

Power System Distributed Dynamic State Prediction

Md. Ashfaque Rahman, *Student Member, IEEE*, and Ganesh Kumar Venayagamoorthy, *Senior Member, IEEE*

Abstract—The security of the power system can be enhanced with the prediction of the dynamic state variables. To increase the security, a distributed predictor is developed based on the Elman Recurrent Neural Network (ERNN) in this study. To develop a scalable distributed predictor, the whole network is divided in a number of ERNNs. They take the current and the previous actual states from its own and its neighbors to predict the near future values of the states. The concept is inspired from the Cellular Computational Network (CCN) framework. Each ERNN is a cell in the CCN framework. Through simulation, the effectiveness of the proposed network is shown for a single step and a multi-step predictor and their accuracies are analyzed for IEEE 68-bus system.

Index Terms: Back-propagation through time, cellular computational network, dynamic systems, Elman recurrent neural network, state prediction.

I. INTRODUCTION

Modern electric power is a complex interconnected system spread over thousands of miles. The protection of the grid depends on frequent monitoring, and proper analysis of the system states. To have a clear view of the states, estimators are used to remove the measurement errors [1]. Contingency analysis is used to direct the operation in a safer region by anticipating any upcoming event [2]. Load forecasting is used to keep the production ready for probable load change [3].

Though the importance of the state predictor can be easily understood, it is not in practice as of today. Perhaps the reason is the slow rate of measurement collection that makes it a random process. It cannot be predicted if it becomes random. But, with the advent of the Phasor Measurement Units (PMUs), the data collection rate has increased tremendously [4]. With this increased rate, the dynamics of the states can be explored properly. Though the PMUs are not deployed at mass scale so far, it is assumed that they will take the place of traditional power meters in future.

Prediction is not completely new in power systems. In fact, it is used as an intermediate stage of the Kalman filter based state estimator [5]. It is a single step prediction that is used in the filtering stage. It is difficult to extend the result for multiple steps. Another important drawback of the Kalman filter is that it is a non-distributable method. Predictor is also used as a part of situational awareness [6], [7]. A good number of work is also done on predicting voltage sags [8].

Neural Networks (NNs) have always been good candidates for prediction. Due to the process of training, they adapt to

unknown functions easily. A lot of work is done on the use of the NNs to predict different systems across numerous fields of research including traffic system, bankruptcy, protein structure etc. [9]–[11]. It is also used in power systems. An artificial NN is used in developing the predictor of Kalman filter in [12].

In this paper, a dynamic state predictor is developed using the concept of cellular framework and Elman network. Cellular Computational Network (CCN) is a framework to split the whole network in small cells [13]. It ensures the distributability of the computation process. On the other hand, Elman network is a simple recurrent neural network (RNN) with an extra layer of memory [14]. It is very suitable for predicting the dynamic systems.

The main contributions of this paper are,

- A distributed state predictor using ERNN is developed with inspiration from the cellular computational network framework. The ERNNs are trained using back-propagation through time method.
- The performance of the predictor is tested with IEEE 68-bus system. The accuracy for a single-step as well as for a multi-step predictor is analyzed.

The rest of the paper is organized as follows. The background of the CCN, and the Elman network and its training are discussed in Section II. The nature of the state variable and the application of the predictor are analyzed in Section III. The specifics of the setup of the experiment are described in Section IV. The simulation results are analyzed in Section V. The paper is concluded with the plan for future work in Section VI.

II. BACKGROUND

Though the concept of state estimation is not new, prediction is not used in power system. As mentioned earlier, it has a special use in Kalman filtering which is neither distributable nor extensible to multi-step.

A. Cellular Computational Network

Neural network is widely used to predict states of dynamic systems. It has input, hidden and output layers which are connected with different weights. With the training session, the weights are trained and the trained weights are used for testing unknown inputs. There are many versions of the canonical NN to serve different purposes.

One of the major drawbacks of the NNs and its variants is the size of the weight matrix. They directly depend on the number of input, hidden and the output neurons. For a $10 \times 20 \times 5$ basic network, the size of input layer weights \mathbf{w} , and output layer weights \mathbf{v} become 10×20 , and 20×5 respectively. For very large systems, the numbers of neurons at the input and the output layer are very large, and so are the sizes of \mathbf{w} , and \mathbf{v} . In effect, the network becomes inoperative.

The authors are with Real Time Power and Intelligent Systems Laboratory, Clemson University, Clemson, SC, USA (marahma@clemson.edu, gkumar@ieee.org). Dr. Venayagamoorthy is also with the School of Engineering, University of Kwazulu-Natal, Durban, South Africa. This work is supported in part by the US National Science Foundation (NSF) under grant #1312260, and the Duke Energy Distinguished Professor Endowment Fund. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF and Duke Energy Foundation.

To solve the problem of distributability, the network is divided in smaller connected networks. As the effects of the distant states are very small, their impacts can be ignored. This introduces a new network named cellular computational network (CCN). In fact, CCN is a class of sparsely connected dynamic recurrent networks (DRNs) as shown in Fig. 1. By proper selection of a set of input elements for each output variable in a given application, a DRN can be modified into a CCN which significantly reduces the complexity of the neural network and allows use of simple training methods for independent learning in each cell thus making it scalable [13].

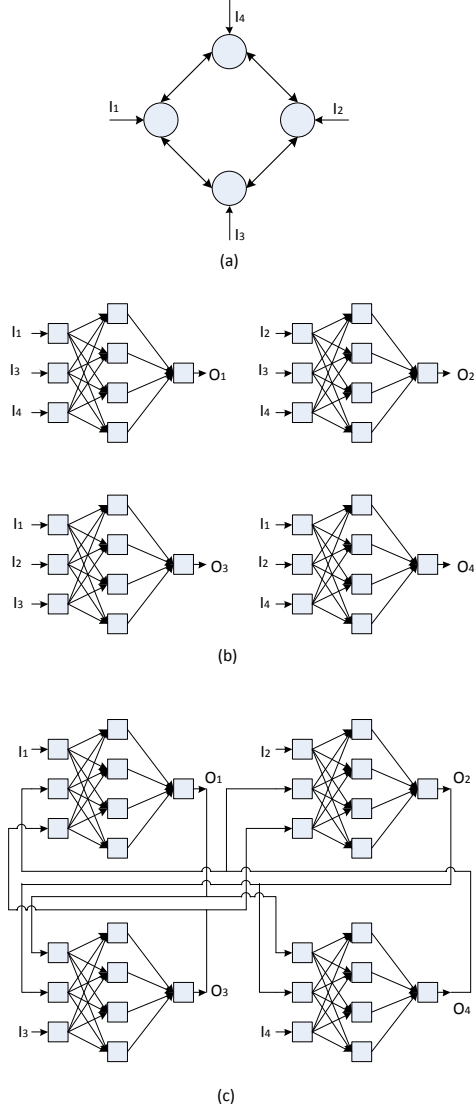


Fig. 1: An example of CCN using a four node network. The individual cells are formed in (b). An equivalent of the recurrent network is shown in (c) [13].

Though the basic CCN requires the connection of the cells and exchange of their results, the cells are kept separate in this study. As a result, the whole predictor is made up of some separate cells with only one output as shown in Fig. 2. The

analysis of the connected network is left as a future work.

B. Elman Recurrent Neural Network

One of the most well known variants of NN is Elman Recurrent Neural Network (RNN). In general, the RNNs are useful for tracking the dynamic states. Beside the input, output and the hidden layer, Elman network has a context layer as shown in Fig. 2. The output of the hidden layer are fed in itself before being multiplied with the output weights. Let the main input be denoted with x_i , and the context layer input be denoted with x_c .

The full input $x_F(t) (= [x_i^T x_c^T]^T)$ is multiplied with the input layer weight matrix w and reaches the hidden layer. A bias is added with the input layer. Let the hidden layer include N neurons. The output of the hidden layers, a are passed through normalizing units and produces d . The normalized values d are saved in the context layer for using in the next time step. The next input $x_F(t + 1)$ is formed with input and context layer neurons. The values of d are also multiplied with the output layer weights v to form \hat{y} for the current step t .

$$a = w x_F$$

$$d_n = \frac{1}{1 + e^{-a_n}} \quad \text{for } n = 1 \dots N \quad (1)$$

$$\hat{y} = v^T d$$

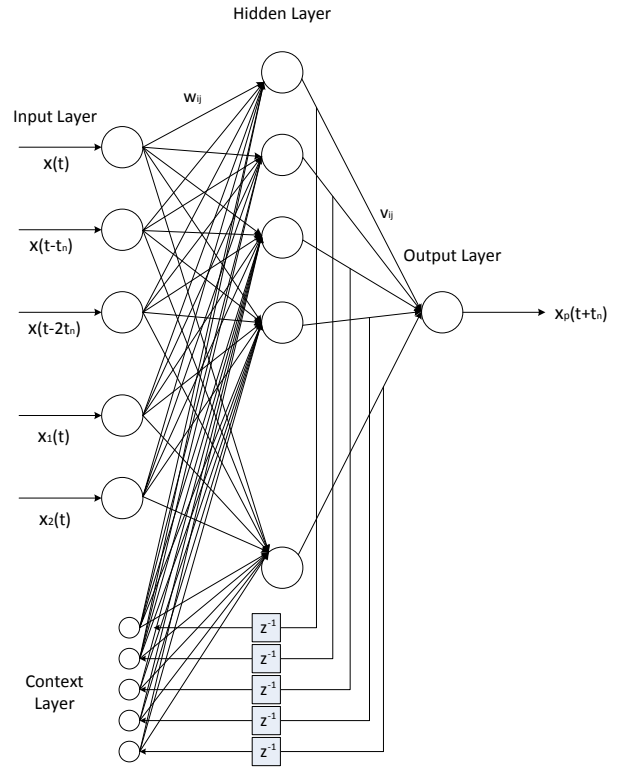


Fig. 2: The structure of the Elman recurrent neural network. The output of the hidden layer is fed back in the next time step.

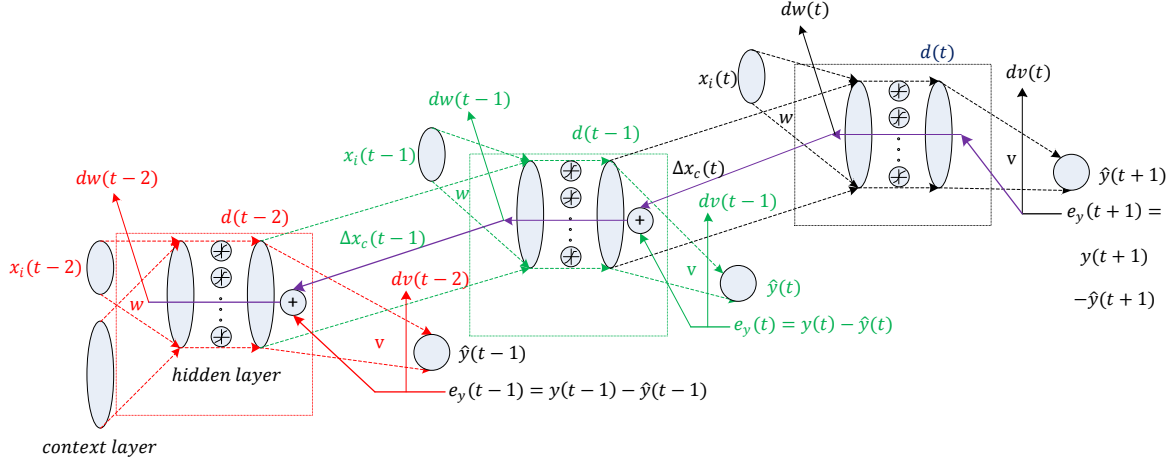


Fig. 3: The working principle of the back-propagation through time. A two step unfolded network showing back propagation of error at time t .

C. Training of the Elman Network

Before using, the network needs to be trained with a known sequence of input and its corresponding output. The training of the Elman network is different from the training of the canonical NN. It requires unfolding of the recurrent network to basic networks and back-propagation is applied on those networks. This special method is known as back-propagation through time (BPTT) [15].

1) *Basic back-propagation*: Back-propagation is a very simple and effective tool for training a non-recursive network. The known sequence is fed in the network and an output is found. The output is compared with the expected output and the error e_y is determined. The error is fed back to the network and the contribution of different parts i.e. e_d , and e_a are calculated. Then the weights are updated according to the contributions. It can be done in batch processing, i.e. the errors are summarized for all training samples and the weights are updated with that. The process is described in the following set of equations.

$$\begin{aligned}
\mathbf{e}_y &= \mathbf{y} - \hat{\mathbf{y}} \\
\mathbf{e}_d &= \mathbf{v}^T \mathbf{e}_y \\
e_{an} &= d_n(1 - d_n)e_{dn} \quad \text{for } n = 1 \dots N \\
\Delta \mathbf{v} &= \gamma_m \Delta \mathbf{v} + \gamma_g \mathbf{e}_y \mathbf{d}^T \\
\Delta \mathbf{w} &= \gamma_m \Delta \mathbf{w} + \gamma_g \mathbf{e}_a \mathbf{x}_F^T \\
\mathbf{v} &= \mathbf{v} + \Delta \mathbf{v} \\
\mathbf{w} &= \mathbf{w} + \Delta \mathbf{w}
\end{aligned} \tag{2}$$

Here, γ_g , and γ_m are the learning gain and the momentum gain respectively. γ_g determines how fast the network should learn from one sample of the training sequence. On the other hand, γ_m determines the rigidity of the network to changes.

2) *Back-propagation through time*: Due to the recurrent nature of the Elman network, the training is different. BPTT is

an offline process of training the recurrent networks. A simple training method is shown in Fig. 3.

To integrate the effects of previous input, the network is unfolded upto a certain times, h . This is the depth of the network. In Fig. 3, the process of BPTT is shown for $h = 3$. For a single time t , the inputs are fed for $\mathbf{x}_i(t-2)$, $\mathbf{x}_i(t-1)$, and $\mathbf{x}_i(t)$. The weights \mathbf{w} , and \mathbf{v} remain the same throughout the forward process. The error for prediction of the last stage $\mathbf{e}_y(t+1)$ is fed back to update \mathbf{v} , and \mathbf{w} . For \mathbf{v} , the update is similar to (2). However, as \mathbf{w} is affected by the values of \mathbf{d} , the values of $\mathbf{e}_d(t-1)$ will be affected with both $\Delta \mathbf{d}(t)$, and $\mathbf{e}_y(t)$. If $\Delta \mathbf{x}_F(t)$ represents the corresponding change of input, from the gradient based analysis, the expression of it is found as follows,

$$\Delta \mathbf{x}_F(t) = \mathbf{e}_a^T(t) \mathbf{w} \tag{3}$$

$\Delta \mathbf{x}_c(t)$ is the part of $\Delta \mathbf{x}_F(t)$ corresponding to the context layer. It gets added with the effect of \mathbf{e}_y to form the complete effect on $\mathbf{d}(t-1)$,

$$\mathbf{e}_d(t-1) = \Delta \mathbf{x}_c(t) + \mathbf{v}^T \mathbf{e}_y(t) \tag{4}$$

The process runs till the starting network and the necessary corrections are determined. Then the last one of them are used to update the values of \mathbf{w} and \mathbf{v} .

III. PREDICTION AND THE LATEST TECHNOLOGIES

There is an important relation between the rate of data collection and the effectiveness of the predictor. Only one step ahead prediction is useless for a fast collection rate. It is discussed in the following section.

A. Nature of the States

In the traditional SCADA system, measurements are taken at a very slow rate of around 1 sample per 2-4 seconds. Under this rate, the collected samples miss some important changes

and they may look random. So, the use of the static WLS estimator is logical for this rate.

However, the estimator is getting faster day by day, and it requires a faster rate of collection of data. With the slowest PMU rate, i.e., 30 samples per second, the measurements show a complete dynamic nature. From the actual values of some simulations under different situations, it is observed that the states never change abruptly except the case of a fault. Even during a fault, they do not reach the final value within one sample. The actual values of the voltage magnitude of a bus of 16-machine 68-bus NY-NE system are shown in Figure 4.

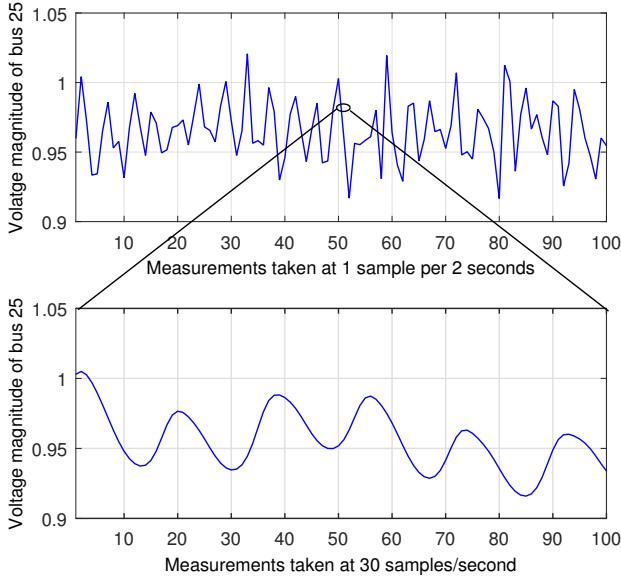


Fig. 4: The trend of one state under two different sampling rates. The upper one is taken at SCADA rate where the lower one presents the PMU rate.

As the SCADA rate cannot explore the dynamic natures of the states, a predictor becomes unsuitable for the purpose. With PMU rate, it can be an important part of the system.

B. Application of the state predictor

Any time ahead prediction can be helpful in the operation of the power system. The number of time steps depends on the time needed for the operator to take any action. Different actions require different amount of time to be taken. As a result, multiple predictor can run with different time steps to serve different purposes. It is important to remember that the accuracy of the prediction depends on the time step. If the prediction time is longer, the accuracy reduces.

The main application of the predictor can be in state estimation, contingency analysis, load forecasting, automatic voltage regulation, load frequency control, automatic generation control, economic dispatch etc.

IV. TEST POWER SYSTEM

The efficiency of the proposed predictor is tested in IEEE 16-machine 68-bus NY-NE test system. It has a total of 135 states with 83 transmission lines as shown in Fig. 5. For each bus, there are two computation cells for estimating the

magnitude and the angle as shown in Fig. 6. As a result, there exists a total of 135 ERNN cells which are separate from each other. They take the states of all neighboring cells. The states are taken directly from the simulation of the 68-bus system in Real Time Digital Simulator. In practice, they can be taken from the state estimator.

A. Training Data

For both the single-step and the multi-step predictions, the training is done with 20000 data sampled at 30 Hz from the simulation of the system. The samples are iterated over 100-500 epochs to fit the training signals properly. To ensure a good amount of disturbance in the system, pseudo random binary signals (PRBS) are applied in the excitation control of the generators while collecting the measurements. A part of the 16 PRBS signals for 16 generators is shown in Fig. 7. The random changes take randomness in the training signals. The corresponding changes in the outputs of the generators are shown in Fig. 8.

The voltage magnitudes and the angles are separated. So, for each cell, there are two separate networks at each cell. Each cell uses the previous values of the corresponding state as well as the current value of it to predict the next value. If it predicts for n step ahead result, it uses the previous values of $t - t_n$ and $t - 2t_n$. For the neighbors, it only uses the data of current values.

The number of hidden layer neurons are taken to be double of the input layer. The number of the neurons of the context layer is equal to that of the hidden layers. As the network of each cell is producing either the magnitude or the angle, there is only one output neuron for each network.

The context layer is initialized with random weights. The starting inputs for this layer are zeros. As shown in Fig. 2, there is a one step delay between the output and the input part of the context layer.

B. Testing Data

The testing is done with 15428 samples of data taken under the same condition as of the training data. It is also initiated with zero values in the context layer. Though it gives some incorrect results in the beginning, the effects wear out very soon.

V. SIMULATION RESULTS

Accuracy is the most important quality of any predictor. Though the accuracy of the voltage magnitude is well enough, the angles suffer a lot. The reason is the very low variation of phase angles over time. It makes the tracking difficult for the predictor.

To measure the accuracies of the predictors, Mean Absolute Percentage Error (MAPE) is taken for a single state over a specific number of time samples. MAPE is defined as,

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{V_{jt} - \hat{V}_{jt}}{V_{jt}} \times 100\% \right| \quad (5)$$

Here, MAPE is taken over n time steps for bus j . V_{jt} , and \hat{V}_{jt} represents the true and predicted value of the corresponding state V .

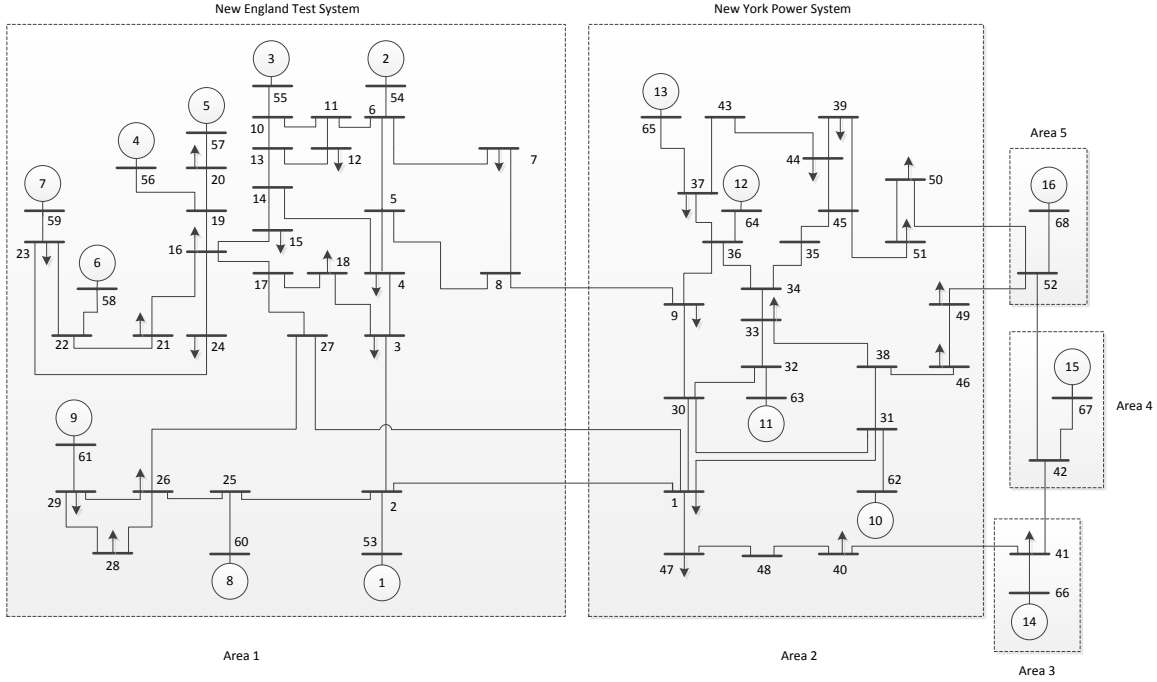


Fig. 5: IEEE 68-bus NY-NE test system.

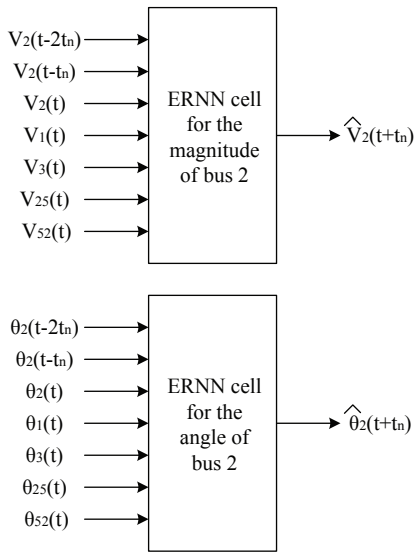


Fig. 6: Inputs and outputs for the two cells of bus 2.

A. Single step prediction

The accuracy is important for both the training session as well as for the testing session. With a repetition of 100-500 epochs, the training session usually gets an acceptable accuracy as shown in Fig. 9. For the testing session, the accuracies of the phase angles and the voltage magnitudes of bus 8, 25, and 52 are shown in Figs. 10, and 11. These are carried out for a single step prediction.

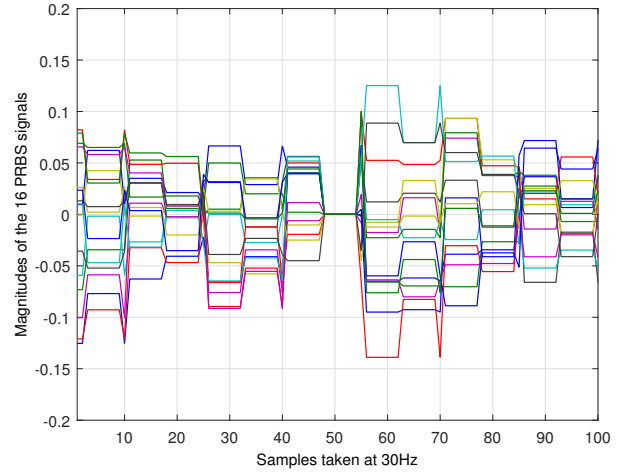


Fig. 7: PRBS signals used to perturb the generators in the power system.

B. Multi-step prediction

As the single step prediction is not suitable for some operations, multi-step predictions are needed. The training is done for multiple steps and the testing is shifted accordingly. The inputs are shifted according to the step size. For example, if the step size is six, the previous values are taken from $t - 6$ and $t - 12$. It is found that the multi-step prediction is less accurate than the single step prediction. A six step ahead prediction results are shown in Figs. 12, and 13.

The MAPEs and the standard deviation of the absolute

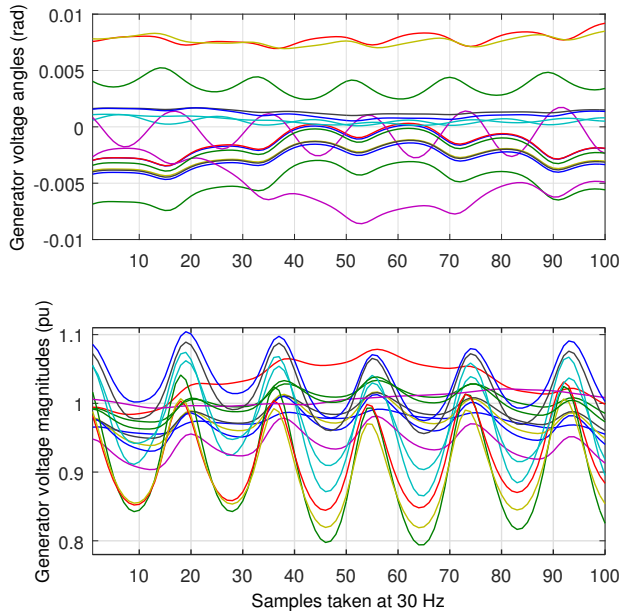


Fig. 8: The output of the generator voltages due to the PRBS signals.

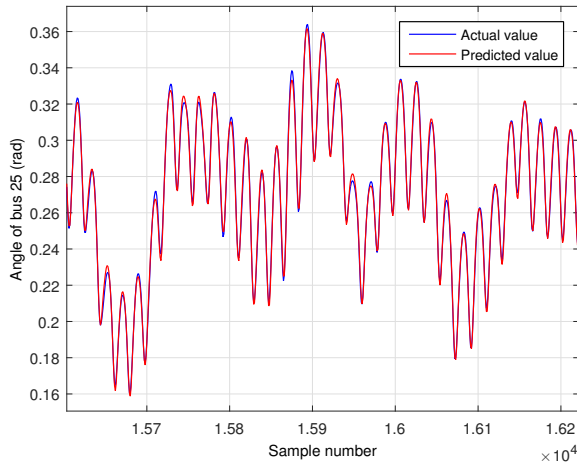


Fig. 9: A part of the final training values and the predicted training values.

prediction error (STD) for the single and the multi-step predictions are shown in Table 1.

TABLE I
MAPE±STD FOR PREDICTIONS

Bus	Single-step		Six-step	
	V	θ	V	θ
8	0.77±0.61	3.48±2.29	1.84±1.21	16.5±15.4
25	0.85±0.57	3.81±2.42	2.41±1.52	15.5±12.9
52	0.55±0.39	6.81±4.37	2.19±1.27	24.4±31.4

VI. CONCLUSION

A CCN inspired ERNN based dynamic state predictor is proposed for power systems and the simulation results are shown in this study. The results show that the network works well for the single state prediction. The accuracy of the multi-step predictor is also satisfactory.

The fully connected CCN is left as the future work. The single step prediction can be effectively used for distributed dynamic state estimation. It is also left as a future work. The relation between the step size and the accuracy of the predictor is also an important area of research. Though not explored, the accuracy may get better with some periods of steps.

REFERENCES

- [1] A. Abur and A. Exposito, *Power System State Estimation: Theory and Implementation*. New York, NY: Marcel Dekker Inc., 2004.
- [2] F. D. Galiana, "Bound estimates of the severity of line outages in power system contingency analysis and ranking," *IEEE Trans. Power Apparatus and Systems*, no. 9, pp. 2612–2624, 1984.
- [3] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, M. Damborg *et al.*, "Electric load forecasting using an artificial neural network," *IEEE Trans. Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.
- [4] J. Stewart, T. Maufer, R. Smith, C. Anderson, and E. Ersonmez. (2016, January) Synchrophasor security practices. [Online]. Available: https://cdn.selinc.com/assets/Literature/Publications/Technical%20Papers/6449_SynchrophasorSecurity_EE_20100913_Web.pdf
- [5] Z. Huang, K. Schneider, and J. Nieplocha, "Feasibility studies of applying kalman filter techniques to power system dynamic state estimation," in *International Power Engineering Conference, 2007*, Dec 2007, pp. 376–382.
- [6] K. Balasubramaniam, G. K. Venayagamoorthy, and N. Watson, "Cellular neural network based situational awareness system for power grids," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*,. IEEE, 2013, pp. 1–8.
- [7] —, "Situational awareness system for power grids," Graduate Research and Discovery Symposium (GRADS), 2013, paper 61.
- [8] M. R. Qader, M. H. Bollen, and R. N. Allan, "Stochastic prediction of voltage sags in a large transmission system," *IEEE Transactions on Industry Applications*, vol. 35, no. 1, pp. 152–162, 1999.
- [9] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction: neural network approach," *Transportation Research Record*, no. 1453, 1994.
- [10] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *IJCNN International Joint Conference on neural networks*, 1990, pp. 163–168.
- [11] D. Kneller, F. Cohen, and R. Langridge, "Improvements in protein secondary structure prediction by an enhanced neural network," *Journal of molecular biology*, vol. 214, no. 1, pp. 171–182, 1990.
- [12] P. Rousseaux, D. Mallieu, T. V. Cutsem, and M. Ribbens-Pavella, "Dynamic state prediction and hierarchical filtering for power system state estimation," *Automatica*, vol. 24, no. 5, pp. 595 – 618, 1988. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0005109888901082>
- [13] B. Luitel and G. K. Venayagamoorthy, "Cellular computational networks—A scalable architecture for learning the dynamics of large networked systems," *Neural Networks*, vol. 50, pp. 120–123, 2014.
- [14] J.-G. Wu and H. Lundstedt, "Prediction of geomagnetic storms from solar wind data using Elman recurrent neural networks," *Geophysical research letters*, vol. 23, no. 4, pp. 319–322, 1996.
- [15] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

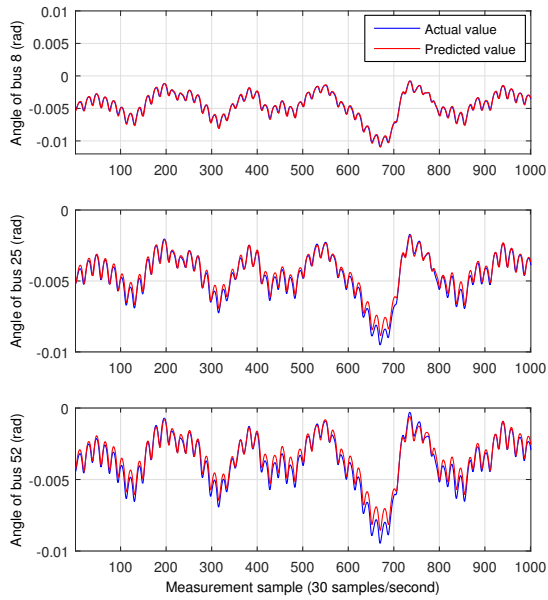


Fig. 10: A part of the testing values and the predicted testing values for a single step prediction of phase angle.

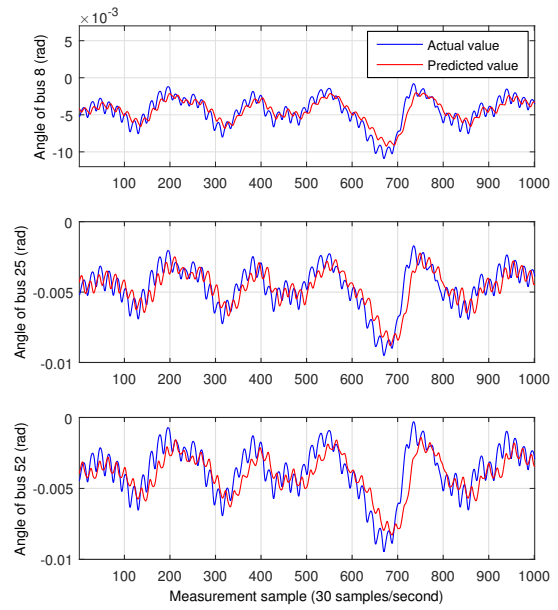


Fig. 12: A part of the testing values and the predicted testing values for a six step prediction of phase angle.

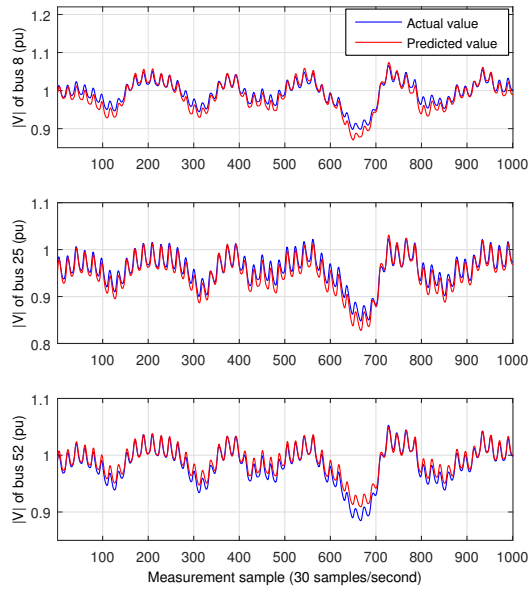


Fig. 11: A part of the testing values and the predicted testing values for a single step prediction of voltage magnitude.

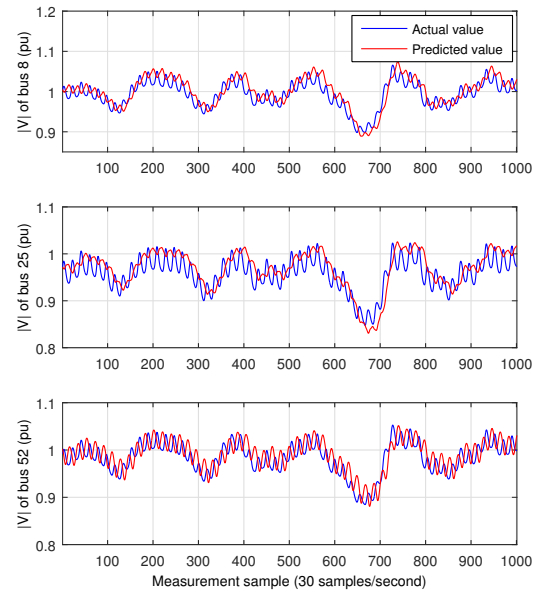


Fig. 13: A part of the testing values and the predicted testing values for a six step prediction of voltage magnitude.