Flexible Machine Learning Based Cyberattack Detection Using Spatiotemporal Patterns for Distribution Systems

Mingjian Cui, *Senior Member, IEEE*, Jianhui Wang, *Senior Member, IEEE*, and Bo Chen, *Member, IEEE*

***Abstract*—This letter develops a flexible machine learning detection method for cyberattacks in distribution systems considering spatiotemporal patterns. Spatiotemporal patterns are recognized by the graph Laplacian based on system-wide measurements. A flexible Bayes classifier (BC) is used to train spatiotemporal patterns which could be violated when cyberattacks occur. Cyberattacks are detected by using flexible BCs online. The effectiveness of the developed method is demonstrated through standard IEEE 13- and 123-node test feeders.**

***Index Terms*—Cyberattack detection, distribution systems, graph Laplacian, machine learning, spatiotemporal patterns.**

I. Introduction

THE growing deployment of distributed energy resources (DERs), microgrids, and other distribution-level technology and assets has completely changed the way the distribution systems have been designed and operated traditionally. Also, as an increasing number of sensors (e.g., micro-PMUs) are being developed and deployed on the distribution system in conjunction with the conventional SCADA systems, advanced metering infrastructure (AMI), and other field devices to enable data-driven observability and grid-edge data analytics [1], the attack surface to the distribution management system (DMS) is inevitably enlarged. DMS and associated monitoring and control systems are among the key actors for making decisions and exchanging information.

However, existing cybersecurity technologies employed in distribution systems are still vulnerable to cyberattacks. It is highly necessary to develop cyber-resilient DMS functions and cybersecurity technologies to enable future energy delivery systems to accurately detect, dynamically adapt, successfully survive and reject a cyberattack. Unlike conventional cyberattack detection techniques, such as naive Bayes classifiers (BCs) which are highly based on the normality assumption, this letter attempts to capture the continuous attribute of spatiotemporal patterns among system measurements by developing flexible BCs. Essentially, spatiotemporal patterns of measurement data under normal conditions would be compromised when cyberattacks occur. Based on this concept, this letter seeks to address two critical questions for the cyberattack detection on distribution systems. (i) Is it possible to quantitatively capture the spatiotemporal patterns between cyberattack scenarios and normal scenarios? (ii) Can operators deploy flexible BCs to enhance the accuracy of conventional cyberattack detection methods?

In this letter, we seek to integrate the spatiotemporal patterns of system measurements into a flexible BC for cyberattack detection. Specifically, spatiotemporal patterns are captured by the generalized graph Laplacian (GGL) matrix for system measurements. For the training process of the proposed flexible BC, they are taken as its input variables, while the labels of cyberattack templates are taken as its output variables. For the testing process, the online spatiotemporal patterns captured by GGL are put into the proposed flexible BC, which subsequently outputs the cyberattack detection results.

The organization of this letter is as follows. In Section II, the flexible BC detection method is briefly introduced based on spatiotemporal patterns using the graph Laplacian. Case studies and result analysis performed on the public distribution load data are discussed in Section III. Concluding remarks are summarized in Section V.

II. Flexible Machine Learning Method

The overall procedure of the developed cyberattack detection method is shown in Fig. 1 and summarized as follows. *Step i)*: an unsupervised machine learning method, namely GGL, is used to characterize spatiotemporal patterns of system measurements; *Step ii)*: a supervised machine learning method (i.e., flexible BC) is used to train the spatiotemporal patterns characterized by the GGL matrix; and *Step iii)*: two sets of metrics, namely the true positive rate (TPR) and contingency table, are used to evaluate the performance of different detection methods. Detailed information on each step is described in the following.

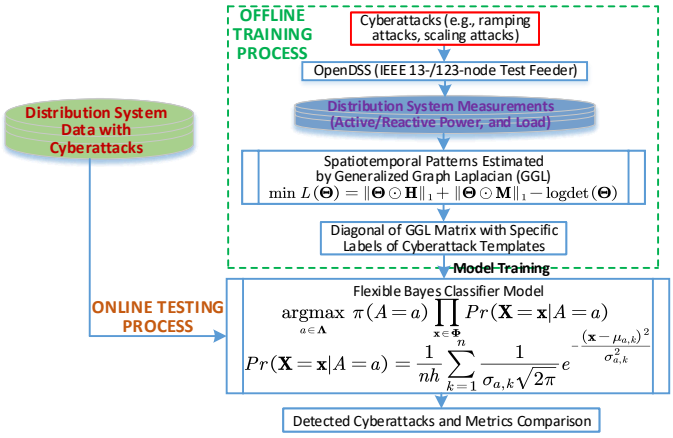


Fig. 1. Flowchart of the developed cyberattack detection method.

A. Spatiotemporal Patterns Using Graph Laplacian

As an unsupervised machine learning method, graph learning techniques can quantitatively represent the spatiotemporal

M. Cui and J. Wang are with the Department of Electrical and Computer Engineering at Southern Methodist University, Dallas, TX 75275, USA (email: {mingjiancui, jianhui}@smu.edu).

B. Chen is with the Energy Systems Division, Argonne National Laboratory, Argonne, IL 60439, USA (email: bo.chen@anl.gov).

Manuscript received, 2019.

patterns [2]. Among them, the GGL can maintain all the edges with positive weights and practically introduce additional connectivity due to negative weights [3]. To estimate the GGL matrix, the Lagrangian optimization problem can be constituted as:

$$\min\ L(\mathbf{\Theta}) = \|\mathbf{\Theta} \odot \mathbf{H}\|_1 + \|\mathbf{\Theta} \odot \mathbf{M}\|_1 - \text{logdet}\,(\mathbf{\Theta}) \quad (1)$$

$$\mathcal{L}\,(\mathbf{A}) = \left\{ \mathbf{\Theta} \in \mathcal{L} \;\middle|\; \begin{array}{l} (\mathbf{\Theta})_{ij} \leq 0 \ \text{ if } \ (\mathbf{A})_{ij} = 1 \\ (\mathbf{\Theta})_{ij} = 0 \ \text{ if } \ (\mathbf{A})_{ij} = 0 \end{array} \right\}_{\forall i,j\ i \neq j} \quad (2)$$

where $\mathbf{H}$ is the regularization matrix and $\mathbf{H} = \alpha\,(\mathbf{I} - \mathbf{II})$. $\mathbf{I}$ is an identity matrix. $\mathbf{II}$ is an all-ones matrix. α is the regularization parameter. $\mathbf{\Theta}$ is the estimated GGL matrix. $\mathcal{L}$ is the target set of graph Laplacians. $\mathbf{A}$ is the similarity matrix. $\odot$ means the element-wise multiplication of two matrices. $\|\cdot\|_1$ means the sum of absolute values of all elements (ℓ_1-norm). $\text{logdet}\,(\cdot)$ means the natural logarithm of a determinant. $\mathbf{M}$ is the Lagrange multiplier matrix.

B. Flexible BC for Cyberattack Detection

By using spatiotemporal patterns as inputs, conventional naive BCs are usually handled by discretization and assume that they follow a Gaussian distribution. However, this assumption based on numerical attributes cannot hold for all of the domains (or classes). Compared with naive BCs, the developed flexible BC is based on the nonparametric kernel estimation which does not require any normality assumption and outperforms in most domains. Also, the flexible BC can store every continuous attribute value it sees during the training process.

Let $f\,(x)$ be defined as an ideal probability density function of one spatiotemporal pattern x of measurements assumed to be tampered with cyberattacks, and let $\widehat{f}(x)$ be an approximate estimate of $f\,(x)$ based on n samples of pattern x. We assume that a kernel density estimation function $\widehat{f}_n\,(x)$ can be perfectly used to fit the ideal function $f\,(x)$ of one spatiotemporal pattern x. That is to say, $\widehat{f}_n\,(x)$ is strongly pointwise consistent if $\widehat{f}_n\,(x) \rightarrow f\,(x)$ is guaranteed for all samples of the spatiotemporal pattern x. This assumption can be mathematically expressed by:

$$Pr\left(\lim_{n\to\infty} \left|\widehat{f}_n\,(x) - f\,(x)\right| < \epsilon\right) = 1, \quad \forall \epsilon:\ \epsilon > 0 \quad (3)$$

where ϵ is the fitting error and can be set as any positive value that is sufficiently small.

Let A be the variable denoting the template of a cyberattack instance, and let $\mathbf{X}$ be a vector variable denoting the observed spatiotemporal patterns. Also, let a represent a particular cyberattack template, and let $\mathbf{x}$ represent a particular observed spatiotemporal pattern vector. Given a particular spatiotemporal pattern ($\mathbf{X} = \mathbf{x}$), let the actual conditional distribution of the cyberattack template ($A = a$) be $Pr\,(A = a|\mathbf{X} = \mathbf{x})$. Then the flexible Bayes estimation $\widehat{Pr}\,(A = a|\mathbf{X} = \mathbf{x})$ is a strongly consistent estimator of $Pr\,(A = a|\mathbf{X} = \mathbf{x})$. The Bayes rule can simply be used to compute the probability of each cyberattack template given the vector of observed values for spatiotemporal patterns. Thus, the flexible BC's objective is given by:

$$\begin{aligned} \arg\max_{a\in\mathbf{\Lambda}} Pr &\left(A = a | \underbrace{\mathit{\Phi}_1^{\mathbb{S}}, \cdots, \mathit{\Phi}_i^{\mathbb{S}}, \cdots \mathit{\Phi}_{N_\text{S}}^{\mathbb{S}}}_{Spatial\ Patterns}, \underbrace{\mathit{\Phi}_1^{\mathbb{T}}, \cdots, \mathit{\Phi}_j^{\mathbb{T}}, \cdots \mathit{\Phi}_{N_\text{T}}^{\mathbb{T}}}_{Temporal\ Patterns} \right) \\ &= \frac{\pi\,(A=a) \prod_{\mathbf{x}\in\mathbf{\Phi}} Pr\,(\mathbf{X}=\mathbf{x}|A=a)}{\sum_{a\in\mathbf{\Lambda}} \pi\,(A=a) \prod_{\mathbf{x}\in\mathbf{\Phi}} Pr\,(\mathbf{X}=\mathbf{x}|A=a)} \quad (4) \\ \Longrightarrow &\arg\max_{a\in\mathbf{\Lambda}}\ \pi\,(A=a) \prod_{\mathbf{x}\in\mathbf{\Phi}} Pr\,(\mathbf{X}=\mathbf{x}|A=a) \quad (5) \end{aligned}$$

where $\mathit{\Phi}_i^{\mathbb{S}}$ and $\mathit{\Phi}_i^{\mathbb{T}}$ represent the spatial and temporal patterns estimated by GGL, respectively. N_S is the number of measurements in the spatial domain. N_T is the number of time windows. $\pi\,(A = a)$ is the prior probability of the attack template a. $Pr(\cdot)$ is the conditional probability function. $\mathbf{\Lambda}$ represents the set of four cyberattack templates, i.e., scaling, ramping, random, and smooth-curve attacks. The flexible BC's constraints are given by:

$$\begin{aligned} Pr\,(\mathbf{X}=\mathbf{x}|A=a) &= \frac{1}{nh}\sum_{k=1}^{n} G\,(\mathbf{x};\mu_{a,k},\sigma_{a,k}) \\ &= \frac{1}{nh}\sum_{k=1}^{n} \frac{1}{\sigma_{a,k}\sqrt{2\pi}} e^{-\frac{\left(\mathbf{x}-\mu_{a,k}\right)^2}{\sigma_{a,k}^2}} \quad (6a) \end{aligned}$$

$$\mathbf{\Lambda} = \{Scaling, Ramping, Random, Smooth\} \quad (6b)$$

$$\begin{aligned} \mathbf{\Phi} &= \left\{\text{diag}\left(\mathbf{\Theta}^{\mathbb{S}}\right),\ \text{diag}\left(\mathbf{\Theta}^{\mathbb{T}}\right)\right\} \\ &= \left\{\mathit{\Phi}_1^{\mathbb{S}}, \cdots, \mathit{\Phi}_i^{\mathbb{S}}, \cdots \mathit{\Phi}_{N_\text{S}}^{\mathbb{S}}, \mathit{\Phi}_1^{\mathbb{T}}, \cdots, \mathit{\Phi}_j^{\mathbb{T}}, \cdots \mathit{\Phi}_{N_\text{T}}^{\mathbb{T}}\right\} \quad (6c) \end{aligned}$$

$$\mathit{\Phi}_i^{\mathbb{S}} \in \text{diag}\left(\mathbf{\Theta}^{\mathbb{S}}\right),\ \mathit{\Phi}_j^{\mathbb{T}} \in \text{diag}\left(\mathbf{\Theta}^{\mathbb{T}}\right),\ \mathbf{x} \in \mathbf{\Phi} \quad (6d)$$

where k ranges over the training points of attribute $\mathbf{X}$ in cyberattack A. $G\,(\cdot)$ is the Gaussian kernel function. Eq. (6a) shows the continuous attribute estimated by kernel smoothing density functions. Eq. (6b) denotes the set of four cyberattack templates that are inspired by [4]. Eq. (6c) and (6d) show the set of spatiotemporal patterns obtained by the diagonal of GGL matrix $\mathbf{\Theta}$. h is the bandwidth that can be selected by the mean integrated squared error (MISE) function, given by:

$$MISE\,(h) = \mathbf{E}\left[\int \left(\widehat{Pr}_h\,(\mathbf{X}|A) - Pr\,(\mathbf{X}|A)\right)^2 \mathrm{d}\mathbf{x}\right] \quad (7)$$

C. Evaluation Metrics

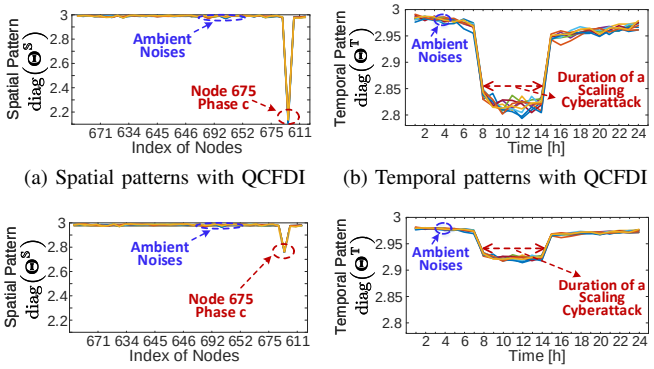
1) Metric I: TPR is defined as the percentage of the number (TP) of detected cyberattacks that are actually observed in the real system measurements over the total number (N_A) of cyberattacks, given by:

$$TPR = TP/N_\text{A} \times 100\% \quad (8)$$

2) Metric II: Based on the contingency table, a suite of indicators can be derived for the performance evaluation of cyberattack detection, including the probability of detection (POD), critical success index (CSI), frequency bias score ($FBIAS$), and success ratio (SR). Detailed information of these indicators can be found in [5].

III. Case Studies and Results

The raw load data is obtained from the Pecan Street Dataport [6]. 80% of the measurement data in a whole year (292 days) is used for training with 28,032 samples, while 20% of those (73 days) is used for testing with 7,008 samples. Four cyberattack templates are simulated on each node connected



(a) Spatial patterns with QCFDI (b) Temporal patterns with QCFDI

(c) Spatial patterns with SCFDI (d) Temporal patterns with SCFDI

Fig. 2. Robustness analysis of spatiotemporal patterns using GGL against ambient noises.

with load, including scaling, ramping, random, and smooth-curve attacks. Detailed information of cyberattack templates is described in Appendix A. Two distribution systems with 13 and 123 buses are simulated using OpenDSS [7]. Active and reactive power data in distribution systems is assumed to be vulnerable under cyberattack scenarios.

A. Robustness of Spatiotemporal Patterns Representation

To verify the robustness of spatiotemporal patterns recognized by GGL, Fig. 2 shows the results with different ambient noises on system measurements. Figs. 2a and 2b are with quickly changing false data injection (QCFDI) attacks. Fig. 2a shows an example of spatial patterns with cyberattacks preset on Node 675 Phase c. Fig. 2b presents an example of temporal patterns with cyberattacks preset from 8 hour to 14 hour. As can be seen, the GGL can accurately capture both spatial and temporal patterns when QCFDI attacks occur. Also, the results are robust for different ambient noises. Figs. 2c and 2d are with slowly changing false data injection (SCFDI) attacks. The slow change amplitude of SCFDI attacks is set as 10% of that of QCFDI attacks. Similarly, the GGL can accurately capture both spatial and temporal patterns when SCFDI attacks occur. Also, the results are robust for different ambient noises.

B. Effectiveness Analysis of Flexible BCs

Fig. 3 compares Support Vector Machine (SVM) [8], naive BCs, and flexible BCs using the visualized performance diagram on the IEEE 13-node test feeder. Four representative nodes are deployed for comparison. Fig. 4 shows the visualized performance diagram on the IEEE 123-node test feeder. Eight representative nodes are deployed for comparison. As can be seen, for different system nodes, the flexible BC method (rectangles closest to the top right corner) performs better than SVM and the naive BC method (circles and triangles). In addition, it can be observed that smooth-curve attacks are relatively challenging to detect for all methods. This is because they are very secretive while presenting a smooth trending together with neighboring measurements at both the beginning and the end of one attack. Fig. 5 compares SVM, naive BC, and flexible BC methods with respect to different scaling attack parameters (λ_S=0.2, 0.5, and 1.0). The TPR metric is used for validation. As shown in this figure, for all parameters, flexible BCs perform better than both SVM and naive BCs with higher TPR values. Also, TPR values are increased with scaling attack parameters (from 0.2 to 1.0).

TABLE I
COMPARISON OF DIFFERENT DETECTION METHODS FOR CYBERATTACKS

Methods	Flexible BC	Naive BC	SVM	Decision Tree
TPR [%]	98.75	95.46	96.38	95.26

To quantitatively evaluate the performance of the developed method, Table I compares different detection methods for cyberattacks in the IEEE 123-node test feeder. As can be seen, the flexible BC shows the largest TPR metric compared with the naive BC, SVM, and decision tree methods. This is because the flexible BC does not require any normality assumption and can store every continuous attribute value it sees during the training process.

IV. Discussion and Analysis

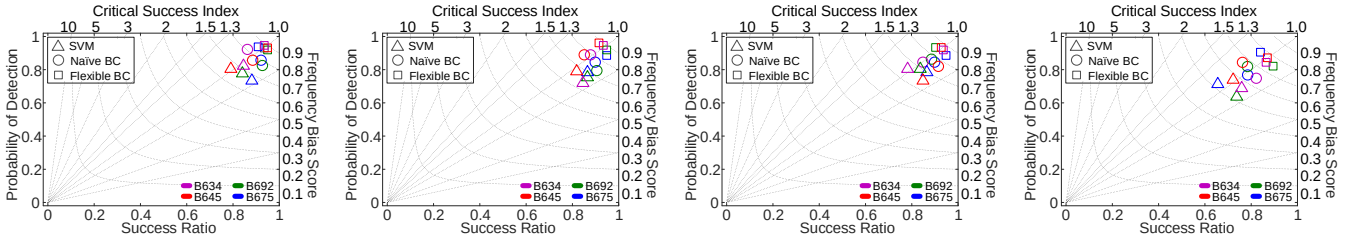
Spatiotemporal patterns of measurements have been widely used in the areas of renewable forecasting and plug-in electric vehicles (PEVs) in recent years. Inspired by this background, deploying spatiotemporal patterns for cyberattack detection has a broad prospect by coordinating with machine learning techniques. Complex distribution networks can be defined as a graphical model where variables are associated with highly nonlinear target functions, and complex spatial and temporal relationships exist among such variables even for cyberattacks. Since distribution systems are running based on complicated physical laws and rules, describing the spatiotemporal patterns by machine learning paves a way for mapping such relationships that could be significantly compromised by the injected cyberattacks. For the future work of this letter, deep learning techniques will be further involved. That is to say, the spatiotemporal patterns will be mapped to a linear space by using the Long Short-term Memory (LSTM) network to improve the potential detection accuracy for cyberattacks.

V. Conclusion

In this letter, we develop a flexible machine learning based cyberattack detection method by using the generalized graph Laplacian (GGL) and flexible Bayes classifiers (BCs). Spatiotemporal patterns are quantitatively characterized by GGL, which could be compromised when cyberattacks occur. The flexible BCs are used for training spatiotemporal patterns of system measurements and detecting cyberattacks online. Numerical results of case studies verify the effectiveness of the developed cyberattack detection method based on machine learning techniques.

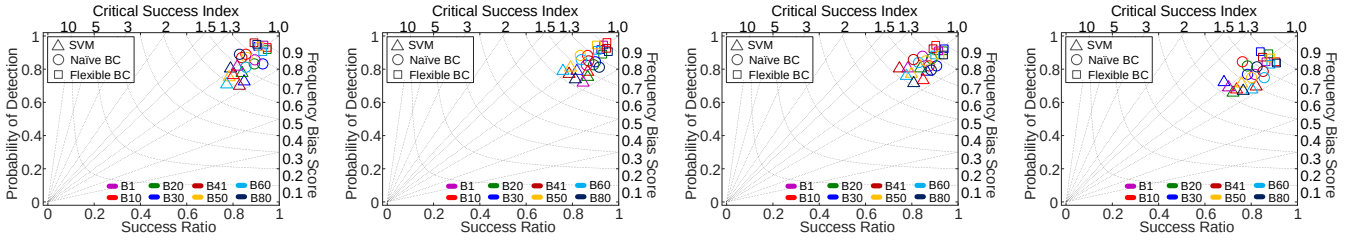
Appendix A
Templates of Cyberattacks

In this letter, we are not aiming to develop new adversary models of cyberattack templates. Inspired by existing adversary models for attacking automatic generation control (AGC) [9], we assume that attackers with advanced skills could migrate these adversary models to those on micro-PMU measurements in distribution systems. The cyberattack templates can be divided into four categories: scaling, ramping, random, and smooth-curve, which are briefly described as follows. Note that this letter does not aim to develop new templates for cyberattacks.



(a) Scaling attack (b) Ramping attack (c) Random attack (d) Smooth-curve attack

Fig. 3. Performance diagram of contingency table for different methods on the IEEE 13-node test feeder.



(a) Scaling attack (b) Ramping attack (c) Random attack (d) Smooth-curve attack

Fig. 4. Performance diagram of contingency table for different methods on the IEEE 123-node test feeder.

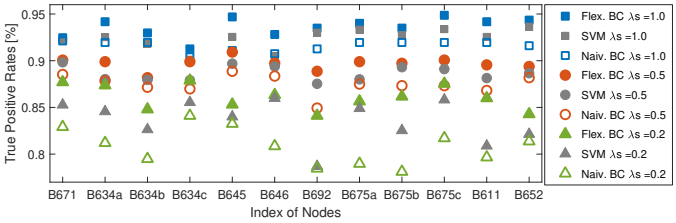


Fig. 5. Comparison of different scaling attack parameters (λ_S=0.2, 0.5, and 1.0) on the IEEE 13-node test feeder.

A. Scaling Attack

Scaling attacks involve modifying the values in a specified duration multiplied by a scaling attack parameter λ_S:

$$\widetilde{m}_t = (1 + \lambda_S) \times m_t, \quad \forall t : t_s < t < t_e \tag{9}$$

where t_s and t_e represent the start- and end-time of one cyberattack, respectively. m_t is the original measurement without any cyberattacks. $\widetilde{m}_t$ is the measurement tampered with cyberattacks.

B. Ramping Attack

Ramping attack considers both up- and down-ramping anomalies. This attack is more challenging to detect for operators. The values in the specified range are multiplied by a ramping coefficient λ_R.

$$\widetilde{m}_t = [1 + \lambda_R \times (t - t_s)] \times m_t, \ \forall t : t_s < t < \lfloor \frac{t_s + t_e}{2} \rfloor \tag{10}$$

$$\widetilde{m}_t = [1 + \lambda_R \times (t_e - t)] \times m_t, \ \forall t : \lfloor \frac{t_s + t_e}{2} \rfloor < t < t_e \tag{11}$$

where $\lfloor \cdot \rfloor$ indicates the floored value which is used to present the approximate intermediate point between t_s an t_e.

C. Random Attack

This attack involves the addition of values returned by a uniform random function to measurements.

$$\widetilde{m}_t = m_t + \lambda_{RA} \times \mathsf{rand}\,(t)\,, \quad \forall t : t_s < t < t_e \tag{12}$$

where $\mathsf{rand}\,(\cdot)$ is a uniformly distributed random number generator that can be achieved by a built-in function in MATLAB. λ_{RA} is a scale factor. The start- and end-time of one random attack is assumed to be randomly set by attackers.

D. Smooth-Curve Attack

Smooth-curve attacks are implemented by replacing the set of contiguous start and end points in the original measurements. In this letter, a polynomial fitting is used to generate a smooth curve and replace the original measurements with neighboring points.

REFERENCES

[1] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.

[2] M. Cui, J. Wang, A. R. Florita, and Y. Zhang, "Generalized graph Laplacian based anomaly detection for spatiotemporal microPMU data," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3960–3963, Sep. 2019.

[3] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[4] M. Cui, J. Wang, and M. Yue, "Machine learning based anomaly detection for load forecasting under cyberattacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5724–5734, Sep. 2019.

[5] M. Cui, J. Zhang, A. R. Florita, B.-M. Hodge, D. Ke, and Y. Sun, "An optimized swinging door algorithm for identifying wind ramping events," *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 150–162, Jan. 2016.

[6] Pecan Street Data. [Online]. Available: https://www.pecanstreet.org/category/dataport/

[7] R. C. Dugan, "Reference guide: The open distribution system simulator (OpenDSS)," *Electric Power Research Institute, Inc*, vol. 7, p. 29, 2012.

[8] C. Wang, Z. Wang, J. Wang, and D. Zhao, "SVM-based parameter identification for composite ZIP and electronic load modeling," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 182–193, Jan. 2019.

[9] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.