Emerging Applications of Reservoir Computing in Cyber Physical Systems Security

Kian Hamedani

Email : hkian@vt.edu

Committee Members: Dr. Marius K. Orlowski(chair), Dr. Lingjia Liu, Dr. Yang (Cindy) Yi

Abstract

In this report a new method for attack detection of smart grids with reservoir computing (RC) is introduced. 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 readout 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 preliminary results show 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 10000 simulations. The future step is to use more sophisticated attack models like dynamic attacks and also propose deep structures of RC models to improve the performance of attack detection when we are dealing with more sophisticated attacks like dynamic attacks. Looking at other aspects of cyber physical systems(CPS) security, such as high performance computing(HPC), smart infrastructures and internet of things (IoT) would also be another approach to follow in the future.

Index Terms

Smart grids; attack detection; neuromorphic computing; reservoir computing; delayed feedback networks; temporal encoder; state vector estimation

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 meters per second 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].

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].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 [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, 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 (PMUs) have been extensively investigated in order to improve the performance of state vector estimation [30] , [31]. Extended distributed state estimation (EDSE) was studied by Cramer *et al*. [32]. EDSE uses graph partition 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.

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 (ESN) [12] are two most popular RC systems. The difference between LSM and ESN is that, LSM 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

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

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], [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 (LFP) is used to encode the information [23]. Studies show that interspike interval encoding carries more information than rate encoding [24], [25]. Therefore, in this report, 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 (CPS) efficiently and effectively using RC. To be specific, in this report, 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.

II. 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 report. 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 **??**. The data used consists of 10000 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 non-attacked 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 leaky-integrate and fire (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 applied to the LIF neuron.

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 (ADC) and digital-to-analog (DAC) converters, is avoided when interfacing with analog signals [38]–[41]. In general, analog implementation has the advantage of implicit real-time operation, resulting in small design area and low power [42]–[44]. 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

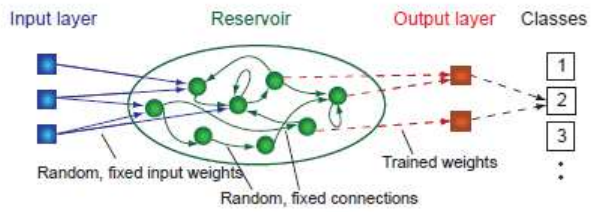

Fig. 1. A Reservoir Computing System [15].

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.

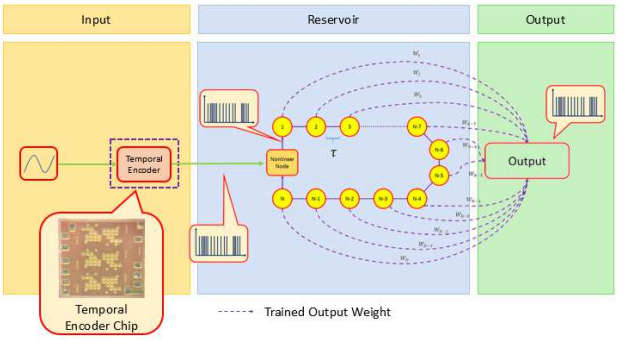

Fig. 2. Hardware implementation of delayed feedback reservoir system.

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 multi-layer 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 is then used to evaluate the performance of the system.

B. 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 [45]. MATPOWER 5.1 can be used to produce the smart grids' measurement matrix [46]. 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. [47].

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

$$z = Hx + n. \tag{1}$$

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, $\breve{z}$, becomes

$$\breve{z} = Hx + a + n. \tag{2}$$

We assume that the attack is a Gaussian random vector with 0.05 variance [47].

C. 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 [48]. 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 non attacked 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 non-hidden 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 [49], an equation was introduced to convert the spike trains to the analog current:

$$I^i = \sum_{t^j} K\left(t - t^j\right) \mathcal{H}\left(t - t^j\right), \tag{3}$$

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 [49]; and

$$K(t - t^j) = V_0 \left(\exp(-((t - t^j)/\tau_s)) - \exp(-((t - t^j)/\tau_f))\right), \tag{4}$$

D. 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], 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]:

$$\hat{y}(k) = \sum_{i=1}^{N} w_i \times x\left[k\tau - \tau/N(N - i)\right], \tag{5}$$

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. .Fig.3 shows the block diagram of our RC-based attack detection algorithm.

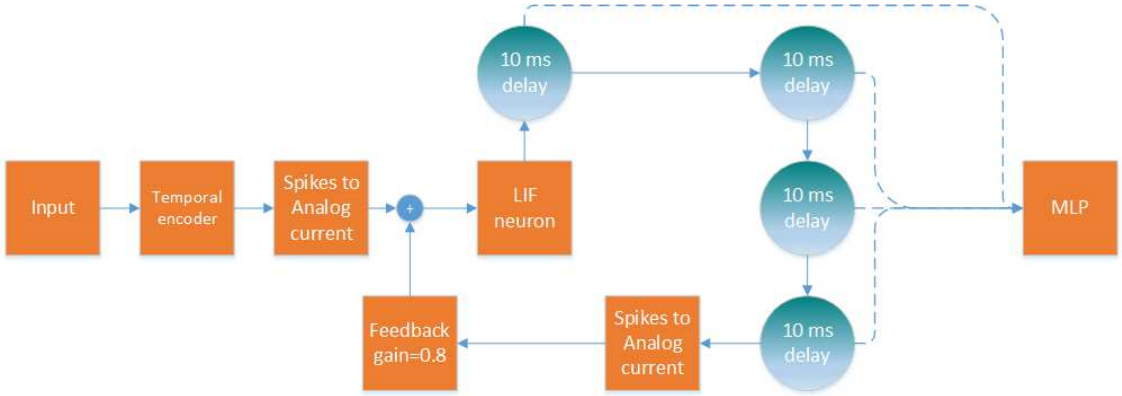

Fig. 3. Block diagram of the DFN+MLP system for attack detection.

E. State Vector Estimation

As mentioned in Section II-B, $\rho = \|\breve{z} - H\hat{x}\|_2^2$ needs to be computed for SVE. If the value of ρ exceeds a predefined threshold value, it is said that an attack has occurred, non-attack 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

$$\text{Accuracy} = (\text{TP} + \text{TN}) \,/\, (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \,, \tag{6}$$

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

III. Preliminary Performance Evaluation

As the main evaluation results, Figs. 4 & 5 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. 4 & 5.

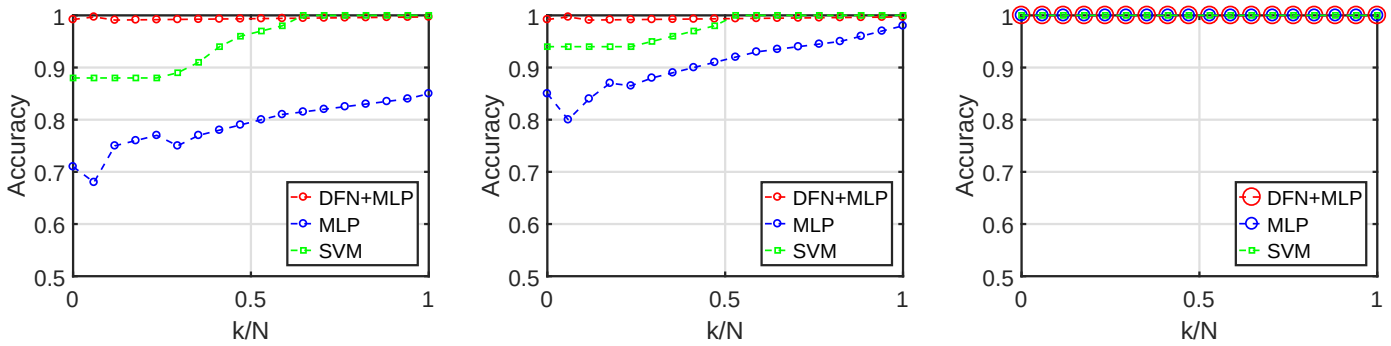

Fig. 4. Accuracy of direct attack detection for three different methods, a=0.1,1,10.

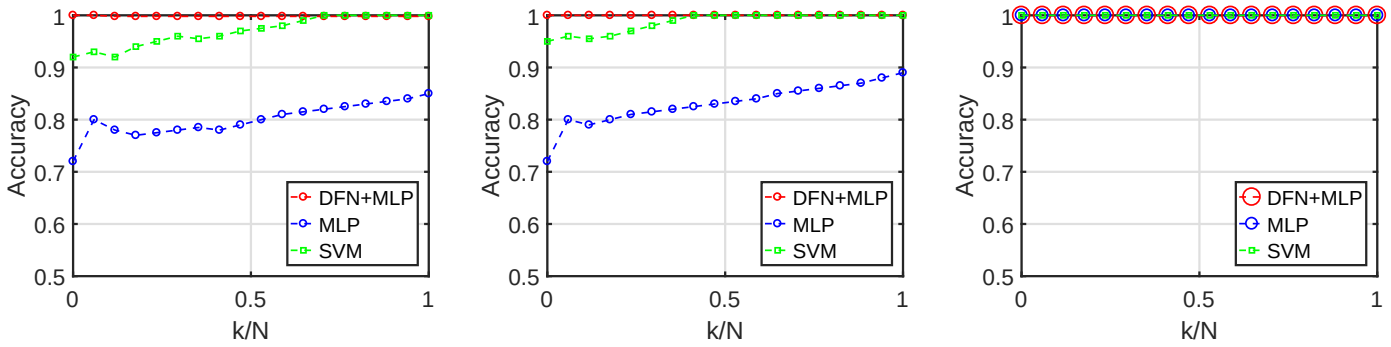

Fig. 5. Accuracy of hidden attack detection for three different methods, a=0.1,1,10.

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. 4 & 5 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. 4 & 5.

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).

IV. Future Plan

The attack that has been used in the priliminary work, is not time variant. The next step would be to use a more challenging attack which is time variant. This attack is called dynamic attack. Dynamic attacks are performed in a way that the state of the smart grid system will gradually be manipulated toward the state desired by the adversary [51]. In order to design this attack, the attack vector has to be defined as a function of time

$$z = Hx(t) + n + a(t). \tag{7}$$

In order to deal with more sophisticated attacks, more sophisticated networks have to be provided. To do so, a deep structure of DFNs will be proposed that will have more computational power to detect the attacks. For DFR computing systems, depth in time arises from the delayed signal that combines with the new input. However, for both RNNs and DFRs, single reservoir does not create any depth in space. Similar to stacking feedforward neural networks in deep learning field, depth in space could also be achieved by stacking multiple reservoirs on top of each other between the input and output layers. Along with the analog implementation of DFR, we investigate the possibility of merging deep learning and DFR. Two deep DFN structures, deep DFN and MI-deep DFN, are proposed. In the deep DFN model, the output from the previous layer will be injected into the successive reservoir layers.

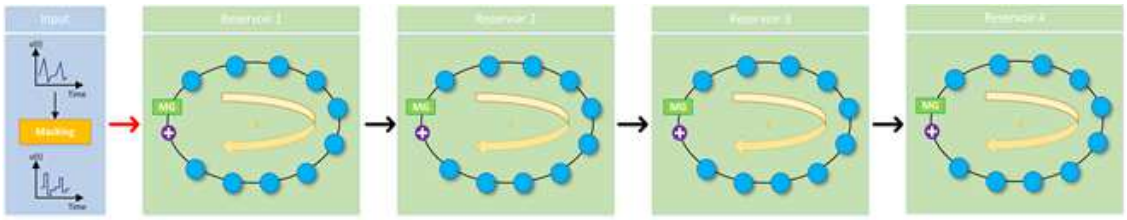

Fig. 6. Deep Structure of DFN.

Proposing deep structures of other types of RC models like ESN is also another approach for the future steps to deal with sophisticated attacks and compare the results with deep structure of DFNs.

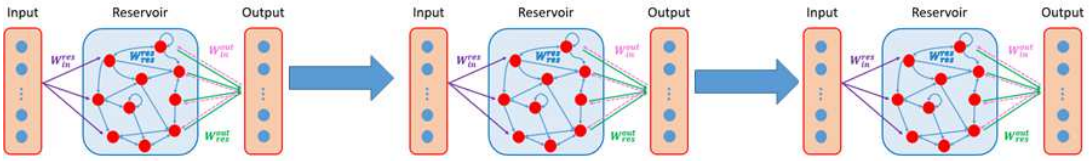

Fig. 7. Deep Structure of ESN [52].

Finally we will see if these models can also be applied to other cyber physical systems security domains like HPC and internet of things (IoT). CPS and IoT have significant overlaps. CPS emphasizes the tightly coupled integration of computational components and physical world. IoT has an emphasis on the connection of things with networks.If an IoT system interacts with the physical world via sensors or actuators, then it can also be classified as a CPS [53].

V. Conclusion

In this report, 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 sates 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 work 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. In the future steps, more complex attack models like dynamic attacks will be used. Deep structures of RC systems, like deep DFN or Deep ESN will be applied on more complex attack models in order to extract the spatio-temporal correlation among smart grid meters better. After that, other aspects of the CPS security, such as HPC systems security and IoT will also be studied and we will see if the proposed models for smart grids security are able to be applied on other security domains or not .

VI. Publications

During my Phd studies, my research will mainly be focused on different emerging applications that RC systems might have, in different areas like cyber-security and wireless communications. So far, I have had some pulications that are listed below. Some of them have been published and some of them are in preparation.

- Hamedani, K., Liu, L., Atat, R., Wu, J., and Yi, Y. (2018). ”Reservoir computing meets smart grids: attack detection using delayed feedback networks”. IEEE Transactions on Industrial Informatics, 14(2), 734-743.
- Zhao, C., Hamedani, K., Li, J., and Yi, Y. (2018). ”Analog Spike-Timing-Dependent Resistive Crossbar Design for Brain Inspired Computing”. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 8(1), 38-50.
- Li, J., Liu, L., Zhao, C., Hamedani, K., Atat, R., and Yi, Y. (2017). ”Enabling Sustainable Cyber Physical Security Systems Through Neuromorphic Computing”. IEEE Transactions on Sustainable Computing.

- Li, J., Zhao, C., Hamedani, K., and Yi, Y. (2017, May)." Analog hardware implementation of spike-based delayed feedback reservoir computing system". In Neural Networks (IJCNN), 2017 International Joint Conference on (pp. 3439-3446). IEEE.
- Bai, K., Li, J., Hamedani, K., and Yi, Yang (Cindy). (2018, June). "Enabling a New Era of Brain-inspired Computing: Energy Efficient Analog Delayed Feedback Reservoir Computing System Design". Accepted , in Design Automation Conference (DAC), 2018 55th ACM/EDAC/IEEE . IEEE.
- K. Hamedani, Lingjia Liu, Jialing Li, Shiyan HU and Yang Yi, "Detecting Dynamic Attacks in Smart Grids: A Spiking Neural Network-Based Approach", in Preparation.

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, June 2015.

[2] E. Camacho, T. Samad, M. Garcia-Sanz, and I. Hiskens, "Control for renewable energy and smart grids," in *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, April 2016.

[4] J. Wu, S. Guo, J. Li and D. Zeng, "Big Data Meet Green Challenges: Big Data toward Green Applications," *IEEE Systems Journal.*, 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 Systems Journal.*, vol. 10, no. 3, pp. 873–887, Sep 2016.

[6] S. Soter and R. Wegener, "Development of induction machines in wind power technology," in *in Proc. IEEE Int. Electric 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 *IEEE Intl. Joint Conf. on Neural Netw. (IJCNN)*, July 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.*, accepted, Oct. 2017.

[12] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," in *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 *Eur. Symp. Artif. Neural Netw., Bruges, Belgium*, 2007, pp. 471–482.

[14] X. Hinaut and P. Dominey, "On-line processing of grammatical structure using reservoir computing," in *Intl. Conf. on Artificial 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, M. C. Soriano, G. V. Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," in *Nature Commun. 2*, Aug. 2011.

[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," in *IEICE Nonlinear Theory and Its App.*, 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 *3rd ACM Intl. Conf. on Nanoscale Computing and Commun.*, pp. 1–5, Sep. 2016.

[19] C. Zhao, B. T. Wysocki, C. D. Thiem, N. R. McDonald, J. Li, L. Liu, and Y. Yi, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," in *IEEE Trans. on Multi-Scale Computing Systems*, vol. 2, no. 4, Sep. 2016, pp. 256–276.

[20] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," in *Trends in Neurosciences*, vol. 33, no. 3, Jan. 2010, pp. 1964–1974.

[21] D. Reich, F. Mechler, K. P. Purpura, and J. Victor, "Interspike intervals, receptive fields, and information encoding in primary visual cortex," in *J. of Neuroscience*, vol. 20, no. 5, Mar. 2000, pp. 1964–1974.

[22] S. M. Chase and E. Young, "First-spike latency information in single neurons increases when referenced to population onset," in *National Academy of Sciences of the United States of America*, vol. 104, no. 12, Jan. 2007, p. 51755180.

[23] J. Lisman, "The theta/gamma discrete phase code occuring during the hippocampal phase precession may be a more general brain coding scheme," in *Hippocampus*, vol. 15, no. 7, 2005, pp. 913–922.

[24] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," in *Biophysical Journal*, vol. 1, no. 6, 1961, pp. 445–466.

[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," in *Neuron*, vol. 61, no. 4, 2009, pp. 597–608.

[26] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *ACM Trans. on Inf. and System Security (TISSEC)*, 2011.

[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, July 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 Tuts.*, vol. 19, no. 1, pp. 397–422, 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, 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*, June 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," *Information Systems*, 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 Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[35] Y. Liu and X. J. Wang, "Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron," in *J. of Comput. Neuroscience*, 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.*, June 2017.

[37] J. Li, C. Zhao, and Y. Yi, "Energy efficient and compact analog integrated circuit design for delay-dynamical reservoir computing system," in *IEEE Intl. Joint Conf. on Neural Netw. (IJCNN), invited*, 2017.

[38] C. Zhao, B. T. Wysocki, Y. Liu, C. D.Thiem, N. R. McDonald and Y. Yi, "Spike-time-dependent encoding for neuromorphic processors," in *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 3, 2015, pp.23.

[39] Y. Yi, L. Yongbo, Y. Liu, B. Wang, X. Fu, F. Shen, H. Hou, and L. Liu, "FPGA based spike-time dependent encoder and reservoir design in neuromorphic computing processors," in *Microprocessors and Microsystems*, vol. 46, 2016, pp.175-183.

[40] Y. Yi, Y. Zhou, X. Fu, and F. Shen, "Modeling differential through-silicon-vias (TSVs) with voltage dependent and nonlinear capacitance," in *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Microelectronics (JSAM)*, vol. 3, no. 6, 2013, pp.234-241.

[41] Y. Yi, P. Li, V. Sarin, and W. Shi, "A preconditioned hierarchical algorithm for impedance extraction of three-dimensional structures with multiple dielectrics.," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11 2008, pp.1918-1927.

[42] L. Smith, *Neuromorphic Systems: past, present and future*. Springer, 2010.

[43] 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.

[44] K. Ramanaiah and S. Sridha, "Hardware implementation of artificial neural networks," *i-Manager's Journal on Embedded Systems*, vol. 3, no. 4, p. 31, 2014.

[45] 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, June 2011.

[46] 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,

[47] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proc. Intl. Joint Conf. on Neural Netw. (IJCNN)*, 2016. Dec 2011.

[48] A. Patrascu and V. Patriciu, "Cyber protection of critical infrastructures using supervised learning," in *Proc. IEEE Control Systems and Computer Science (CSCS)*, May 2015, p. 461C468.

[49] Q. Yu, H. Tang, K. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning heteroassociation of spatiotemporal spike patterns," *Plos one*, vol. 8, no. 11, p. e78318, 2013.

[50] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proc. Intl. Joint Conf. on Neural Netw. (IJCNN)*, 2016.

[51] J. Kim, L. Tong and R.J. Thomas, Dynamic attacks on power systems economic dispatch, in *Signals, Systems and Computers, 2014 48th Asilomar Conference*, pp. 345–349.

[52] C. Gallicchio, A. Micheli, L. Pedrelli "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 2, no. 268, pp. 87–99, 2017.

[53] D. Yao, X. Shu, L. Cheng and S.J. Stolfo "Anomaly Detection as a Service: Challenges, Advances, and Opportunities," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 3, no. 3, pp. 1–173, 2017.