887, which had a peak power production of 12 kilowatts (KW) [6]. Since then, technological advancements have enabled a greater power generation, higher electrical conversion efficiency, and lower cost per kilowatt.

Cybersecurity is essential for ensuring the overall reliability of smart grids. Among possible cyber-attacks, the most critical one is the false data injection (FDI) [7]. Adversaries can launch these attacks by compromising smart meters to introduce malicious measurements.¹ If these malicious measurements affect the outcome of the state estimation, they can mislead the power grid control algorithms, possibly resulting in catastrophic consequences such as blackouts in large geographic areas. Therefore, attack detection is the most essential step for minimizing the damages resulting from the FDI. The efficiency and effectiveness of FDI detection can have a significant impact on the overall performance of smart grids. Feedforward neural networks have been applied on FDI detection but they did not yield good results because the spatio-temporal correlation of data is not considered in training [7].

On the other hand, it is found in [8] that recurrent neural networks (RNNs) are capable of exploiting the underlying correlation within the data. It was shown that under fairly mild and general assumptions, RNNs are universal approximations of dynamic systems. However, training a fully connected RNN in many cases is very difficult or even impossible [9]. Due to the difficulty of training traditional RNNs, reservoir computing (RC) recently attracted a lot of attention due to its simple training methods [10], [11]. Liquid state machine (LSM) [8] and echo state networks (ESNs) [12] are two most popular RC systems. The difference between LSM and ESN is that, LSM

Manuscript received June 27, 2017; revised October 5, 2017; accepted October 13, 2017. Date of publication November 2, 2017; date of current version February 1, 2018. Paper no. TII-17-1377. (Corresponding author: Lingjia Liu.)

K. Hamedani, L. Liu, and Y. Yi are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: hkian@vt.edu; ljliu@ieee.org; y631y347@ku.edu).

R. Atat is with the Electrical Engineering and Computer Science Department, University of Kansas, Lawrence, KS 66045 USA (e-mail: r487a045@ku.edu).

J. Wu is with the Department of Electrical Engineering, Universidad de Chile, Santiago 1025000, Chile (e-mail: wujs@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2017.2769106

¹ There are many online YouTube videos teaching how to hack smart meters.

uses spiking trains as the input, which has to be encoded by temporal or other encoding schemes, on the other hand ESN deals with regular data that is not a spike [8], [12]. In general, a typical RC system is composed of three different layers: the input layer, the reservoir, and the readout/output layer. The reservoir is mainly composed of randomly connected neurons where the weights of the connections between neurons stay unchanged during the training. The readout/output layer uses a linear combination of the reservoirs to produce the desired output [12], [13]. It has been shown in [12] and [14] that RC systems achieve better performance than traditional RNNs in many applications.

It is observed that delayed feedback networks (DFNs) are also capable of acting as RC systems [15]. The set of sparsely connected neurons (reservoirs) in LSM and ESN are replaced by a nonlinear node. This approach not only simplifies the structure of RC systems but also demonstrates a very significant computational efficiency [15]. The parallelism that exists in many other structures of artificial neural networks may simply be changed by a nonlinear node in which the input is inserted into that node [15]. It has been demonstrated in [16] that DFN performs very similar to other RC systems. The delayed networks with feedback system creates short term dynamic memory, which enables the network to mimic transient neural responses [17]. Transfer functions are the mathematical representations of the correlation between the input and output signals. In RC, nonlinear transfer functions are used to achieve the desired nonlinear mapping. Inspired by the Mackey–Glass function, we have designed an analog delay-based reservoir node with compact delay [18]. Similar to traditional delayed feedback reservoir designs, the introduced delayed feedback reservoir also consists of a single nonlinear node with a delay loop. The spiking nonlinear neural node serves the same purpose as well because the input of the delayed feedback reservoir is mapped nonlinearly to a higher dimensional space.

Several schemes have been introduced to encode the neural information. Rate encoding and temporal encoding are the two most popular ones [19]. In rate encoding, a code consists of a number of spikes occurring in a time frame after the stimulus appears [20]. Temporal encoding is subdivided into three main groups: latency code, interspike intervals, and phase of firing [21]. In latency code, the time in which the first spike occurs is used for encoding [20]. Interspike interval coding is another scheme that uses the intervals between different spikes for encoding [21], [22]. In the temporal encoding using phase of firing, the phase of the local field power is used to encode the information [23]. Studies show that interspike interval encoding carries more information than rate encoding [24], [25]. Therefore, in this paper, we use interspike interval temporal encoding as the encoder of our RC systems.

Equipped with the platform of analog spiking RC architecture, we will be able to conduct anomaly detection in cyber physical systems efficiently and effectively using RC. To be specific, in this paper, we show that by using DFNs and MLPs it is possible to efficiently and effectively detect attacks in smart grids. Compared to existing attack detection algorithms in smart grids, our introduced design shows a great deal of robustness

with respect to various attack variations. The main contributions of this paper are the following:

- 1) First, to the best of our knowledge, this is the first work to introduce the concept of RC for attack detection in smart grids. It is shown through simulations that the RC-based attack detection performs better than existing approaches. Furthermore, the accuracy of the attack detection of the RC-based approach is insensitive to attack variations such as the magnitude of the attack and the number of compromised meters.
- 2) Second, we modify the DFN so that it is able to take spike trains as the input. Note that spike encoding is more biologically plausible and very similar to the way that information is encoded in our brains. Several modifications are conducted on the existing DFN architecture in the literature: 1) A block is added to convert the spike train into analog signals before the nonlinear node and in the feedback loop. 2) The leaky-integrate and fire (LIF) neuron model is introduced as the nonlinear node in the DFN tailoring toward the input spike train.
- 3) Third, a multilayer perceptron is introduced as the readout layer that can deal with both nonlinear data and classification tasks.

We will show that the average attack detection rate based on the accuracy metric for 10 000 simulations will be above 99%. This paper is organized as follows. Section II reviews the related works in smart grid security. Section III the proposed design will be described; in Section IV the simulation results are presented and compares the results of the proposed algorithm with current existing methods and we will discuss why our proposed method outperforms the other methods in literature. Section V concludes the paper.

II. RELATED WORK

FDI problem in smart grids was first introduced in [26]. In [27], a summary of all the proposed methods for FDI detection and the advantages and disadvantages of each methods is presented. Tan *et al.* [28] present a survey of the recent data driven approaches in smart grid security. So far, many algorithms have been introduced for FDI in smart grids. Within these methods, the state vector estimation (SVE) [26] is among the first introduced algorithms. Machine learning techniques have also been introduced to FDI detection of smart grids. To be specific, feedforward neural network, K-nearest neighbor, support vector machines (SVM), and sparse logistic regression have been applied to FDI detection recently [7]. However, most of these techniques rely on manually chosen meta-parameters/parameters for the corresponding model. Even though the feedforward neural network allows for certain autonomy, its performance is usually strictly suboptimal when dealing with correlated data. Machine learning approaches show better results than support vector estimation methods when applied on IEEE test systems [29]. The effectiveness of the Precision Measurement Units have been extensively investigated in order to improve the performance of SVE [30], [31]. Extended distributed state estimation (EDSE) was studied by Cramer *et al.* [32]. EDSE uses graph partition

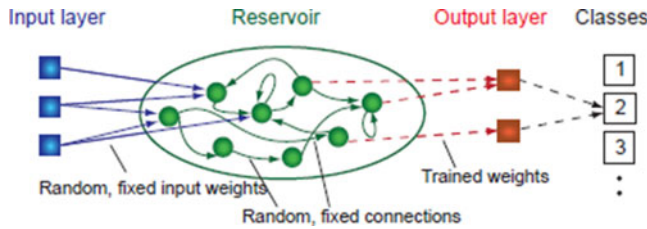


Fig. 1. RC System[15].

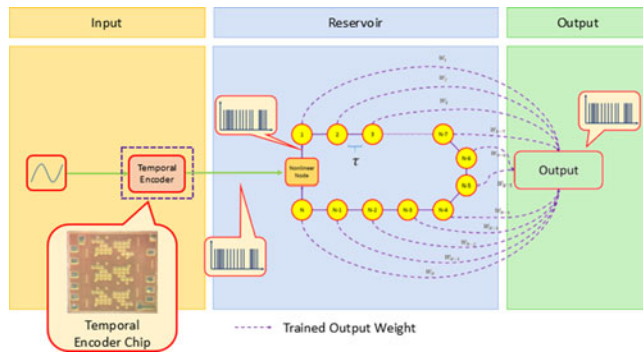


Fig. 2. Hardware implementation of delayed feedback reservoir system.

algorithms to divide each power system to several subsystems and in each subsystem three main categories are considered for the buses: boundary bus, internal bus, and adjacent bus. EDSE-based methods show better performance than the traditional state estimation methods. In [33], the compromised nodes are detected through the analysis of the existing relationship between the physical properties of the power system and FDI.

III. RC DESIGN FOR ATTACK DETECTION IN SMART GRIDS

A. Realizing RC Using DFN

Fig. 1 shows the structure of a RC system. The only difference between traditional RC models such as ESN and LSM and DFN is in the reservoir layer [13]. As it can be seen in Fig. 1, the reservoir layer in traditional models of RC contains neurons, which are sparsely connected using recursive connections, however, in the DFN there is only one nonlinear node and the output or the state of this nonlinear node is shifted in time in order to produce the states of other nodes or virtual nodes [16]. Fig. 2 illustrates the structure of the DFN used in this paper. The first layer is the input layer in which the temporal encoder used in [19] is applied. The temporal encoder details are explained in Section III-B. The data used consists of 10 000 vectors of measurements extracted from MATPOWER 5.1 [34]. Half of the measurements were attacked by a random Gaussian vector. The variance of the attack is set to 0.05. The vector of the combined attacked and nonattacked data are saved and the temporal encoder is applied on the data. For any sample in that vector, a corresponding spiking train is produced. In this way, we are able to convert the measurement matrix extracted from 57 buses to its corresponding temporal code. The size of the measurement matrix coming from the MATPOWER is 137 due to the

fact that several meters will be on the same bus. In the next step, these spikes are applied on the nonlinear node of the DFN. There are several design choices for the nonlinear node. Since we are interested in spiking neural networks, the input node of the reservoir layer will be a LIF neuron [35]. These produced spikes will have to be converted to an analog current before being applied to the LIF neuron. A corresponding spike train will be generated for any analog current.

Delay exists in almost all the systems with dynamics. Inevitably, delays may even occur in the brains when information is transmitted from one neuron to the other. Delay differential equations are used to mathematically represent delayed systems[36]. For any delayed systems, the dynamics of the system depend not only on the current states but also on previous states. Such systems exhibit the characteristics of high dimensionality and short term memory, which are the two prerequisites for any RC systems[37].

Compared to the traditional RC systems, delayed feedback RC has practically similar performance [15]. Different from the traditional reservoir, delayed feedback reservoir is constructed by a single nonlinear node and a delay loop. Output from the reservoir will undergo a training process in which a training algorithm is employed. The objective of the training is to ensure that the weighted sum of the state approaches the target output value. The input is injected directly to the nonlinear node. In order to compensate the loss of parallelism, a masking procedure is carried out before the nonlinear node. During the masking procedure, the input signals are scaled whereby they will be in the transient regime [15]. After the masking procedure, the signals are then transferred to the nonlinear node where the nonlinear mapping takes place. Similar to the traditional RC, the output weight connections are the only trained weights.

Inspired by the delayed feedback reservoir, we introduce an analog hardware implementation of the delayed feedback RC system with the capability of processing spike-based signals directly. With the analog implementation, the use of peripheral components, such as analog-to-digital and digital-to-analog converters, is avoided when interfacing with analog signals. In general, analog implementation has the advantage of implicit real-time operation, resulting in small design area and low power [38]–[40]. In our design, the spike train produced by the LIF neuron is shifted 10 milliseconds (ms) in time to produce the state of the second node in the reservoir. This process is repeated four times until we obtain a different state. Before signals are injected into the nonlinear node, information is usually encoded.

In general, there are two types of encoding strategies: rate encoding and temporal encoding. Rate encoding scheme ensures that the input information is represented by the number of spikes, whereby other spike characteristics are ignored. On the other hand, temporal encoding encodes information into the inter-spike intervals. Using temporal encoding, analog signals will be encoded into spike-based information, which not only possess a compact form but also are energy efficient. In our design, we use temporal encoding and an iterative structure is adapted in the temporal encoder where the number of neurons and the number of spikes are in an exponential relationship. In this way, less neurons would be needed to achieve the same number of spikes.

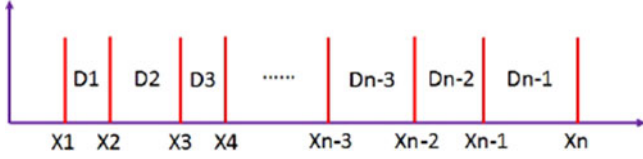


Fig. 3. Interspike intervals [19].

The temporal encoder ensures that only one neuron is in the dynamic mode in which the power consumption is greatly reduced. Our introduced temporal encoder has been fabricated using 180 nm CMOS process and symmetry scheme to maximize the die area utilization. Our design not only employs the internal verification technique, but also uses the output temporal code, which exhibits high error-tolerance mechanism achieved via exploiting the additional inspection spikes. Apart from possessing high accuracy, the introduced neuron also exhibits low power consumption when compared to other state-of-the-art neuron designs [17]. We could extract five different states for every sample in the measurement matrix. These states will be used to train a multilayer perceptron (MLP). The feature used for training the MLP is the times at which spikes are occurring for the corresponding state of every sample. Since half of the samples are attacked, the corresponding label of the attacked data for training the reservoir state is considered as one and zero otherwise. After the MLP was trained by the training data, the test data are then used to evaluate the performance of the system.

B. Temporal Encoder

The encoder introduced in [19] serves as the temporal encoder of our design. The corresponding inter-spike intervals can be expressed as follows

$$D_i = f(C_i, V_i) - f(C_{i-1}, V_{i-1}). \quad (1)$$

The function $f(X, Y)$ is expressed as

$$f(C_i, V_i) = (C_{i+1}) [\beta (V_i - \gamma) + \theta] \quad (2)$$

where the parameters specify the characteristics of the encoder, such as charging and refractory periods. C_i and V_i are the parameters of membrane capacitance and threshold firing voltage of the temporal encoder, respectively.

Using the above temporal encoder, any sample in the measurement matrix is encoded in the interspike interval distances, D_i . There may be different number of intervals based on the number of neurons used in the temporal encoder for any sample. In this experiment, for the sake of simplicity, we choose the number of neurons in the encoder to be $N = 3$, which results in four different spikes, X_1 to X_4 or three intervals D_1 to D_3 which can be seen in Fig. 3. Due to the following equation mentioned in [19], there is a relationship between the number of spikes produced and the number of neurons used in the temporal encoder:

$$S_N = 2^{N-1} \quad (3)$$

where, S_N is the number of spikes produced by the temporal encoder and N is the number of neurons used in the encoder.

C. Smart Grid Attack Detection Formulation

Smart grids are used to make a reliable power transmission network and connection between consumers and generators. They are really vulnerable to cyber-attacks, and thus, it is a very important and challenging task to provide a secure network of smart grids [41]. MATPOWER 5.1 can be used to produce the smart grids' measurement matrix [42]. MATPOWER allows the users to run the toolbox with different numbers of buses. In our experiment, the number of buses is set to 57 resulting in 137 different measurements. Note that it is pointed out in [7] that the parameter that really impacts smart grid attack detection is the number of compromised meters instead of the number of buses. The reason that we pick 57 is because this number is almost in the middle of the range of the number of the buses that is provided by MATPOWER. This configuration will result in 137 meters, which is large enough for us to study the effect of different number of the attacked meters in that configuration [43].

The system model that is used to study the attack detection in smart grids is defined in [26]

$$z = Hx + n. \quad (4)$$

The measurement vector, which is the output of different meters on the buses is z ; H is the state vector; x is the voltage phase of the buses; and n is the environment noise. When attack is present, an attack vector, a , is added to the measurement. Accordingly, the measurement, \tilde{z} , becomes

$$\tilde{z} = Hx + a + n. \quad (5)$$

We assume that the attack is a Gaussian random vector with 0.05 variance [43].

SVE is the first method introduced to perform attack detection for smart grids. This method consists of calculating a residual stated as ρ . If the value of ρ exceeds a predefined threshold value, it is said that the z vector has been attacked and the meters are compromised [26].

$$\rho = \|\tilde{z} - H\hat{x}\|_2^2 \quad (6)$$

where \hat{x} is the vector estimated using SVE algorithm. \hat{x} is then estimated as follows:

$$\hat{x} = (H^T \wedge H^{-1}) H^T \wedge z. \quad (7)$$

The only action that has to be done in SVE algorithm is to estimate the \hat{x} , and to do so, \wedge needs to be calculated, where \wedge is defined as a diagonal matrix with its diagonal elements are the reciprocals of the variance of the measurements. For example, the j th diagonal element of the \wedge is equal to the reciprocal of the variance of the j th element of the z . SVE method is a very simple method for implementation but it has many shortcomings with different attack situations [26].

In the case where $a = HC$, the attack is hidden (see Appendix A). It means that SVE is incapable of detecting the bad measurements [26]. In the case of hidden attacks, the residual value is less than the threshold and the attack cannot be detected

by SVE. In order to perform a hidden attack, the cyber attacker has to have access to at least a specific number of attacks. In [26], it has been shown that it is not possible for the attacker to choose any arbitrary c and multiply it by H to perform the hidden attack. This means that in order to make a hidden attack, the attacker has to have access to at least k measurements, in which $k > m - n$ where m is the number of meters and n is the number of buses. In our system, $m = 137$ and $n = 57$. In this case, $m - n = 80$ meaning that with high chance the attack will be a hidden or stealth attack when there are more than 80 compromised measurements in our smart grid network. Under this scenario, the SVE becomes an inefficient attack detector.

D. Modeling the Wind Power Generators in MATPOWER

MATPOWER can also be used to study renewable energy, especially wind powers [44]. There are six power generators for a 57-bus smart grid network. It is possible to substitute the power of these generators with the power obtained from wind power generators. Accordingly, (8) gives the power produced by a wind power generator as the source of energy [45]

$$P_{\text{avail}} = 1/2 \rho A v^3 C_P \quad (8)$$

where P_{avail} is the power converted from wind; ρ is the air density, which is assumed to be equal to 1.23 kg/m^3 ; A is the sweep area of the wind turbine blades; v is the speed of the wind; and C_P is the coefficient of the power. Albert Betz, a German physicist, has shown that the maximum value for the power coefficient is equal to 0.59. This is called Betz Limit or Betz Law. Based on that, the performance of a wind power generator cannot exceed 0.59 [46]. In this study, the value of C_P is set to 0.4 while the area of the generator is set to 8495 m^2 , six different values ranging from 0 to 12 m/s are used for the wind speed. These power values are inserted in the MATPOWER to produce H matrix [46].

E. Smart Grid Attack Detection Using DFN and MLP

The FDI problem can also be formulated as a classification problem. So far, many machine learning algorithms have been suggested to deal with this problem [47]. To the best of our knowledge, this problem has never been studied from RC's point of view. We are the first to study this problem using RC methods. In the FDI problem we face two classes of data: attacked data and nonattacked data, we can assign two different labels for these two classes and figure out the classification of data.

In this experiment, two different sets of data are used. The data which has been attacked by a hidden attack and the data, which its measurements have been attacked by direct or nonhidden attack vectors. The experiments are performed on 1000 samples and the experiments are repeated 10 times. The first step is to encode z using the temporal encoder. Then, every spike train extracted from the temporal encoder is converted to an analog current. In [48], an equation was introduced to convert the spike trains to the analog current

$$I^i = \sum_{t^j} K(t - t^j) \mathcal{H}(t - t^j) \quad (9)$$

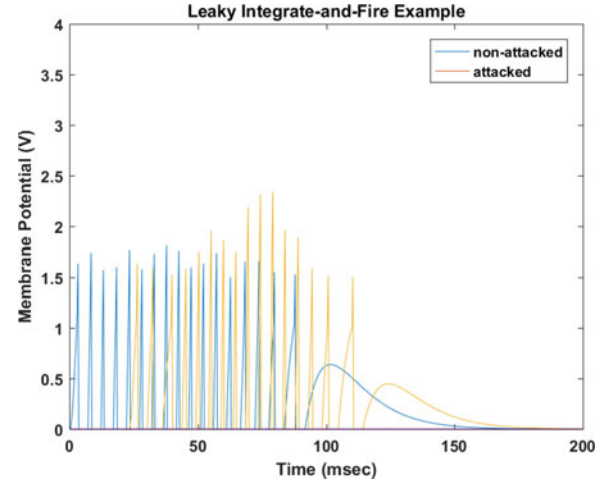


Fig. 4. Average DFN states for attacked and nonattacked data.

where \mathcal{H} is the Heaviside function; I^i is the analog current of the i th sample in the z ; and t^j is the time of occurrence of the j th spike in the corresponding spike train of the i th sample achieved from the temporal encoder [48]; and

$$K(t - t^j) = V_0 (\exp(-(t - t^j)/\tau_s) - \exp(-(t - t^j)/\tau_f)) \quad (10)$$

where τ_s is set to 10 ms and τ_f to 2.5 ms. The values of τ_s and τ_f have to be chosen somehow that $\tau_s/\tau_f = 4$. V_0 is a normalization factor to make sure that the maximum value of kernel does not exceed one [48].

Up to now what we can generate analog current signals from (9) and (10) corresponding to the temporal codes extracted from the temporal encoder. The next step is to apply these current on the DFN to produce the corresponding states. As mentioned in Section III-A, the nonlinear node of the DFN is chosen to be an LIF neuron. The analog current signals for the attacked samples and nonattacked samples were applied to the DFN. The output of the LIF neuron is shifted 10 ms in time to produce the state of the first virtual node. This process is repeated in four times until we obtain four virtual nodes. Note that the state of the fourth virtual node is shifted 40 ms compared to the nonlinear node. Then, the state of the fourth virtual node is multiplied by 0.8, which is the feedback gain, and is then added to the new incoming analog current. Now for both attacked and nonattacked samples, five different states are generated. Fig. 4 shows the average total state of attacked and nonattacked samples.

From Fig. 4, it is clear that the timing of the spikes for the average states of the two data classes are very different. It can be seen that average spikes produced for the attacked samples are more likely to fire at smaller times and the ones fired for nonattacked samples are more likely to fire at larger times. Furthermore, it is possible to use these timings as a feature to classify these two groups. Therefore, in the next step we utilize an MLP and train the MLP with these features [49].

F. Training an MLP With the Timing of Spikes

As demonstrated in [15], the readout layer can be trained with a linear algorithm. In the introduced training algorithm in [15],

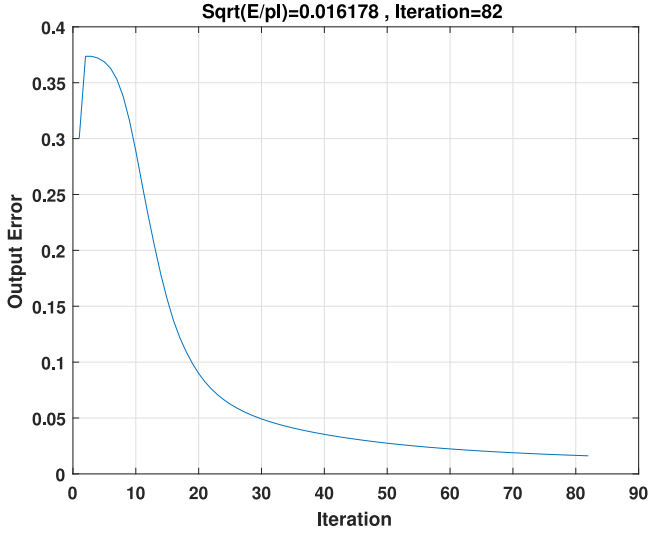


Fig. 5. Error Plot for training an MLP with DFN states spikes timings.

a weight was assigned to the every state extracted from the DFN in a way that the desired output values can be estimated with the least possible error. The following expression provides a good summary of the training algorithm in [15]

$$\hat{y}(k) = \sum_{i=1}^N w_i \times x[k\tau - \tau/N(N-i)] \quad (11)$$

where $\hat{y}(k)$ is the estimated output; w_i is the connection weight; x is the state vector; and N is the number of states. The above algorithm is linear and not iterative so it won't be very precise. Therefore, in our RC-based attack detector, we adopt an MLP for output estimation. The algorithm used for training the MLP is backpropagation. The label of y is set to 1 for training the samples being attacked and 0 for the samples not being attacked. The time in which attacks spikes happen for different states are saved in a vector and are used as features for training the MLP. The MLP is trained with two different hidden layers and one output layer. The desired output for the attacked sample is 1 otherwise it is 0. As it can be seen in Fig. 5, the MLP is trained after 82 iterations. Then, the weights are saved to be applied on the test data to evaluate the performance of the system. In the next step, SVE algorithm, MLP, and SVM are applied on the samples to compare against the performance of our introduced RC-based attack detection strategy. We use Gaussian radial basis function as the kernel of the SVM classifier (see Appendix B for details) [43].

As it can be seen in Fig. 5, the training mean square error (MSE) reduces to 0.016, which indicates that MLP is capable to distinguish between the states spike timings of the attacked data and the nonattacked data. The learning rate is set to 0.01 and momentum factor to 0.5. If the output value is greater than 0.5, it is considered as 1, else it is considered as 0. In that sense, the training accuracy is almost 100%, meaning that almost 100% of the samples are classified accurately as attacked and nonattacked. In order to quantify the detection performance, the *accuracy* metric is defined in (12). We used 50% of the samples

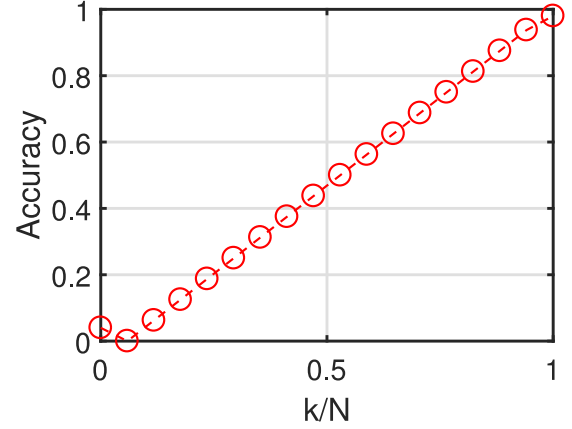


Fig. 6. Accuracy of the SVE.

for training and the rest are saved for testing and validation. The results of applying the DFN algorithm are presented in Section IV. Fig. 7 shows the block diagram of our RC-based attack detection algorithm.

G. State Vector Estimation

As mentioned in Section III-C, $\rho = \|\tilde{z} - H\hat{x}\|_2^2$ needs to be computed for SVE. If the value of ρ exceeds a predefined threshold value, it is said that an attack has occurred, nonattack is detected otherwise [26]. Accordingly, we can calculate the value of ρ when the measurement vector is attacked by the same attack vector mentioned in the previous section. The value of ρ achieved for attacked vectors with different number of compromised measurements is used to evaluate the performance of SVE. However, we show in the next section that there are some drawbacks with SVE. The performance metric used to evaluate the detection performance is the accuracy, which is defined as

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (12)$$

where TP, TN, FP, and FN correspond to the number of true positive, true negative, false positive, and false negative samples, respectively.

The attack detection performance of SVE can be clearly seen in Fig. 6. As seen in the figure, the accuracy of SVE is severely affected by the number of compromised measurements even when the attack is not hidden. This is due to the fact that the performance of SVE depends heavily on the residual value. When the number of compromised measurements is small, the accuracy of the SVE drops significantly. In Section IV, we will show that this issue can be completely resolved by the introduced RC-based DFN+MLP attack detector.

IV. PERFORMANCE EVALUATION

As it was mentioned in Section III only 50% of the data are used for training and the rest is saved for test and validation. In this section, we will detail the performance evaluation of the three aforementioned algorithms: RC-based method (DFN+MLP), MLP, and SVM. We have totally 5000 samples for testing and validation. Half of them are attacked and half

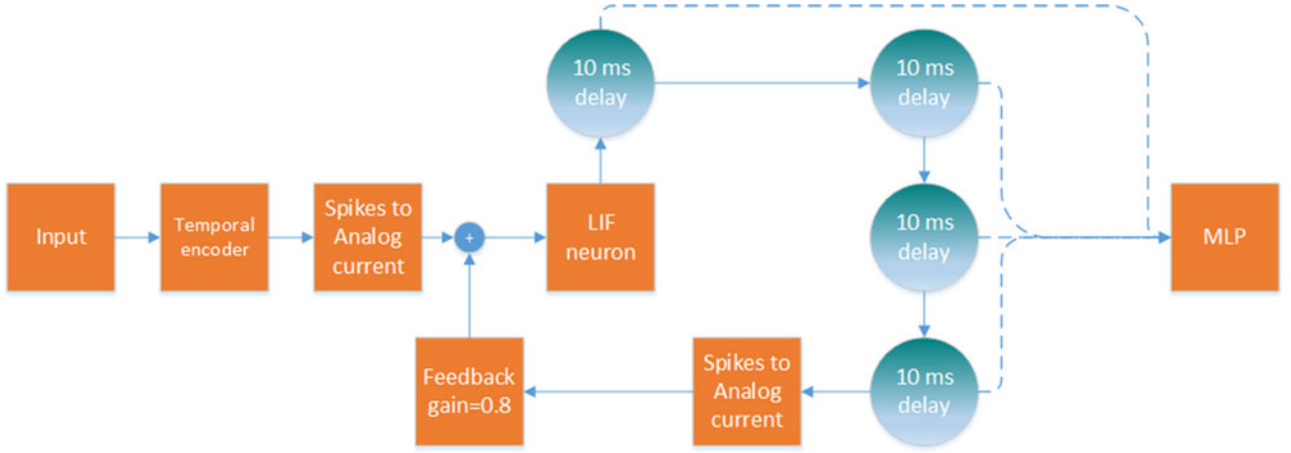


Fig. 7. Block diagram of the proposed DFN+MLP system for attack detection.

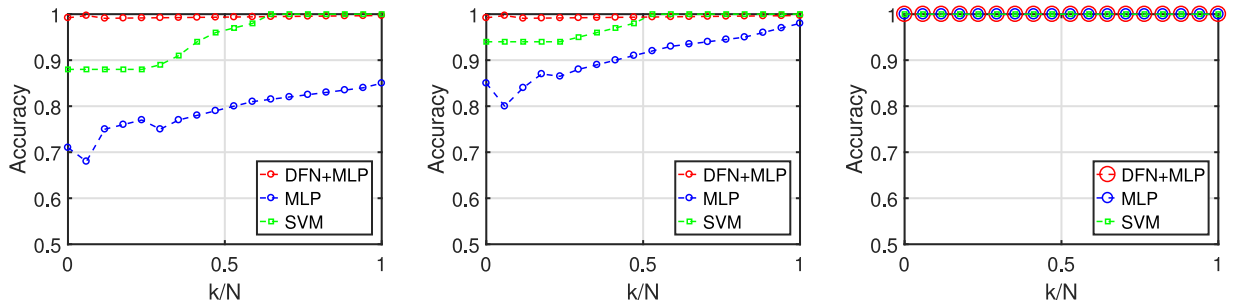


Fig. 8. Accuracy of direct attack detection for three different methods, $a = 0.1, 1, 10$.

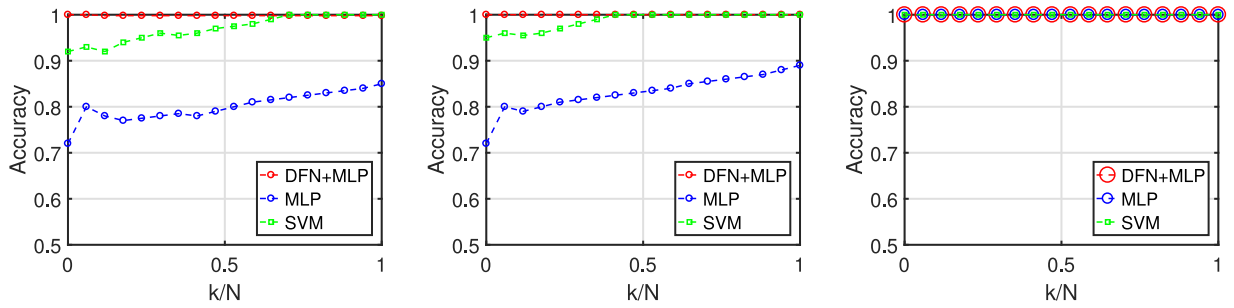


Fig. 9. Accuracy of hidden attack detection for three different methods, $a = 0.1, 1, 10$.

of them are not. As the main evaluation results, Figs. 8 and 9 show the accuracy of the proposed method for the two types of attacks in smart grids, hidden and direct, as a function of the attack magnitude a . Three different values of the attack magnitude are used: $a = 0.1$, $a = 1$, and $a = 10$. Note that since SVE is not capable of detecting hidden attacks [26], we did not evaluate its performance in Figs. 8 and 9.

From the figures, we can clearly observe that the performance of both MLP and SVM are very sensitive to attack magnitudes as well as the number of attacked meters. Unlike SVE, both MLP and SVM can detect hidden attacks. However, their detection performances are very sensitive to attack parameters. For example, the accuracy of both MLP and SVM increases as the attack magnitude increases. This means that MLP and SVM can detect attacks accurately when attacks have large magnitudes.

However, when attacks have small magnitudes, MLP and SVM will detect attacks with less certainty. To be specific, for the case of MLP, the accuracy is 100% when the magnitude of the attack is 10 and can be as low as 70% when the attack magnitude is 0.1. This is not very desirable for attack detection in smart grids where the attack magnitude can be arbitrary. For the RC-based DFN+MLP method, we can see that the variations of attack magnitude do not cause any significant change to the accuracy. To be specific, the accuracy variation due to the change in attack magnitude is very small for RC-based approach and the accuracy is close to 100% in all attack magnitudes. This clearly suggests that the attack detection performance of the RC-based approach is robust under different attack magnitudes. Figs. 8 and 9 also show the accuracy as a function of the number of compromised meters for different attack detection strategies.

As discussed in Section III-C and Section III-G, SVE is not capable of detecting hidden attacks, therefore, we did not evaluate its performance in Figs. 8 and 9. From the figures we can see that the introduced RC-based approach is much more robust than the MLP and the SVM method under different number of compromised meters. Furthermore, comparing the two figures, we can observe that unlike existing detection strategies (SVE, MLP, and SVM) the RC-based DFN+MLP method provides uniform performance under different attack methods (direct and hidden). In this study, 50% of the samples are attacked and the rest not, which means we are dealing with a balanced data set and if the number of attacked and nonattacked samples are significantly different the data set is imbalanced [43]. The imbalanced data set is very likely to compromise the performance of the learning algorithm [50]. In such scenarios F1 score is used to evaluate the performance of the learning algorithm [43], [51]. F1 measure can handle the imbalanced data. In [43] the detection performance evaluation is studied for both balanced and imbalanced data set extracted from IEEE 30-bus system.

$$F1 = (2TP) / (2TP + FP + FN). \quad (13)$$

In that study [43], the performance plots, accuracy for balanced data set and F1 measure for the imbalanced data set, do not show any meaningful difference.

V. CONCLUSION

In this paper, we introduced a RC-based (DFN+MLP) attack detection strategy for smart grids. The introduced method constitutes of three main steps. The first step is encoding the measurement vector with temporal encoder and converting the produced spikes to their corresponding analog currents. In the second step, these analog currents are applied on an LIF neuron and shifted in time to produce the states of virtual nodes. The output of the fourth virtual node is multiplied by a feedback gain and added to the new incoming data in order to preserve the recurrent nature of the DFN. The spiking times of these states are used to train an MLP for classification. Simulation results have shown that this algorithm can robustly detect attacks under different attack variations such as magnitudes and the number of compromised meters compared to existing methods such as SVE, MLP, and SVM. It is also important to note that this paper is the first effort to solve FDI problems in smart grids through RC. The proposed model can be applied on any classification task that there is spatio-temporal correlation between the samples of the data set. In our next work we will show that we have been able to apply this model successfully on face recognition task from video frames. Since there are spatio-temporal correlations among the meters in smart grids, RC-based attack detection can take full advantage of this spatio-temporal correlation yielding a better performance compared to existing solutions.

APPENDIX A

$$\begin{aligned} \rho &= \|\tilde{z} - H\hat{x}\| \\ &= \|z + a - H(x + (H^T \wedge H)^{-1} H^T \wedge a)\| \end{aligned}$$

$$\begin{aligned} &= \|z - Hx + (HC - H((H^T \wedge H)^{-1} H^T \wedge HC))\| \\ &= \|z - Hx + (HC - HC)\| \\ &= \|z - Hx\| \leq \tau \end{aligned} \quad (14)$$

where, τ corresponds to the threshold of detection.

APPENDIX B

SVM is a binary classifier. It tries to find two parallel hyperplanes that maximizes the margin between two classes.

$$\begin{cases} w^T s_i + b = +1 & \text{if } y_i = +1 \\ w^T s_i + b = -1 & \text{if } y_i = -1 \end{cases} \quad (15)$$

where y_i is the label of each class; s_i is each data point, and w is the weight matrix that has to be found by SVM [43]. Margin is the region in between the two hyperplanes and the weights have to maximize this margin. The margin is defined as $D = 2/w^T w$. In order to have the maximum margin, the following optimization problem has to be solved by the SVM

$$\min_w \frac{1}{2} w^T w \quad (16)$$

$$\text{s.t. } y_i(w^T s_i + b) \geq 1, i = 1, 2, \dots, m. \quad (17)$$

The Lagrangian optimization has to be used to find b and w [43]

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^m \alpha_i [y_i(w^T s_i + b) - 1]. \quad (18)$$

In (17) α corresponds to Lagrange multiplier. The Lagrange dual can be replaced by (17) because (17) satisfies Karush–Kuhn–Tucker conditions as

$$L(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j s_i^T s_j \quad (19)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0. \quad (20)$$

This solution works for linearly separable data samples, but when the data points are not linearly separable $s_i^T s_j$ has to be replaced by a kernel function, $K(s_i, s_j)$ [43] as

$$K(s_i, s_j) = \phi(s_i)^T \phi(s_j) \quad (21)$$

where ϕ is a function that maps the data points to higher dimension in order to make them linearly separable. In this paper, we use Gaussian radial basis function as the kernel

$$K(s_i, s_j) = e^{-\lambda \|s_i - s_j\|^2}. \quad (22)$$

REFERENCES

- [1] Y. Mao, Y. Luo, J. Zhang, and K. B. Letaief, "Energy harvesting small cell networks: Feasibility, deployment, and operation," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 94–101, Jun. 2015.
- [2] E. Camacho, T. Samad, M. Garcia-Sanz, and I. Hiskens, "Control for renewable energy and smart grids," *Int. J. Impact Control Technol.*, 2011, pp. 19–25.
- [3] P. He and L. Zhao, "Noncommutative composite water-filling for energy harvesting and smart power grid hybrid system with peak power constraints," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2026–2037, Apr. 2016.

- [4] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green Challenges: Big data toward green applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.
- [5] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green Challenges: Greening big data," *IEEE Syst. J.*, vol. 10, no. 3, pp. 873–887, Sep. 2016.
- [6] S. Soter and R. Wegener, "Development of induction machines in wind power technology," in *Proc. IEEE Int. Elect. Mach. Drives Conf.*, 2007, pp. 1490–1495.
- [7] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.
- [8] W. Maass, T. Natschlger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [9] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.
- [10] S. Mosleh, C. Sahin, L. Liu, R. Y. Zheng, and Y. Yi, "An energy efficient decoding scheme for nonlinear mimo-ofdm network using reservoir computing," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2016, pp. 1166–1173.
- [11] S. Mosleh, L. Liu, C. Sahin, R. Y. Zheng, and Y. Yi, "Brain-inspired wireless Communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
- [12] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," *German Nat. Res. Cntr. Inf. Technol.*, 2001.
- [13] B. Schrauwen, D. Verstraeten, and J. Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. Eur. Symp. Artif. Neural Netw., Bruges*, 2007, pp. 471–482.
- [14] X. Hinaut and P. Dominey, "On-line processing of grammatical structure using reservoir computing," in *Proc. Int. Conf. Artif. Neural Netw.*, 2012, pp. 596–603.
- [15] L. Appeltant, "Reservoir computing based on delay-dynamical systems," in *These de Doctorat, Vrije Universiteit Brussel/Universitat de les Illes Balears*, 2012.
- [16] L. Appeltant *et al.*, "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, Aug. 2011, Art. no. 468.
- [17] M. Tateno and A. Uchida, "Nonlinear dynamics and chaos synchronization in mackey-glass electronic circuits with multiple time-delayed feedback. nonlinear theory and its applications," *IEICE Nonlinear Theory Appl.*, vol. 3, no. 2, pp. 155–164, 2012.
- [18] C. Zhao, J. Li, L. Liu, L. S. Koutha, J. Liu, and Y. Yi, "Novel spike based reservoir node design with high performance spike delay loop," in *Proc. 3rd ACM Int. Conf. Nanoscale Comput. Commun.*, Sep. 2016, pp. 1–5.
- [19] C. Zhao *et al.*, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 4, pp. 256–276, Sep. 2016.
- [20] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends Neurosci.*, vol. 33, no. 3, pp. 1964–1974, Jan. 2010.
- [21] D. Reich, F. Mechler, K. P. Purpura, and J. Victor, "Interspike intervals, receptive fields, and information encoding in primary visual cortex," *J. Neurosci.*, vol. 20, no. 5, pp. 1964–1974, Mar. 2000.
- [22] S. M. Chase and E. Young, "First-spike latency information in single neurons increases when referenced to population onset," *Nat. Acad. Sci. United States Amer.*, vol. 104, no. 12, pp. 5175–5180, Jan. 2007.
- [23] J. Lisman, "The theta/gamma discrete phase code occurring during the hippocampal phase precession may be a more general brain coding scheme," *Hippocampus*, vol. 15, no. 7, pp. 913–922, 2005.
- [24] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophys. J.*, vol. 1, no. 6, pp. 445–466, 1961.
- [25] C. Kayser, M. A. Montemurro, N. K. Logothetis, and S. Panzeri, "Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns," *Neuron*, vol. 61, no. 4, pp. 597–608, 2009.
- [26] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *ACM Trans. Inf. Syst. Security*, vol. 14, 2011, Art. no. 13.
- [27] R. XU, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du, "Achieving efficient detection against false data injection attacks in smart grid," *IEEE Access*, vol. 5, pp. 13 787–13 798, Jul. 2017.
- [28] S. Tan, D. De, W. Song, J. Yang, and S. Das, "Survey of security advances in smart grid: A data driven approach," *IEEE Commun. Surveys Tut.*, vol. 19, no. 1, pp. 397–422, First Quarter 2017.
- [29] D. Srinivasan and G. Venayagamoorthy, "Guest editorial on neural networks and learning systems applications in smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1601–1603, Aug. 2016.
- [30] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 198–207, Aug. 2015.
- [31] S. Bi and Y. Zhang, "Graphical methods for defense against false-data injection attacks on power system state estimation," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1216–1227, May 2014.
- [32] M. Cramer, P. Goergens, and A. Schnettler, "Bad data detection and handling in distribution grid state estimation using artificial neural networks," in *Proc. IEEE Eindhoven PowerTech*, Jun. 2015, pp. 1–6.
- [33] A. Adnan, A. Mahmood, and Z. Tari, "Identification of vulnerable node clusters against false data injection attack in an AMI based smart grid," *Inf. Syst.*, vol. 53, pp. 201–212, Oct. 2015.
- [34] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [35] Y. Liu and X. J. Wang, "Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron," *J. Comput. Neurosci.*, vol. 10, pp. 25–45, 2001.
- [36] J. Li, L. Liu, C. Zhao, K. Hamedani, R. Atat, and Y. Yi, "Enabling sustainable cyber physical security systems through neuromorphic computing," *IEEE Trans. Sustain. Comput.*, to be published.
- [37] J. Li, C. Zhao, and Y. Yi, "Energy efficient and compact analog integrated circuit design for delay-dynamical reservoir computing system," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, to be published.
- [38] L. Smith, *Neuromorphic Systems: Past, Present and Future*. New York, NY, USA: Springer-Verlag, 2010.
- [39] A. Basu and P. E. Hasler, "Nullcline-based design of a silicon neuron," *IEEE Trans. Circuits Syst.*, vol. 57, no. 11, pp. 2938–2947, Nov. 2010.
- [40] K. Ramanaiah and S. Sridha, "Hardware implementation of artificial neural networks," *i-Manager's J. Embedded Syst.*, vol. 3, no. 4, 2014, Art. no. 31.
- [41] R. Anderson, A. Boulanger, W. Powell, and W. Scott, "Adaptive stochastic control for the smart grid," *Proc. IEEE*, vol. 99, no. 6, pp. 1098–1115, Jun. 2011.
- [42] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.
- [43] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 1395–1402.
- [44] Y. Kumar, "Study of power and renewable systems modeling and simulation tools," M.S. thesis, Dept. Elect Eng, Univ. Toledo, Toledo, OH, USA, 2015.
- [45] O. Bekri, M. Fellah, and M. F. Benkhoris, "Impact of the wind generator on the power flow in the electric grid," in *Proc. IEEE Int. Symp. Environ. Friendly Energies App.*, Nov. 2014, pp. 1–6.
- [46] "Wind turbine power calculations, RWE npower renewables," *Mechanical and Electrical Engineering Power Industry, The Royal Academy of Engineering*, Dec. 2012.
- [47] A. Patrascu and V. Patriciu, "Cyber protection of critical infrastructures using supervised learning," in *Proc. IEEE Control Syst. Comput. Sci.*, May 2015, Art. no. 461C468.
- [48] Q. Yu, H. Tang, K. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero association of spatiotemporal spike patterns," *PLOS One*, vol. 8, no. 11, 2013, Art. no. e78318.
- [49] K. Hamedani, S. Seyyedsalehi, and R. Ahamdi, "Video-based face recognition and image synthesis from rotating head frames using nonlinear manifold learning by neural networks," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1761–1769, 2016.
- [50] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [51] F. Wu, X. Jing, S. Shan, W. Zuo, and J. Yang, "Multiset feature learning for highly imbalanced data classification," in *Proc. AAAI, Conf. Artif. Intell.*, Feb. 2017, pp. 1583–1589.



Kian Hamedani received the B.S. degree in electronics from Iran University of Science and Technology, Tehran, Iran in 2009 and the M.Sc. degree in biomedical engineering-bioelectric from Amirkabir University of technology (Tehran Polytechnic), Tehran, Iran in 2012. He is currently working toward the Ph.D. degree in MICS research group at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.

His research interests include Neural Networks, Neuromorphic computing, machine learning, data analysis, and circuit design.



Lingjia Liu (SM'15) received the B.S. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA.

He spent the summer of 2007 and spring of 2008 in the Mitsubishi Electric Research Laboratory. Prior to joining the Electrical and Computer Engineering (ECE) Department, Virginia Tech (VT), Blacksburg, VA, USA, he was an Associate Professor in the Electrical Engineering and Computer Sciences Department, University of Kansas (KU), Lawrence, KS, USA. He spent more than three years working in the Standards & Mobility Innovation Laboratory, Samsung Research America (SRA). He was leading Samsung's efforts on multiuser MIMO, coordinated multipoint (CoMP), and heterogeneous networks in LTE/LTE-Advanced standards. His research interests include emerging technologies for 5G cellular networks including machine learning for wireless networks, massive MIMO, massive MTC communications, and mmWave communications.

Dr. Liu is currently an Editor for the IEEE TRANSACTION ON WIRELESS COMMUNICATIONS, an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, and an Associate Editor for the *EURASIP Journal on Wireless Communication and Networking*, and *Wiley's International Journal on Communication Systems*. He received Air Force Summer Faculty Fellow in 2013, 2014, 2015, 2016, and 2017, Miller Scholar at KU in 2014, Miller Professional Development Award for Distinguished Research at KU in 2015, and 2016 IEEE GLOBECOM Best Paper Award. He was the recipient of Global Samsung Best Paper Award twice at SRA (in 2008 and 2010, respectively).



Rachad Atat received the B.E. degree with Distinction in computer engineering from the Lebanese American University (LAU), Beirut, Lebanon, in 2010, the M.Sc. degree in electrical engineering from the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, in 2012, and the Ph.D. (Hons.) degree in electrical engineering from the University of Kansas (KU), Lawrence, KS, USA, in July 2017.

His current research interests include device-to-device communications, cybersecurity, and Internet of Things.

Dr. Atat received the Best Paper Award from the 2016 IEEE Global Communications Conference (GLOBECOM) and the NSF Travel Grant in 2016. He was a recipient of the KU Engineering Fellowship Award and the KAUST Discovery Scholarship Award.



Jinsong Wu (SM'11) received the Ph.D. degree in electrical and computer engineering, Queen's University, Kingston, Canada, in 2006.

He is currently with the Department of Electrical Engineering, Universidad de Chile, Santiago, Chile. He was the leading Editor and a coauthor of the comprehensive book, entitled *Green Communications: Theoretical Fundamentals, Algorithms, and Applications* (CRC Press, Sep. 2012.)

Dr. Wu is elected as a Vice Chair Technical Activities, the IEEE Environmental Engineering Initiative. His paper has won 2017 IEEE Systems Journal Best Paper Award. He was the Founder and Founding Chair of the IEEE Technical Committee on Green Communications and Computing. He is also the co-founder and founding Vice-Chair of the IEEE Technical Committee on Big Data. He is the Founder and Editor of the IEEE Series on Green Communication and Computing Networks in IEEE Communications Magazine. He is Area Editor in the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He was Series Editor in the IEEE JOURNAL OF SELECTED AREAS ON COMMUNICATIONS (JSAC) Series on Green Communications and Networking.



Yang (Cindy) Yi (SM'17) received the B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA.

She is currently an Assistant Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. Her research interests include very large scale integrated circuits and systems, computer aided design, and neuromorphic computing.

Dr. Yi is currently an Associate Editor for cyber journal of selected areas in microelectronics and has been serving on the editorial board of international journal of computational & neural engineering. She has been selected as technical program committee members for various international conferences and symposiums. She is the recipient of 2016 Miller Professional Development Award for Distinguished Research, 2016 United States Air Force (USAF) Summer Faculty Fellowship, 2015 NSF EPSCoR First Award, and 2015 Miller Scholar.