IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2019, accepted January 14, 2020, date of publication January 17, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967638

Scalable Distribution Systems State Estimation Using Long Short-Term Memory Networks as Surrogates

ZHIYUAN CAO[1], YUBO WANG[2], CHI-CHENG CHU[1], AND RAJIT GADH[1]
[1]Smart Grid Energy Research Center, University of California at Los Angeles, Los Angeles, CA 90095, USA
[2]Siemens Corporate Technology, Princeton, NJ 08540, USA

Corresponding author: Zhiyuan Cao (oreki47@ucla.edu)

ABSTRACT Modern distribution systems are confronted by increasing penetration of distributed energy resources, making state estimation a critical application for distribution systems. However, existing state estimation schemes are often time-consuming and therefore, hard to scale up for large systems. In this context, this paper has proposed using a surrogate model to accelerate state estimations. Long-short-term memory (LSTM) recurrent neural networks have been applied to produce a fast yet coarse surrogate of the system states, which captures the temporal correlations between consecutive states. We have further applied an autoencoder to reduce the input size of LSTM networks, thereby shrinking LSTM network size and increasing the scalability of the proposed method. The surrogate states from LSTM are then fed into the forward/backward sweep state estimator as initial values. As a result, the state estimation convergence is accelerated by the LSTM surrogates. The proposed method is tested on IEEE 123-bus and 8500-node three-phase unbalanced test systems. Experimental results show that the proposed LSTM networks significantly reduce the computational time of distribution systems state estimation.

INDEX TERMS Distribution system state estimation, long short term memory, neural networks, surrogate model.

I. INTRODUCTION

State estimation (SE) is a backbone application in power systems. SE refers to the process of estimating system state variables using measurements such as Supervisory Control and Data Acquisition systems, Phasor Measurement Units (PMUs), and smart meters [1]. Traditionally, SE is deployed in transmission systems where the state variables are usually defined as the bus-level voltage magnitudes and phase angles. In recent years, the rising of distributed energy resources (DERs) brings increasing uncertainties to distribution systems. This trend calls for distribution system state estimation (DSSE) to provide system state information to monitor and manage the distribution systems.

In contrast to transmission systems, distribution systems are characterized by high R/X ratio, short-line, and unbalanced phases [2]. As a result, methods such as weighted least square (WLS) that are well-developed in transmission systems tend to suffer from poor convergence when applied to distribution systems [3], and there is a need to study state estimation schemes in distribution systems.

To this end, much research focus has been placed on DSSE and can be traced back to the 1990s [4], [5]. More recently, authors in [6] discussed three estimators, including WLS, weighted least absolute value, and Schweppe Huber generalized M estimators, and concluded that the WLS estimator tends to give the best performance. However, the results are under the assumption of balanced phases, and such an assumption does not hold in most distribution systems. In [7], the authors proposed a DSSE approach using smart metering data.

In [8], the authors proposed a method based on compressive sensing and several PMUs to provide system measurements. Because PMUs provide direct measurements of the states, the state space equation involving pure PMU measurements becomes linear, and the complexity of solving state estimation is significantly reduced. On the other hand, PMUs are extremely expensive compared to traditional

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

measurement devices. Hence, the optimal placement of PMUs for DSSE is also heatedly studied [9]–[11].

In most circumstances, the power flow equation is non-convex. Therefore, researchers have studied heuristic methods to solve DSSE. Authors in [12] proposed a hybrid method by integrating WLS and firefly algorithm. Reference [13] presented a three-phase DSSE based on particle swarm optimization algorithm. As reported in these papers, heuristic methods have an advantage of better accuracy compared to traditional WLS methods, but undesirable computational efficiency. As a result, it is challenging to implement DSSE based on heuristic methods into large distribution systems.

Finally, dynamic state estimation (DSE) has been attracting increasing attention in recent years. DSE involves a state prediction step and a state estimation step. Traditionally, DSE is based on Kalman filter (KF) to model the two steps together [14], [15]. More recently, approaches such as exponential smoothing [16] and recursive least square [17] are also proposed for solving the state prediction step. Both KF-based and regression-based methods rely on the quasi-state assumption, and this assumption does not hold in distribution systems with heavy DER penetrations. State forecasting based on shallow neural networks (NN) is also proposed [18]–[20]. NN-based methods do not require the quasi-state assumption. However, shallow NNs have limited scalability [21] and face computational efficiency and accuracy challenges when applied to large-scale distribution systems.

The aforementioned methods assume that there is measurement redundancy in the systems. However, existing distribution systems often do not have the hardware infrastructure to provide measurement redundancy. In this context, a three-phase DSSE method using forward/backward sweep has been proposed in [22] by taking advantage of the radial structure of most distribution systems. The forward/backward iterative methods can be implemented with a lack of measurement redundancy, which is often the case in distribution systems. Another way to deal with a lack of measurement redundancy is to rely on pseudo-measurements. Pseudo-measurements are historical measurements that often have a lower accuracy compared to physical ones. Hence, modeling of pseudo-measurements for higher accuracy is also studied [23], [24].

To conclude, the state-of-the-art DSSE schemes often involve many iterations and have low computational efficiency. On the other hand, modern DSSE is challenged by the growing size of the system. These trends call for a fast DSSE scheme that can be applied to distribution systems at scale.

In this context, we propose a DSSE scheme based on surrogate modeling. The surrogate model has been widely used in various domains, such as aircraft design and antenna design [25]. A surrogate model is built when the original problem is challenging to solve, or computational efficiency is low. The surrogate model provides a suboptimal result of the original problem while being significantly cheaper in terms of resource cost. Specifically, we have built an LSTM recurrent neural network as the surrogate models to provide an initial "guess" of the system states. The LSTM model takes previous states as inputs and estimates current state values. Using previous states as input allows LSTM networks to capture the temporal correlation between consecutive states and produce surrogates with better accuracy. However, the size of the inputs snowballs as the size of the system increases. As a result, a large LSTM network has to be built, which reduces computation efficiency. To overcome this huddle, we advocate using autoencoder to compress the dimension of LSTM input, which in turn decreases the size of the LSTM networks and improves its computational efficiency. Finally, using the surrogate states as a better "guess," the iterative forward/backward sweep state estimator could converge much faster. Our contributions are summarized as follows.

- We have developed an LSTM-based surrogate method that incorporates the temporal correlation between previous state information.
- We have applied an autoencoder to compress the input size of LSTM networks, thereby shrinking the size of LSTM networks and improving accuracy.
- We have proposed a surrogate-based DSSE scheme, which improves the computational efficiency and make the proposed DSSE scheme applicable to large systems.

The remainder of this paper is organized as follows. Section II reviews the formulation of the DSSE problem and introduces our DSSE scheme. In Section III, the LSTM surrogate model is presented together with the autoencoder. Section IV gives a brief description of the forward/ backward sweep solver. In Section V, numerical studies are performed on IEEE 123-bus and 8500-node systems. Finally, Section VI concludes the paper.

II. PROPOSED METHODOLOGY

A. PROBLEM DEFINITIONS

In a power system, the state space model with additive white noise can be expressed as follows.

$$z_k = f(x_k) + w_k, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector; $z_k \in \mathbb{R}^m$ is the measurement vector; $f(\cdot)$ is the vector-valued measurement function; k is the time index; w_k is the measurement noise. The DSSE aims to recover the system voltage given available real and pseudo-measurements. In general, real measurements include sub-station level bus voltage magnitudes and angles, PMU readings, and smart meters. And pseudo-measurements include historical data such as historical energy consumption average in a given period.

Because the measurement function $f(\cdot)$ is non-linear, there is no algebraic solution to the proposed problem. In distribution systems, the problem is usually solved numerically by first initializing the state vector. It is then followed by iteratively updating the state vector until it finally converges.

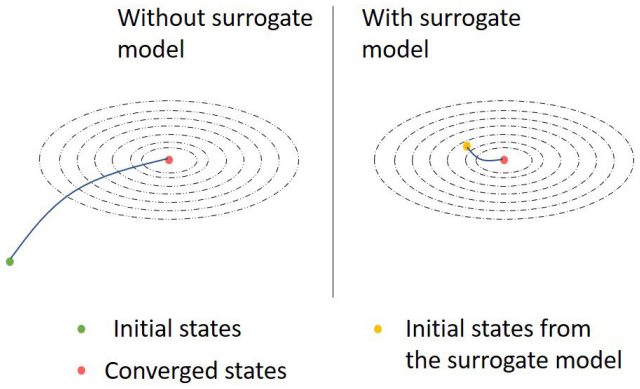

FIGURE 1. Illustration of applying the surrogate model to DSSE.

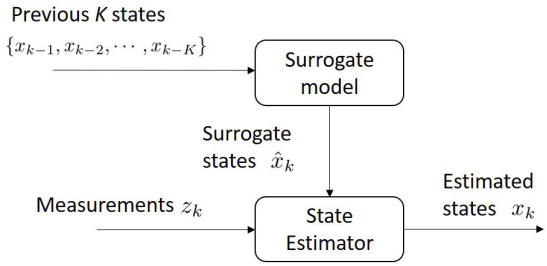

FIGURE 2. Architecture of the proposed method.

B. FAST DSSE THROUGH SURROGATES

The idea of applying the surrogates to DSSE is illustrated in Figure 1. Traditionally the state vector is initialized to a unit vector. Because a surrogate model can provide a suboptimal result to the original problem with minimal computation cost, the results from a surrogate model can be used to initialize the state vector to the vicinity of the converged states, thereby reducing the cost of iterations.

Based on this idea, we propose the following DSSE architecture, summarized in Figure 2. It consists of a surrogate model – the state surrogate, and a state estimator. At time k, given the information of the previous states, the surrogate model first provides a coarse but fast estimate of the state variables, denoted as $\hat{x}_k$. This coarse estimate is further used to initialize the state estimator. Together with the measurements z_k, the state estimator provides a refined estimate of the system states x_k.

III. STATE SURROGATES

A. PROBLEM FORMULATION

We start by formulating our state surrogate problem as

$$x_k = g(x_{k-1}, x_{k-2}, \ldots, x_{k-K}), \tag{2}$$

where K is the number of time steps we look back; $g(\cdot)$ is a mapping from previous K state variables to the current state variable. With the quasi-state assumptions, this formulation can be reduced to

$$x_k = A_k x_{k-1} + w_k, \tag{3}$$

where w_k is the process noise, and A_k is the state transition matrix. This simplified formulation is used in [16], [17], [26]. However, the quasi-state assumption may fail to hold in modern distribution networks with heavy DER penetrations.

The formulation displayed in Eq. 2 has several benefits and one shortcoming. First, it does not directly assume an explicit form of the relationship between current state variable x_k and previous states $\{x_{k-1}, x_{k-2}, \ldots, x_{k-K}\}$, but attempts to learn such a mapping $g(\cdot)$ from data. Hence, the quasi-state assumption is not required with this formulation. Second, this formulation incorporates K states from previous time-points to learn the temporal correlations of previous states.

On the other hand, because K previous states are used as input, the size of the input grows significantly with the system size and can impact the computational efficiency of the surrogate model. Section III-C further discusses how this challenge is overcame.

Our goal is to find the mapping $g(\cdot)$ through data. In another word, given a set of N historical states $Y = \{x_{k,1}, x_{k,2}, \ldots, x_{k,N}\}$ and its previous states $X = [\{x_{k-1,1}, x_{k-2,1}, \ldots, x_{k-K,1}\}, \{x_{k-1,2}, x_{k-2,2}, \ldots, x_{k-K,2}\}, \cdots, \{x_{k-1,N}, x_{k-2,N}, \ldots, x_{k-K,N}\}]$, the goal is to approximate $g(\cdot)$ such that the square loss is minimized, which is expressed as

$$\min\ L = \sum_{i=1}^{N} ||x_{k,i} - g(x_{k-1,i}, \cdots, x_{k-K,i})||_2^2. \tag{4}$$

B. THE LSTM MODEL

In this paper, the long short-term memory (LSTM) [27], [28] networks is built to approximate function $g(\cdot)$. LSTM belongs to the family of recurrent neural networks (RNNs), which can use their internal states (memory) to process sequential inputs, thus capable of learning temporal correlations.

Figure 3 presents a single LSTM block. Externally, an LSTM block takes the memory cell states c_{t-1} and the intermediate output h_{t-1} from previous time index, and subsequent input x_t of the current time index as the input of the block. Internally, an LSTM block contains four gates to maintain states. These are the forget gate f_t, the input gate i_t, the input node g_t and the output gate o_t. Finally, the intermediate output h_t and memory state c_t are fed to the LSTM block of the next time index. The gates and outputs are updated with the following equations

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{5}$$
$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{6}$$
$$g_t = \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g), \tag{7}$$
$$o_t = \sigma(W_{ox}x_t + W(oh)h_{t-1} + b_o), \tag{8}$$
$$c_t = g_t i_t + c_{t-1}f_t, \tag{9}$$
$$h_t = \phi(c_t)o_t, \tag{10}$$

where W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox}, W_{oh} are weight matrices and b_f, b_i, b_g, b_o are biases; σ and τ represents the sigmoid and tanh activation function respectively.

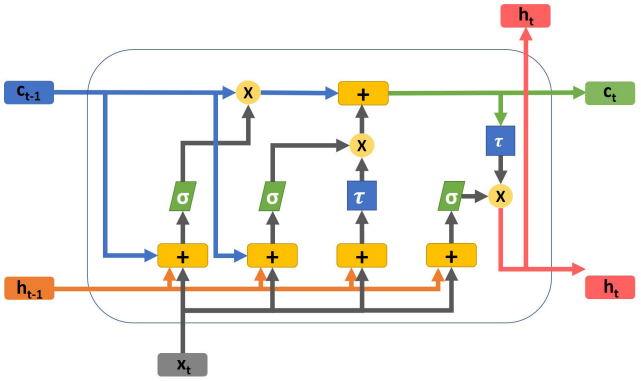

FIGURE 3. Structure of a single LSTM block.

The LSTM networks are sensitive to the magnitude of inputs. Therefore, we rescaled the input before feeding the input into the LSTM networks. After rescaling, the minimum and maximum values of the input are 0 and 1, respectively. This rescale transformation can be expressed as follows.

$$X_{rescaled} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{11}$$

With the LSTM network, we attempt to find the weights $W = \{W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox}, W_{oh}\}$ and biases $B = \{b_f, b_i, b_g, b_o\}$ such that Eq. 4 is minimized, i.e.,

$$\min_{\{W,B\}} L = \sum_{i=1}^{N} ||x_{k,i} - g(x_{k-1,i}, \cdots, x_{k-K,i})||_2^2. \tag{12}$$

In this paper, the Adaptive moment estimation (*Adam*) [29] optimizer is used to train the LSTM networks. *Adam* updates the learning rate automatically as the training goes on. Take weight W as an example. The *Adam* optimizer aim to iteratively update the weight parameters W. During iteration t, we first compute the first and second moment m_t and v_t with

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) h_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) h_t^2, \end{aligned} \tag{13}$$

where h_t and h_t^2 are the sum of gradients and Hessian of the loss function L at iteration t. β_1 and β_2 are two decay parameters. To initialize, we set $m_t = v_t = 0$. Then we compute the bias-corrected estimate of the first and second moment as

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}. \end{aligned} \tag{14}$$

Finally, the parameters W are updated through

$$W_t = W_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}, \tag{15}$$

where $\epsilon = 10^{-8}$ is applied to avoid division by zero. The *Adam* optimization does not change the learning rate α directly during the training. It controls the training speed with the two moments $\hat{m}_t$ and $\hat{v}_t$. The training continues until the weight parameters W converge.

We further apply dropout to the LSTM networks to increase robustness [30]. The dropout is realized by randomly setting the output of some gates in the LSTM block to zero during the forward pass. It should be noted that dropout is only applied during the training phase of the networks. When the autoencoder network is fully trained, all neurons are evaluated as they are supposed to be.

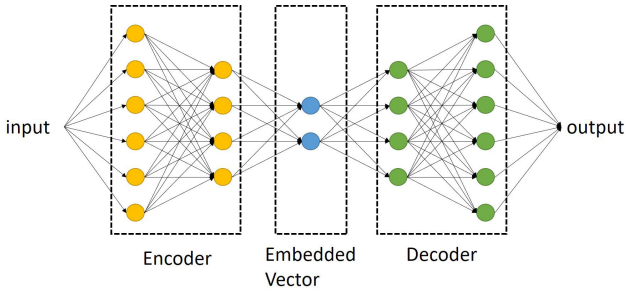

FIGURE 4. The general structure of an autoencoder.

C. DIMENSION COMPRESSION THROUGH AUTOENCODER

Recall that we attempt to use K previous states $\{x_{k-1}, x_{k-2}, \ldots, x_{k-K}\}$ to provide a surrogate of $x_k \in \mathbb{R}^n$. Several factors determine the size of an LSTM network. The length of the input sequence K determines how many steps we look backward and equals to the number of LSTM blocks within a layer. This is a hyper-parameter that can be tuned for better performance. More importantly, the size of the state variable p determines the size of the weight matrices and biases. The size of state variables p is determined by the distribution system and is a fixed number in a given system. In a distribution system with many buses, the p is also large, which can lead to poor computational efficiency as it demands an LSTM network with increased capacity.

To overcome the challenge of large p and oversized LSTM networks, we advocate the autoencoder to compress the dimension of system states x_k and used the encoded state vector x_k' as the input of the LSTM networks.

An autoencoder is a type of fully-connected neural network used to learn data encoding in an unsupervised manner. Figure 4 displays the general structure of an autoencoder, including an encoder, a decoder, and the embedded vector. The input and output of an autoencoder are both a set of M state variables $Z = \{x_{k,1}, x_{k,2}, \cdots x_{k,M}\}$. The encoded state vector, or the embedded vector, is acquired by evaluating the encoder network. And the embedded vector can be reversed transformed back to the original state vector using the decoder network.

With the autoencoder, we attempt to minimize the following loss function.

$$\min_{\{W',B'\}} L' = \sum_{i=1}^{N} ||x_{k,i} - h(x_{k,i})||_2^2, \tag{16}$$

TABLE 1. Number of neurons in the autoencoder network.

Hidden layer	# neurons
1	256
2	128
3	64
4	q
5	64
6	128
7	256

where $h(\cdot)$ is the compression function and is approximated with the autoencoder; W', B' are the weights and biases of the autoencoder network. In this paper, a 7-layer autoencoder is implemented with the number of neurons in each layer displayed in Table 1. The encoder network corresponds to layer 1-3 and decoder 5-7. The number of neurons in those six layers are fixed. Layer 4 corresponds to the embedded vector layer, and the number of neurons q in that layer is treated as a hyper-parameter that is different for different test systems and tuned for optimal performance. Similar to the LSTM network, we rescaled the input and output of the autoencoder to have a minimum value of 0 and a maximum value of 1. We further apply dropouts to reduce overfitting of the autoencoder network.

D. THE LSTM BASED SURROGATE FRAMEWORK

The LSTM based state surrogate framework is presented in Figure 5 and further summarized as follows.

1) The input is fed into the encoder network and output the embedded state vector.
2) The LSTM network takes the embedded vector as input and performs a forward pass to generate LSTM output. The forward pass involves multiple LSTM layers and one fully-connected layer.
3) The LSTM output is fed into the decoder network and outputs the surrogates of the actual states.

As discussed earlier, we rescale the input to the autoencoder/LSTM network such that the minimum and maximum values are 0 and 1, respectively. And we further reversely scaled the output back to its original space. These procedures are not presented in the figure.

IV. STATE ESTIMATION

For state estimation, we place our focus on distribution systems with a lack of measurement redundancy, as most of the distribution systems today still face the problem of measurement scarcity. In this context, the forward/backward sweep method is implemented with incorporation of the state surrogate results [22], [31]. The method involves two routines that are based on the exact line model.

We start by setting voltages of all buses to the initial values, acquired from the surrogate models. During each iteration, the method first performs a forward propagation that updates the line-to-ground voltage of buses. It is followed by a backward propagation that updates the line currents. Finally, convergence is reached when the voltage mismatch between two iterations is smaller than a tolerance.

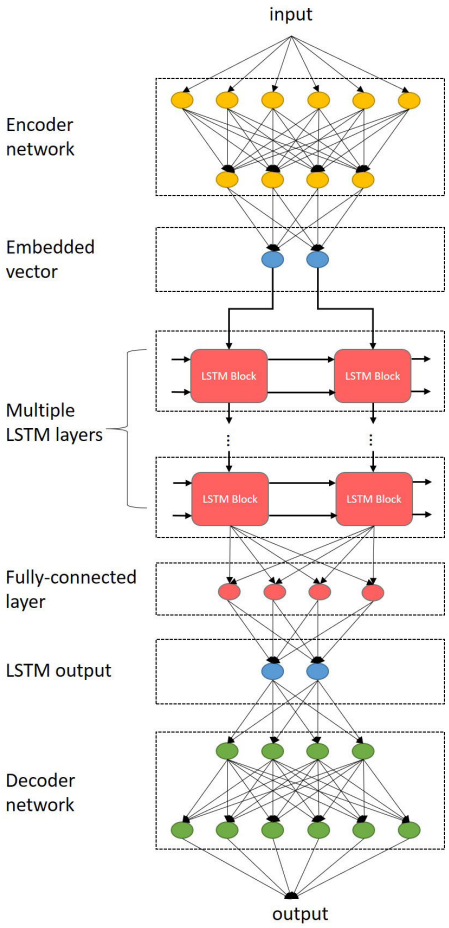

FIGURE 5. Architecture of the proposed LSTM-based surrogate model.

In the forward propagation, we set the voltage of the substation bus n to the measured value. The voltage of buses that are adjacent to the substation bus, denoted as $m \in \mathcal{N}_n$, can be updated with

$$V_n = (U + \frac{1}{2} Z_{mn} Y_{mn}) V_m + Z_{mn} I_m, \tag{17}$$

where V_n, V_m are the bus voltages, respectively. U is the identity matrix. $Z_{mn}, Y_{mn} \in \mathbb{C}^{3\times 3}$ are impedance matrix and shunt admittance matrix respectively, computed through the modified Carson's equation [32]. Once this procedure is finished, for each bus $m \in \mathcal{N}_n$, we update the voltages of its adjacent buses. This process continues until all bus voltages are updated.

During the backward propagation, we updates the branch currents in the reversed order of the forward propagation. We first compute the load currents of the buses. Then the branch current between bus n and one of its neighbors m is computed as

$$I_{mn} = \sum_{p \in \mathcal{N}_n, p \neq m} I_{pn} + I_n^L. \tag{18}$$

Loads are modeled as either constant power, constant current, or constant impedance load. The load currents of constant power and impedance load are computed as follows, whereas the constant current load does not need to be updated.

$$I_n^L = (\frac{S_n}{V_n})^*,$$
$$I_n^L = \frac{V_n}{Z_n}, \tag{19}$$

where S_n is the complex load of bus n.

Finally, convergence is reached when the voltage mismatch between two iterations is smaller than a prefixed tolerance. The algorithm of forward/backward sweep incorporating LSTM surrogates is tabulated in Algo. 1, where V_{curr} are the voltage vectors of current iteration, V_{prev} the previous iteration, and V_{surr} the surrogate results from LSTM.

Algorithm 1 - Forward/Backward Sweep

initialization $V_{curr} = V_{prev} = V_{surr}$; error $e = \infty$,
while $e >$ *tolerance* **do**
  $V_{curr} =$ forward()
  backward()
  $e = ||V_{curr} - V_{prev}||_\infty$
  $V_{prev} = V_{curr}$.

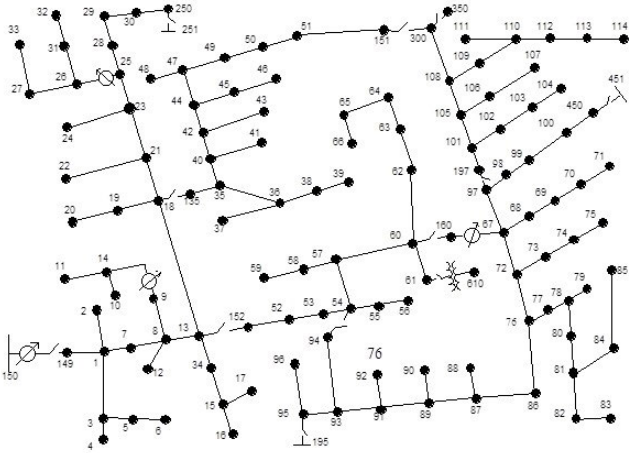

FIGURE 6. One-line diagram of the IEEE 123-bus system [37].

TABLE 2. Number of residential users and maximum load (kW) in selected buses.

Phase	a		b		c	
Bus	Users	Load	Users	Load	Users	Load
46	2	6.3				
47	3	9.5	4	12.9	3	11.8
48	7	21.1	7	24.2	7	21.0
49	3	9.2	7	22.3	5	19.7
50					4	12.7

V. SIMULATION STUDY

In this section, we present an experimental analysis of our proposed DSSE scheme. We first introduce the experimental design. Then we evaluate the performance of the proposed LSTM networks for state surrogates. It is followed by LSTM networks working as a surrogate for state estimation.

A. EXPERIMENTAL DESIGN

1) DATASETS

We collected datasets from multiple sources that includes 706 end-user residential load profiles [33], 101 commercial load profiles [34], 80 electric vehicle charging profiles [35] and 10 PV profiles [36]. We down-sampled commercial/EV/PV data to match the frequency of the residential data, which is one sample every half an hour. All data spans 2 years.

To train the LSTM networks, a full year of data (year 1) containing 17520 records is used. Meanwhile, the test set contains 3000 records that are randomly sampled from year 2. Hence the train and test data have no overlap.

The true states of the system are computed using the datasets described above. A 3% Gaussian noise is added after acquiring the ground truth to simulate noises of the physical measurements. The corrupted measurements are used for training LSTM models and working as surrogates for state estimation.

2) TEST SYSTEMS

The experiment is carried out on the IEEE 123-bus test feeders as well as the 8500-node test feeder. Figure 6 shows a one-line diagram of the 123-bus network. The three-phase system is unbalanced and realized with the following method. A number of users are first assigned to each bus. The bus-level load is combined from the individual load data. Table 2 shows the number of residential users and the maximum loads in kW of bus 46-50 in the 123-bus system. The blanks mean that there is no load for that phase of a bus. The table demonstrates that the test systems have unbalanced phases.

3) EXPERIMENTAL SETUPS

The experiment is implemented with Python and runs on a single machine with an Intel i7-7700 3.6GHz CPU and 64 GB RAM. The LSTM networks are implemented using MXNet [38] and are trained on an Nvidia 1080 GPU with 8 GB memory.

Some of the parameters are different for the two test systems. The depth of LSTM networks refers to the number of LSTM layers used in the model and is set to be 3 and 5 for the 123-bus and 8500-node systems, respectively. Moreover, the embedded vector width is the length of the encoded state variable after being processed by the encoder network of the autoencoder. This parameter is set to be 20 and 30 for the 123-bus and 8500-node systems, respectively.

Other settings are the same for both LSTM networks. The two decay parameters for Adam optimization algorithm are set to be $\beta = 0.99$ and $\beta_2 = 0.999$. The batch size is set to $n_{batch} = 64$. Both LSTM networks are set to train for epoch $e = 500$. We further applied early stopping with the patience set to 20 epochs.

TABLE 3. State surrogate RMSE on voltage magnitudes and angles.

	Method	123-bus	8500-node
Magnitude RMSE (p.u.)	LSTM	7.62×10^{-4}	3.54×10^{-4}
	GBT	7.96×10^{-3}	3.95×10^{-4}
	NN	1.33×10^{-3}	1.63×10^{-2}
Angle RMSE (degree)	LSTM	3.20×10^{-2}	1.56×10^{-2}
	GBT	4.87×10^{-2}	2.38×10^{-2}
	NN	7.36×10^{-2}	6.86×10^{-1}

To evaluate results, we examine the voltage magnitudes (per-unit) errors and angles (degree) errors. The root mean square error (RMSE) is used.

4) BASELINE METHODS

In this paper, we implement a shallow NN and a gradient boosting tree (GBT) model as the baselines. The shallow NN is based on a recent study [20], where the input of the networks is the available measurements of the system, such as metering data and substation-level voltage readings. For the shallow NN, the number of neurons in the hidden layers are 920 and 8080 for 123-bus and 8500-node test systems, respectively. The NN baseline shares the same parameters used in LSTM networks training. On the other hand, the GBT model is a common approach used for regression and is shown to outperform several traditional regression methods [39]. Because of the nature of GBT models, we build one model for each output channel. For instance, for a two-phase bus, four models are built as there are two voltage magnitude channels and two phase angle channels. The GBT model is implemented through LightGBM [40].

B. LSTM FOR STATE SURROGATE

The voltage magnitude and angle RMSEs for LSTM/GBT/NN are presented in Table 3. These results show the superior performance of our proposed LSTM networks against two other baseline methods. LSTM networks perform better than traditional NNs because, first, LSTM networks are capable of capturing temporal correlations of consecutive measurements. Second, the multi-layer LSTM networks with compressed input size have better scalability in terms of approximating state surrogate function $g(\cdot)$ compared to shallow networks. This better approximation ability is even more obvious when the system size is large. On the other hand, the LSTM networks slightly outperform GBT models.

We further plot the ground truth and the surrogate results of LSTM/GBT/NN in Figure 7 for bus 11 phase *a* of the 123-bus system, and Figure 8 for bus 459 phase *b* of the 8500-node test case. In each figure, 20 test instances are randomly selected. The visualizations agree with the results reported previously, showing that our proposed LSTM networks deliver the best performance. Moreover, the performance of shallow NN is significantly worse than the other two methods in the 8500-node system.

Furthermore, we present an analysis of the computation time of training surrogate models and generating surrogates.

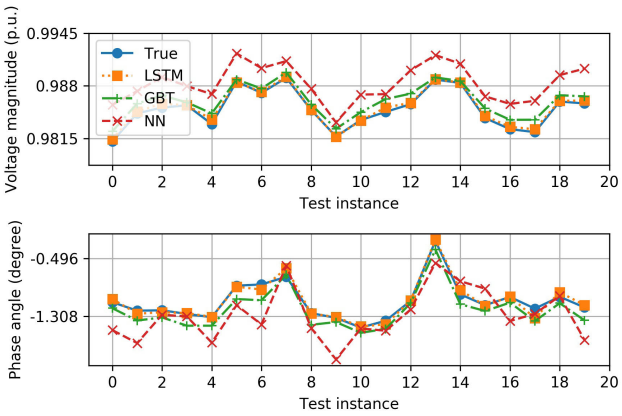

FIGURE 7. Surrogate results and the true values of voltage magnitudes and angles of bus 11 phase *a* of the 123-bus system.

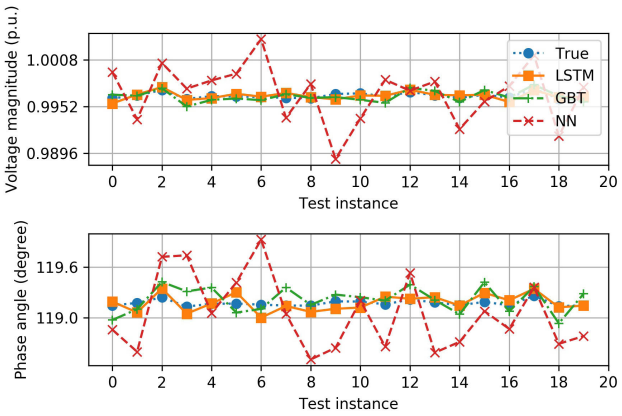

FIGURE 8. Surrogate results and the true values of voltage magnitudes and angles of bus 459 phase *b* of the 8500-node system.

TABLE 4. Training and surrogate time of each method.

Metric	Method	123-bus	8500-node
Training	LSTM	**9.4 min**	**11.1 min**
	GBT	27.0 min	80.0 hr
	NN	13.5 min	20.1 min
Surrogate	LSTM	**0.39 ms**	**0.40 ms**
	GBT	84.1 ms	2140 ms
	NN	0.52 ms	1.32 ms

The training time gives an idea of how much resources we need to expend. The training time might look trivial when the model is trained only once. However, in practice, models are often retrained to gain the best performance, and the training cost can become significant if the model is retrained, say, daily [41]. On the other hand, the time of generating surrogates is crucial because the surrogate values are fed to the subsequent state estimation procedure. Hence a small time cost is preferred. In Table 4, a comparison of the time cost is presented, with the best results made bold.

From the table, we see that the training/surrogate time of the proposed LSTM method is the least among the three methods for both of the test systems. The training/surrogate

TABLE 5. Results of the LSTM networks with/without autoencoder.

Metric	with	without
Voltage RMSE	3.54×10^{-3}	5.21×10^{-3}
Angle RMSE	1.56×10^{-2}	2.04×10^{-1}
Training time	11.1 min	27.4 min
Surrogate time	0.40 ms	1.67 ms

TABLE 6. State estimation time cost with/without a surrogate model.

	123-bus	8500-node
No Surrogate	35.6 ms	646 ms
With LSTM	9.4 ms	107 ms
With GBT	95.2 ms	2273 ms
With NN	13.1 ms	380 ms

time hardly increases as the size of the system increases. Because we applied autoencoder to reduce the dimension of the inputs of the LSTM network, the size of the networks is significantly reduced. Therefore the time efficiency is preserved when the system size increases. On the other hand, we see that the time cost of GBT models increases significantly as the size of the system increases. This is because, for each output channel, one single GBT model needs to be trained. For the 123-bus system, that means 494 models and 8500-node system 15108 models. The results show that our proposed LSTM surrogate modeling can be applied to systems at any scale, but demonstrates a clear benefit when applied to large-scale distribution systems.

Finally, we compare the results with/without autoencoder using the 8500-node test case. Table 5 shows the RMSEs and the training/surrogate time of the LSTM models. To acquire the results, we had to reduce the batch size to 8 to avoid GPU memory overflow. The RMSE results without autoencoder are worse than with autoencoder. This is because the LSTM networks used with autoencoder are relatively small and do not have enough capacity to learn when the size of the input is significant. One option to improve the results without autoencoder is to increase the size of the LSTM networks by adding more LSTM layers and the size of the hidden variable h. However, this is not viable due to hardware limitations. The results demonstrate that dimension compression with autoencoder not only improves the computational efficiency and performance of the LSTM networks but also decreases the size of the LSTM network, allowing the proposed method to be applied to large-scale distribution systems.

C. LSTM SURROGATES FOR STATE ESTIMATION

In this section, we examine the effect of applying LSTM/GBT/NN networks as surrogates for state estimation. Table 6 compares the time cost of solving state estimation with/without surrogates. The results with surrogates are the sum of generating surrogates and solving state estimation. On the other hand, the results without surrogates are acquired by setting the initial system states to unit vectors and run the forward/backward sweep method. From the results, it can be concluded that the time cost of state estimation with LSTM surrogates is significantly less than without a surrogate. NN surrogates have a similar effect, but is outperformed by LSTM networks as the quality of the surrogates from LSTM is superior to that of NN. Moreover, GBT is outperformed by state estimation without a surrogate because it takes too much time for GBT to produce the surrogate even though the quality of its surrogate is similar to LSTM networks.

Finally, the results show that the proposed surrogate-based state estimation with LSTM networks is applicable to large-scale distribution systems such as the 8500-node test system used in the paper. This is because time cost on the part of generating surrogates barely increases and, therefore, can be treated as a fixed cost. On the other hand, the time cost on solving the state estimation increase significantly as the size of the system increases. Therefore, the larger the system, the more time saving can be achieved by using LSTM networks as surrogates.

VI. CONCLUSION

In this paper, we have proposed a scalable distribution systems state estimation scheme using surrogate modeling. LSTM networks are built to take previous states as inputs and output the surrogate, a rough estimate of the current states. We have further compressed the input of the LSTM networks with autoencoders. The surrogates are fed into the forward/backward state estimator to provide an estimate of the state. The proposed framework is tested on IEEE 123-bus and 8500-node test feeders. Experimental results show that LSTM networks with the autoencoder compression can generate surrogates of system states with high accuracy and significantly reduce the computation time of state estimation procedures. The proposed method has sought an appealing path of combining the thrusts of traditional methods with data-driven methods, and it sheds light on future large-scale distribution system state estimation.

Distribution systems often experience frequent topology changes, and this was not considered in this paper. One potential solution is to train multiple LSTM networks, with each network being responsible for one specific topology. However, a more interesting question is whether we can treat the topology status as categorical inputs of the model and use one LSTM network to account for all topologies of a distribution system. This research topic can be further explored based on the current study.

REFERENCES

[1] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC Press, 2004.
[2] W. H. Kersting, *Distribution System Modeling and Analysis*. Boca Raton, FL, USA: CRC Press, 2006.
[3] C. Trevino, "Cases of difficult convergence in load-flow problems," in *Proc. IEEE Summer Power Meeting*, 1970.
[4] C. Hansen and A. Debs, "Power system state estimation using three-phase models," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 818–824, May 1995.
[5] M. Baran and A. Kelley, "A branch-current-based state estimation method for distribution systems," *IEEE Trans. Power Syst.*, vol. 10, no. 1, pp. 483–491, Feb. 1995.

[6] R. Singh, R. Jabr, and B. Pal, "Choice of estimator for distribution system state estimation," *IET Gener., Transmiss. Distrib.*, vol. 3, no. 7, pp. 666–678, Jul. 2009.
[7] A. Al-Wakeel, J. Wu, and N. Jenkins, "State estimation of medium voltage distribution networks using smart meter measurements," *Appl. Energy*, vol. 184, pp. 207–218, Dec. 2016.
[8] M. Majidi, M. Etezadi-Amoli, H. Livani, and M. Fadali, "Distribution systems state estimation using sparsified voltage profile," *Electric Power Syst. Res.*, vol. 136, pp. 69–78, Jul. 2016.
[9] K. Zhu, L. Nordstrom, and L. Ekstam, "Application and analysis of optimum PMU placement methods with application to state estimation accuracy," in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2009, pp. 1–7.
[10] J. Liu, F. Ponci, A. Monti, C. Muscas, P. A. Pegoraro, and S. Sulis, "Optimal meter placement for robust measurement systems in active distribution grids," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1096–1105, May 2014.
[11] Z. Wu, X. Du, W. Gu, Y. Liu, P. Ling, J. Liu, and C. Fang, "Optimal PMU placement considering load loss and relaying in distribution networks," *IEEE Access*, vol. 6, pp. 33645–33653, 2018.
[12] R. Khorshidi, F. Shabaninia, and T. Niknam, "A new smart approach for state estimation of distribution grids considering renewable energy sources," *Energy*, vol. 94, pp. 29–37, Jan. 2016.
[13] S. Nanchian, A. Majumdar, and B. C. Pal, "Three-phase state estimation using hybrid particle swarm optimization," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1035–1045, May 2017.
[14] G. Valverde and V. Terzija, "Unscented Kalman filter for power system dynamic state estimation," *IET Gener. Transm. Distrib.*, vol. 5, no. 1, p. 29, 2011.
[15] J. Zhao, M. Netto, and L. Mili, "A robust iterated extended Kalman filter for power system dynamic state estimation," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3205–3216, Jul. 2017.
[16] M. Hassanzadeh, C. Y. Evrenosoglu, and L. Mili, "A short-term nodal voltage phasor forecasting method using temporal and spatial correlation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3881–3890, Sep. 2016.
[17] J. Zhao, G. Zhang, Z. Y. Dong, and M. La Scala, "Robust forecasting aided power system state estimation considering state correlations," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2658–2666, Jul. 2018.
[18] M. Brown Do Coutto Filho and J. De Souza, "Forecasting-aided state estimation—Part I: Panorama," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov. 2009.
[19] P. N. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, "State estimation in distribution smart grids using autoencoders," in *Proc. IEEE 8th Int. Power Eng. Optim. Conf.*, Mar. 2014, pp. 358–363.
[20] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4796–4805, Nov. 2019.
[21] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review," *Int. J. Autom. Comput.*, vol. 14, no. 5, pp. 503–519, Oct. 2017.
[22] D. Thukaram, J. Jerome, and C. Surapong, "A robust three-phase state estimation algorithm for distribution networks," *Electric Power Syst. Res.*, vol. 55, no. 3, pp. 191–200, Sep. 2000.
[23] R. Singh, B. Pal, and R. Jabr, "Distribution system state estimation through Gaussian mixture model of the load as pseudo-measurement," *IET Gener. Transm. Distrib.*, vol. 4, no. 1, p. 50, 2010.
[24] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1888–1896, Nov. 2012.
[25] E. Iuliano and E. A. Pérez, Eds., *Application of Surrogate-Based Global Optimization to Aerodynamic Design*. Springer, 2016, pp. 25–46.
[26] K.-R. Shih and S.-J. Huang, "Application of a robust algorithm for dynamic state estimation of a power system," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 141–147, Feb. 2002.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, Edinburgh, Scotland, vol. 2. London, U.K.: IEE, 1999, pp. 850–855.
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[30] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*. [Online]. Available: https://arxiv.org/abs/1207.0580
[31] G. Chang, S. Chu, and H. Wang, "A simplified forward and backward sweep approach for distribution system load flow analysis," in *Proc. Int. Conf. Power Syst. Technol.*, Oct. 2006, pp. 1–5.
[32] J. R. Carson, "Wave propagation in overhead wires with ground return," *Bell Syst. Tech. J.*, vol. 5, no. 4, pp. 539–554, Oct. 1926.
[33] *Sgsc: Smart Grid, Smart City*. Accessed: Sep. 25, 2018. [Online]. Available: https://renewablestocktake.com.au/directory/project-680
[34] *Real Time Building Utility Use Data*. Accessed: Apr. 25, 2019. [Online]. Available: http://portal.emcs.cornell.edu/
[35] Y. Wang, B. Wang, C.-C. Chu, H. Pota, and R. Gadh, "Energy management for a commercial building microgrid with stationary and mobile battery storage," *Energy Buildings*, vol. 116, pp. 141–150, Mar. 2016.
[36] T. Zhang, C.-C. Chu, and R. Gadh, "A two-tier energy management system for smart electric vehicle charging in UCLA: A solar-to-vehicle (S2V) case study," in *Proc. IEEE Innov. Smart Grid Technol.-Asia (ISGT-Asia)*, Nov. 2016, pp. 288–293.
[37] K. P. Schneider, B. A. Mather, B. C. Pal, C.-W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. De Araujo, R. C. Dugan, S. Matthias, S. Paudyal, T. E. Mcdermott, and W. Kersting, "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, May 2018.
[38] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," 2015, *arXiv:1512.01274*. [Online]. Available: https://arxiv.org/abs/1512.01274
[39] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[40] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.
[41] M. Grbovic and H. Cheng, "Real-time personalization using embeddings for search ranking at airbnb," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining-KDD*, 2018, pp. 311–320.

ZHIYUAN CAO received the B.S. degree in mechanical engineering from the University of Electronic Science and Technology of China, in 2014. He is currently pursuing the Ph.D. degree with the University of California at Los Angeles, Los Angeles. His research interests include distribution system state estimation and machine learning applications in power systems.

YUBO WANG received the B.S. degree in electrical engineering from Southeast University, China, in 2011, and the M.S. degree in electrical engineering and the Ph.D. degree in mechanical engineering from the University of California at Los Angeles, USA, in 2012 and 2017, respectively. He is currently a Research Scientist with Siemens Corporate Technology. His research interests include optimizations and data analytics with applications in power systems and the Internet of Things.

CHI-CHENG CHU received the B.S. degree from National Taiwan University, in 1990, and the Ph.D. degree from the University of Wisconsin–Madison, in 2001. He is currently the Project Lead of the Henry Samueli School of Engineering and Applied Science with the University of California at Los Angeles (UCLA), Los Angeles, where he is also the Forum Convener with the WINMEC (ESmart Consortium). He has over 15 years of experience in the research and development of software architectures, frameworks, and solutions. He has delivered multiple project solutions and software packages to the industry globally. He is the senior research manager who supervised and steered multiple industry and academia research projects in the fields of smart grids, electric vehicles and renewable energy integration, micro grid systems, sensor networks, RFID technologies, mobile communication, media entertainment, 3D/2D visualization of scientific data, and computer-aided design.

RAJIT GADH received the bachelor's degree from IIT Kanpur, the master's degree from Cornell University, and the Ph.D. degree from Carnegie Mellon University (CMU), all in engineering.

He was a Visiting Researcher with Stanford University. He is currently a Professor of the Henry Samueli School of Engineering and Applied Science with the University of California at Los Angeles (UCLA), the Founder and the Director the Smart Grid Energy Research Center or SMERC, and the Founder and the Director of the WINMEC (ESmart Consortium), UCLA. He has taught as a Visiting Researcher with UC Berkeley. He has been an Assistant, Associate, and a Full Professor with the University of Wisconsin–Madison. His current research interests include modeling and control of smart grids, electric vehicle to grid integration, vehicle to grid (V2G), autonomous electric vehicles, demand response, microgrids, energy storage in the grid, renewable integration, the Internet of Things, and wireless/RFID.