Deep Learning-Aided Cyber-Attack Detection in Power Transmission Systems

David Wilson, Yufei Tang
Florida Atlantic University
Boca Raton, FL 33431, USA
Email: {davidwilson2016, tangy}@fau.edu

Jun Yan
Concordia University
Montréal, QC H3G 1M8, Canada
Email: jun.yan@concordia.ca

Zhuo Lu
University of South Florida
Tampa, FL 33620, USA
Email: zhuolu@usf.edu

***Abstract*—The security of smart grid is a major challenge in grid modernization. While many existing solutions rely on human-defined features to develop machine learning (ML) based attack detectors against prominent exploits, such features are becoming more expensive and less effective in the smart grid. To supplement more high-quality features for ML-based threat monitoring, this paper proposed a stacked autoencoder (SAE) based deep learning framework to develop machine-learned features against transmission SCADA attacks. Compared with the state-of-the-art ML detectors, the proposed framework leverages the automaticity of unsupervised feature learning to reduce the reliance on system models and human expertise in complex security scenarios. Simulations with data collected from a high-fidelity smart grid testbed demonstrated that the machine-learned features effectively enabled more accurate discrimination against SCADA exploits in power transmission systems.**

I. Introduction

The electrical power grid infrastructure in North America is evolving into a transcontinental network of cyber-physical systems, integrating power and energy systems with information and communication technologies for more efficiency, reliability, and sustainability. This ongoing cyber-physical integration, however, exposes a ubiquitous attack surface through which informed exploits may inflict major disruptions or damages to critical systems and processes [1]. As highlighted by multiple federal agencies and utility committees [2] [3], there is a high demand for advanced situational awareness (SA) to provide early warnings and protect electric utilities against adversaries from the cyberspace. The applicable SA solutions need to: (1) accurately capture traces of potential exploits from massive data streams in real-time; (2) automatically assess the situation and activate corresponding responses; and (3) adapt to variations of attack threats as well as system dynamics.

Data-driven detectors based on machine learning (ML) have stood out as a promising solution to such tasks due to their accuracy, automaticity, and adaptability in dynamical, time-varying, and uncertain environments [4]–[7]. Meanwhile, for these ML-based detectors, the quality of features, i.e., the attributes acquired from the data to describe the real-time status of a system-of-interest (SoI), is pivotal to their successful application in the smart grid. To promptly and precisely determine if a SoI is in normal operation, under faults, or being attacked, ideal features need to be sufficiently discriminative, which requires explicit physical models and refined human expertise. For the smart grid, however, such requirements are becoming increasingly challenging and expensive, as the cyber-physical integration creates coupled systems and patterns where we may not be able to explicitly specify the most effective features [8].

Recent advancements in deep learning (DL), a subfield of ML that leverages artificial neural networks in depth to directly extract features from raw data, provides a promising alternative for the data-driven detectors. Using a technique termed feature learning, DL approaches such as autoencoders [9] are capable of extracting novel features in an unsupervised, self-guided manner. Given a set of data samples with raw features as the input, DL creates and refines a set of generative features to reproduce the same dataset at the output. The raw features may consist of original measurements from field sensors as well as event logs from deployed recorders; the generative features are tuned by feature learning techniques to minimize the input-output differences so the original data can be recovered directly from the generative features. While their explicit physical meaning may not be available, these new features are alternative representations of system states learned by machines instead of being defined by humans. And despite its recency, the integration of feature learning has attributed to numerous successes of DL in challenging tasks, from Google's AlphaGo that beat world champions in the Go game [10], to accurate French-English natural language translation [11] and context-aware image understanding [12], paving the way for self-driving cars, data-driven disease diagnosis, and others.

In the smart grid, however, the utility of DL remains to be explored and exploited [13], [14]. Spatial-temporal patterns in a dynamic power grid are different and more complicated than those in images or videos taken in more predictable scenarios. Traces of the attacks are embedded in multi-modality data streams where there is limited knowledge on what features can better pinpoint footprints left by a malicious attack. To further tackle these challenges and advance the knowledge on how DL can benefit the safety-critical smart grid, the paper has proposed a detection framework using unsupervised feature learning to acquire and leverage machine learned-features from raw measurements. The attacks under investigation are informed exploits of transmission supervisory control and data acquisition (SCADA) systems that may lead to false alarms or unreported faults. Measurements collected from a high-fidelity, real-time digital simulator-based testbed will be used to validate the effectiveness of the proposed framework.

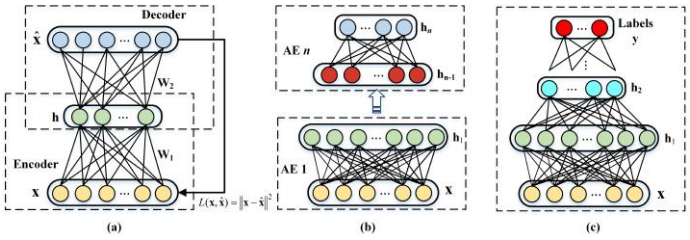

Fig. 1. Stacked autoencoders: (a) traditional autoencoder; (b) layer-wise unsupervised pre-training; and (c) supervised fine-tuning.

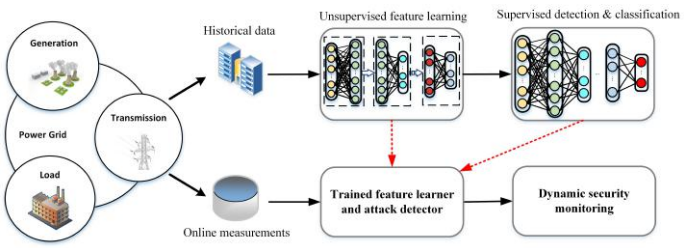

Fig. 2. Overall structure of the proposed framework for power grid security monitoring for attack detection and classification. The historical data is used for the attack feature learning and classification models training. The measurements from the SCADA system are feed into the trained models for online monitoring.

The major contributions of the paper to smart grid security will be two-fold:

1) An unsupervised feature learning framework was proposed for automatic and adaptive attack detection in transmission SCADA systems. The new design supplements machine-learned features to improve the detection accuracy with reduced reliance of system models and human expertise [15].
2) The design has been tested on datasets collected from a high-fidelity smart grid testbed in real time digital simulator (RTDS) [16]. Simulation results have validated that the design achieved higher detection accuracy in normal and attack cases with variations from load stress and measurement noises.

The rest of this paper is organized as follows: Section II introduces the deep autoencoder for unsupervised fault feature extraction. Section III reports the experimental results, including benchmark dataset introduction, extracted fault feature visualization, and comparative detection and classification results. Section IV draws the conclusions with future work.

II. Deep Learning-Aided Attack Detection and Classification

A. Deep Autoencoder for Feature Extraction

In recent works [17], deep networks have been applied to learn features over multiple modalities. Multi-modality learning involves correlated information from multiple sources, which motivates us to explore the inherent connections among various system measurements. Here in this paper, we use unsupervised feature learning techniques to automatically learn highly represented attack features from different raw SCADA data. Specifically, stacked autoencoders (SAE) based on traditional autoencoders (AE) will be used and introduced as follows.

Autoencoder: A typical autoencoder is a three-layer neural network that is trained to attempt to copy its input to its output. By doing this, the hidden layer $\mathbf{h}$ that describes a code can be used to represent the input. The whole network can be viewed as consisting of two parts: an encoder function and a decoder function, as shown in Fig. 1 (a). The encoder function is represented as:

$$\mathbf{h} = f(\mathbf{x}, \theta_f) = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \tag{1}$$

where $\theta_f = [\mathbf{W}_1, \mathbf{b}_1]$ is the parameter set containing a weight matrix $\mathbf{W}_1$ and a bias vector $\mathbf{b}_1$. Typically, a sigmoid function $\sigma(x) = 1/(1+\exp(-x))$ is used for the nonlinear deterministic mapping. The hidden layer code $\mathbf{h}$ can be viewed as a compression of input data with some loss when number of hidden units is less than the number of input units. It can capture the main variations in high-dimensional input data and eliminate less important information through dimension reduction.

Then, the hidden representation $\mathbf{h}$ is mapped back to a reconstruction output $\hat{\mathbf{x}}$ through the decoder as:

$$\hat{\mathbf{x}} = g(\mathbf{h}, \theta_g) = \sigma(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2) \tag{2}$$

where $\theta_g = [\mathbf{W}_2, \mathbf{b}_2]$. The training process of autoencoder is to find both optimal parameter sets θ_f and θ_g by minimizing the squared reconstruction error between $\mathbf{x}$ and $\hat{\mathbf{x}}$ as follows:

$$L(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^{N_T} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \tag{3}$$

where N_T is the number of input training data, and the learned features are embedded in the weight matrix, such as $\mathbf{W}_1$. Once trained, the new input data can be fed into the encoder to perform a nonlinear transformation and obtain the corresponding hidden representation $\mathbf{h}$ for subsequent tasks, such as classification and regression.

The idea of autoencoders has been part of the historical landscape of neural networks for decades [18]. Traditionally, autoencoders were used for dimensionality reduction or feature learning. Recently, theoretical connections between autoencoders and latent variable models have brought autoencoders to the forefront of generative modeling [19]. Autoencoders may be thought of as being a special case of feed forward networks, and may be trained with all of the same techniques, typically mini batch gradient descent following gradients computed by back-propagation.

Stacked autoencoders: A stacked autoencoder is composed of multiple AEs, in which they are treated as individual building blocks stacked in the deep architecture, with the aim of finding highly nonlinear and complex patterns in the data [9], [20]. In general, the whole training process of SAE includes

multiple unsupervised pre-training steps and supervised fine-tuning step, as shown in Fig. 1 (b) and (c). Given a set of training data, the learning of SAE is started with a greedy layer-wise pre-training procedure, which learns a stack of AEs in the encoder network. The key concept in the greedy layer-wise learning is to train one layer each time before starting to train its successive layer. As shown in Fig. 1 (b), the bottom layer AE is firstly trained with the raw data to obtain its hidden representations $\mathbf{h}_1$, and then the obtained hidden representations are used as the input data for training the higher-level AE, and so on. This pre-training process is task-free and focuses on the hierarchical representation learning from unlabeled data in a unsupervised manner.

After the layer-wise pre-training, all hidden representation layers are stacked and a logistic regression layer is added on top of the stacked autoencoders, creating a deep architecture as shown in Fig. 1 (c). The parameters of the whole deep network are first initialized by the corresponding parameters learned in the pre-training phase, and then are fine-tuned with labeled information using back propagation algorithm. Specifically, in order to speed up the learning speed, the batch stochastic gradient descent (SGD) method with momentum can be used to update the weights. In this way, the learned representations can capture more discriminative features in the explicit raw SCADA measurements.

B. Supervised Classifier Training and Online Application

In this section, we summarize the threat monitoring framework based on deep structured feature learning by SCADA measurements in the transmission system, as shown in Fig. 2. In the horizontal axis, there are three main steps: (i) historical or online data acquisition; (ii) unsupervised feature learning based on deep networks; and (iii) supervised event classification. In the vertical axis, there are two main phases: (i) off-line feature learner and classifier training based on historical data. The goal of the off-line training phase is to learn robust and discriminative representations and train a deep neural network classifier; and (ii) on-line threat monitoring based on real time measurements. The detailed procedure is summarized as follows:

Off-line training phase: Step 1: Collect historical data from different system operating conditions as the training data set; Step 2: Perform automatic representation learning from the raw dataset using the deep network structure in a unsupervised learning manner, and obtain robust and high-order feature representations; and Step 3: Stack all representation layers and add a classifier layer on the final representation layer to form a deep neural network model, and train it by using parametric learning algorithm, such as back propagation, to fine tune all parameters in a supervised manner.

Online monitoring phase: Step 4: Acquire online measurements from SCADA in the transmission system; and Step 5: Input these measurements to trained feature learner and classifier, and obtain the results for further advanced applications, such as situation awareness.

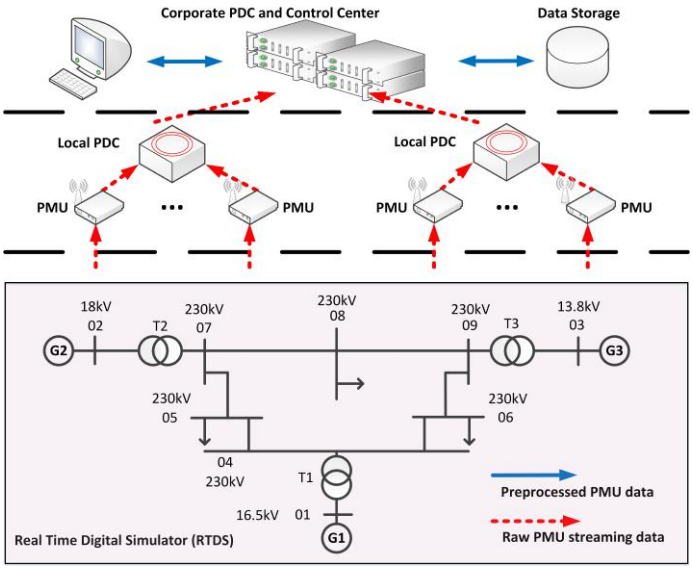

Fig. 3. Overview of the benchmark testbed: a physical layer with simulated power grid using RTDS and a cyber layer with implemented industry-standard data communication, processing, and storage.

III. Experimental Results

A. Benchmark System

The benchmark system in consideration is adopted from [16] and shown in Fig. 3. It is a high-fidelity security testbed built on the WSCC 9-Bus test system with extended modules representing the cyber-physical systems and processes in a smart grid. The RTDS environment simulates four phase measurement units (PMU) and relay-equipped substations, which are operating on commercial control and monitoring devices, hardware, software, as well as industry-standard communication networks and protocols. Heterogeneous sensor data are collected from the testbed and labelled to train the SAE and test its performance. The dataset contains two classes of events, i.e., normal operations with load variation and cyber attacks with control responses, which both have 128 features composed of measurements and event logs collected from the testbed.

Physical power grid: The physical power system is simulated using a RTDS, which is able to emulate electrical machines, controllers, transmission system components, and system load accurately and also provides a hardware-in-the-loop (HIL) simulation environment. The integration of virtual, simulated, and actual hardware components in HIL can capture the essence of the entire power system operation. One of the most important physical components of the testbed is the PMUs approximating an effective wide area measurement system (WAMS). The testbed also consists of hardware relays. The PMU and relays are hardwired from the RTDS back plane. Also, three physical over current and distance protection relays are incorporated in the system. The remaining required relays are modeled as software relays. In addition, the testbed

consists of a hardware Phasor Data Concentrator (PDC) and all PMUs are configured to stream data to the PDC.

Cyber communication system: The communication infrastructure in the benchmark testbed includes a physical network, communication protocols used for transporting measurements and control signals from device to device and between control centers and substations. The RTDS, PMUs, PDC, relays, Corporate PDC, attacker PC, and historian are connected via a network switch which supports copper and fiber optic connections. The RTDS and other substation devices such as relays and PMU communicate using various network protocols. The main communication protocol used for wide area based monitoring system is IEEE C37.118. The relays, PMUs, and PDC setting are configured and monitored using software packages from General Electric (GE) and Schweitzer Engineering Laboratories (SEL). The PMUs are configured to stream synchrophasor data to a GE P30 PDC with configurable data rates up to 120 samples per second. At the substation level, local control is employed using a HIL configuration using relays. Over current and distance protection relays are employed to control the breakers to protect the system from faults. Hence, the testbed incorporates centralized and local controls with industry standard software and hardware to model power system behavior, collect measurements, collect device status from field devices, forward operator commands to field devices, and manage historic data.

B. Feature Visualization

With features learned from the data, we first evaluated their quality for our purpose, which is whether the learned features provided more discriminative information for the two classes. The t-Distributed Stochastic Neighbor Embedding (t-SNE) [21], a well-developed tool for feature visualization, has been employed for this task. The quality of features is evaluated by the density and separability of data points from different classes when samples are mapped by t-SNE onto a lower-dimensional space. With 56 normal and 56 attacked samples, Fig. 4 shows the normalized t-SNE mapping of the original 128 features, 64 learned features, and 32 learned features, respectively. Each normal and attack samples are mapped into red circles and black crosses, respectively, with an initial number of dimension of 50 and a perplexity of 30. Note that the t-SNE mappings are used for feature evaluation only; no classification was performed on these mapped data. More details of t-SNE can be found in [21].

From Fig. 4, we can see that the extracted features are able to provide better distinction between normal and attacked samples. A significant amount of normal samples are overlapped with attacked ones in Fig. 4 (a); these overlaps are removed after 32 machine-learned features are introduced in Fig. 4 (b) and the separation is enhanced with 64 machine-learned features in Fig. 4 (c). Numerically, the cost value of t-SNE, which is the sum of Kullback-Leibler divergences to be minimized, was reduced from 0.1463 with the original features to 0.0728 with 32 learned features (50.2% improvement) and 0.0359 with 64 learned features (75.5% improvement), respectively.

C. Comparative Results

One multiclass dataset consisting of 4966 samples is separated into 36 individual datasets corresponding to the sub-types in [16]. These 36 datasets are balanced against no-event data, each set then being fed into a SAE with encoded data from each layer being fed into a multilayer perceptron (MLP) for binary classification. The classification accuracy of each sub-type then is combined by computing the weighted average in order to obtain results corresponding to the accuracy of each event grouping below. The results are then compared against the accuracy obtaining by training the MLP for classification on all 128 features.

Event groups:

- *Natural Events* – Short circuit fault within the power lines occurring anywhere in the system.
- *Data Injection* – Data corresponding to the short circuit fault in natural events is fed into the system.
- *Remote Tripping Command Injection* – Commands induce tripping of breakers in the system.
- *Relay Setting Change* – Relay parameters are altered to prevent tripping in the case of valid commands or faults.

Table I summarizes the accuracy of the methods in comparison. The introduction of feature learning achieves over 96% in accuracy against three different types of attacks, outperforming the supervised detectors by a small margin [15]. This competitive performance would be beneficial as less details of

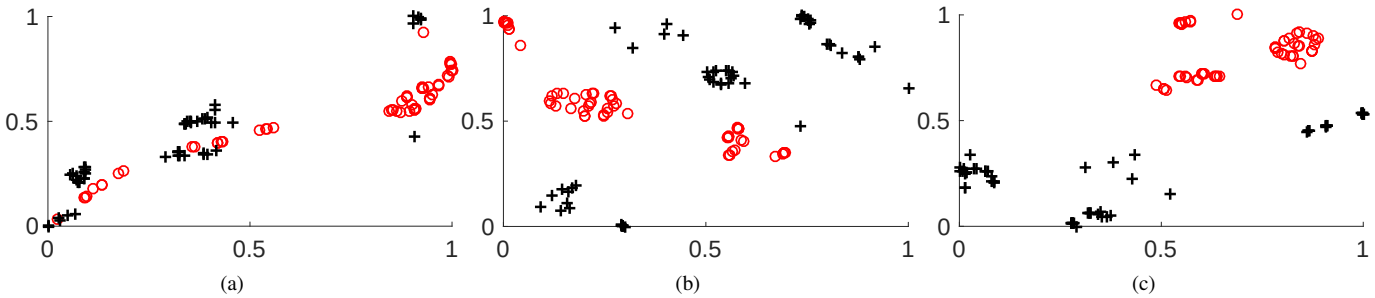

Fig. 4. Visualization of (a) 128 original features, (b) 32 learned features, and (c) 64 learned features, where 112 samples of two classes (56 each) are mapped into two-dimensional space using parametric t-SNE, respectively. The learned features provide better separation of normal and attacked samples.

TABLE I
COMPARISON OF ACCURACY FOR DYNAMIC ATTACK CLASSIFICATION BY USING DIFFERENT DIMENSION OF FEATURES

Event ID	Event Type	Original 128 Features	With 64 Features	With 32 Features	Results Reported in [15]
1	Normal operation with load variation	99.71%	99.72%	**99.89%**	Around 96.00%
2	Data Injection	98.77%	**99.75%**	98.94%	
3	Remote Tripping Command Injection	94.91%	96.79%	**97.34%**	
4	Relay Setting Change	**98.59%**	98.48%	98.53%	

system model or human expertise is required in constructing the effective detector. Also, comparing with the cases where feature learning were not applied, we observe that feature learning improves performance more significantly on Event 3, the remote tripping command injection attacks, than the other three events. The competitive results are favorable in real-world implementations as a reduced number of features will decrease number of inputs and the computational complexity of attack detectors and classifiers.

IV. Conclusions And Future Work

This paper presented a novel framework for attack detection and classification in the smart grid. Using PMU data and event logs collected in a high-fidelity WAMS benchmark, machine-learned features were created and refined with unsupervised feature learning for transmission SCADA attack detection and classification. Stacked autoencoder-based unsupervised feature learning were proposed for shared representation learning. This automatic process can capture useful and rich patterns hidden in the data to identify the attacks, achieving competitive results compared with detectors relying on detailed system models and human expertise. The preliminary results demonstrated that the proposed framework has the potential to provide adaptive and automatic threat monitoring in complex smart grid applications.

In practice, the proposed framework can be implemented as a secondary defense line for supplementary threat monitoring. However, it is notable that the smart grid has multiple sources of uncertainty and variation from subsystems like renewable energies, electric vehicles, which will result in a wide range of system operation points. This could cause insufficient or imbalanced training samples and degrade the online detection performance. In addition, the framework can be improved to not only detect the event but also locate the event to each line. The machine-learned features can also be combined with human-defined features in a more complex setting to combine the knowledge acquired by both humans and machines for accurate, automatic, and adaptive attack detection in the smart grid.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) under grants N00014-17-1-2109, and Florida Center for Cybersecurity (FC^2) Collaborative Seed Award.

References

[1] A. Ashok, M. Govindarasu, and J. Wang, "Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1389–1407, July 2017.

[2] J. McCarthy, O. Alexander, S. Edwards, D. Faatz, C. Peloquin, S. Symington, A. Thibault, J. Wiltberger, and K. Viani, "Situational awareness for electric utilities," *NIST SP 1800-7 Practice Guide*, 2017.

[3] M. Govindarasu, A. Hann, and P. Sauer, "Cyber-physical systems security for smart grid," *PSERC, Future Grid Initiative White Paper*, Feb. 2012.

[4] M. Ozay, I. Esnaola, Y. Vural, S. Kulkarni, and H. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, Aug 2016.

[5] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1644–1652, Sept 2017.

[6] H. Sedghi and E. Jonckheere, "Statistical structure learning to ensure data integrity in smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1924–1933, July 2015.

[7] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 1395–1402.

[8] X. Yu and Y. Xue, "Smart grids: A cyber-physical systems perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, May 2016.

[9] G. Hinton and R. Salakhutdinov, "Reducing the dmensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[11] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] C. Rudin, D. Waltz, R. Anderson, A. Boulanger, A. Salleb-Aouissi, M. Chow, H. Dutta, P. Gross, B. Huang, S. Ierome, D. Isaac, A. Kressner, R. Passonneau, A. Radeva, and L. Wu, "Machine learning for the new york city power grid," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 328–345, Feb 2012.

[14] Y. Tang and J. Yang, "Dynamic event monitoring using unsupervised feature learning towards smart grid big data," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 1480–1487.

[15] U. Adhikari, T. H. Morris, and S. Pan, "Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection," *IEEE Transactions on Smart Grid*, 2016.

[16] U. Adhikari, T. Morris, and S. Pan, "Wams cyber-physical test bed for power system, cybersecurity study, and data mining," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2744–2753, Nov 2017.

[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and Y. N. Andrew, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. CRC Press, 2016.

[20] G. Jiang, H. He, P. Xie, and Y. Tang, "Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2391–2402, Sept 2017.

[21] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.