Time Delay Attack Detection using Recurrent Variational Autoencoder and K-means Clustering

Shahram Ghahremani, David K.Y. Yau, Justin Albrethsen, Rajvir Sidhu, and Ngai-Man Cheung
Singapore University of Technology and Design
{ shahram_ghahremani, david_yau, justin_albrethsen, rajvirkaur_sidhu, ngaiman_cheung } @sutd.edu.sg

2021 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia) | 978-1-6654-3339-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISGTASIA49270.2021.9715557

***Abstract*—Conventional security methods deployed in power plants have difficulty detecting time delay attacks, since they do not alter network packets. However, these attacks can cause damage and instability in power systems, so detecting them is an urgent anomaly detection problem. Current state-of-the-art anomaly detection methods employ machine learning (ML) or statistical regression models in a supervised fashion, which require large amounts of labeled data for training. This data may be hard to practically obtain, so it is preferable to use unsupervised methods, which do not need labeled data. However, unsupervised anomaly detection solutions suffer from high false positive rates, especially under weak and moderate attacks. To improve on existing unsupervised solutions, we develop and present a dual-stage anomaly detection method using a Recurrent Variational Autoencoder (RVAE) and K-means clustering for detecting time delay attacks in power systems. We focus on samples including weak or moderate attacks which existing solutions cannot accurately detect, but can still harm the system if strategically targeted. We call these cases *borderline samples*, and use an additional clustering enhancement to more accurately classify them. Evaluation of the proposed approach on our power plant dataset demonstrates that our approach is effective in detecting time delay attacks, with 14.7% higher area under the ROC curve (AUC) than RVAE for borderline samples.**

I. Introduction

A modern cyber-physical system (CPS) uses network communication technologies for monitoring, automation, and control of physical components. Power systems are a subset of CPSes that rely on communication between spatially distributed sensors and actuators [1]. This connectivity of distributed components poses a cybersecurity challenge to protect communications and data processing.

Motivated by the above security challenge, this paper studies the detection of time delay attacks on a CPS. Time delay attacks are an important type of attack which threaten critical infrastructure such as power systems. In a time delay attack, valid data is maliciously delayed in communication links, causing a CPS to use stale information for time-critical control. This can disrupt control loops leading to system instability and damage to components. While these attacks can be easily detected with secure clock synchronization, this can be difficult to implement in a distributed CPS. Since these attacks can disrupt critical infrastructure, detecting delay attacks is an urgent and relevant problem.

This research was supported in part by the National Research Foundation, Singapore, and the Energy Market Authority, under its Energy Programme (EP award no. NRF2017EWT-EP003-061), and in part by the SUTD-ZJU IDEA Programme (award no. 201805).

Since these attacks are reflected in time-series sensor measurements, they can be treated as an anomaly detection problem [2]. Anomaly detection problems have plenty of existing solutions, such as statistical models or machine learning algorithms [3]. However, a real CPS is complex and difficult to realistically model. Furthermore, most machine learning models to deal with time delay attacks rely on labeled data for training, which may be difficult to generate for every attack scenario, and tedious to label [4], [5]. Unsupervised machine learning algorithms do not require labeled data and have been successful in several anomaly detection applications with CPS data [6]. Between unsupervised methods, a particular family of architectures based on deep autoencoders has recently gained the attention of the research community [7], [8]. Autoencoders consist of two parts, the encoder, which maps the input into its latent representation, and the decoder, which attempts to reconstruct the features back from the latent representation. Once the parameters of the autoencoder are optimized, it can be used for anomaly detection, where a given sample is recognized as abnormal when the reconstruction error is higher than a predefined threshold.

Most existing threshold-based detection algorithms require manual estimation of the threshold parameter based on historical data. This threshold can seriously affect detection efficiency and accuracy [9]. Moreover, sometimes attackers can learn the system configuration and design malicious attacks to bypass the threshold criteria. For example, an attacker may design an attack in such a way that the manipulated signal has a small reconstruction error. Samples containing this type of manipulated attack would be considered normal by the autoencoder. Another concern is that the autoencoder may not encounter every type of normal behaviour during training, and may generate high reconstruction error when such behaviours are present. These scenarios could cause false alarms leading to higher operational costs. This suggests that fixed threshold-based autoencoders are imperfect, and may have difficulty detecting anomalies, especially ones that are strategically designed by attackers.

In this paper, we attempt to resolve the problems with threshold-based autoencoders. Our observations show that a high percentage of false detection happens close to the detection threshold, which we call $borderline$ samples. These samples are usually associated with weak or moderate time delay attacks, which make them difficult to be detected. They are important to detect because they may harm the system

978-1-6654-4407-1/21/$31.00 © 2021 IEEE

if targeted under the right circumstances, or if sustained long enough. We propose a dual-stage anomaly detection approach to detect such challenging anomalies. In the proposed approach, first, the reconstruction error (anomaly score) of the given sample is provided by a trained Recurrent Variational Autoencoder (RVAE). If the reconstruction error is close to the predefined threshold (T), the classification is less accurate. To avoid false positives, we propose an additional step using K-means clustering to further investigate borderline samples.

Our paper makes the following contributions to defense against time delay attacks in CPSes:

- We present an anomaly detection technique using RVAE for non-borderline samples and K-means clustering for borderline samples. The K-means clustering is based on the reconstruction error, and statistical features of the input and reconstructed signal.
- Our clustering enhancement improves RVAE's area under the ROC curve (AUC) by 14.7% for borderline samples. Overall performance improvement will be dependent on what proportion of the dataset is borderline.

In the remainder of this paper we will provide more background on VAE in section II, and a description of our CPS and threat model in section III. Section IV will introduce our simulation design, followed by our anomaly detection framework, and evaluation methods. Our results are presented in section V, before our findings are concluded in section VI.

II. Background of Variational Autoencoder

A deep neural VAE [10] is similar in architecture to a regular Autoencoder (AE). However, unlike a regular AE, a VAE is a generative model and can be used to generate new data, which is important since we cannot train for every scenario. Recent work has demonstrated that the VAE performs much better than AE and principle component analysis (PCA) on handwritten digit recognition and network intrusion detection [11]. In both AE and VAE, reconstruction error can be used as a sign of anomaly where the trained AE or VAE generates high reconstruction error for abnormal samples. In many studies, the threshold of anomaly scores is used to distinguish whether the received signal is malicious or not [7], [8]. In AE and VAE, the anomaly score is defined as reconstruction error which is the mean square difference between the observed signal (P) and the expectation of their reconstruction (P').

As we are dealing with sequential data, we adapt the RVAE architecture used in Kim et al. for anomaly detection [7]. This detection method can be trained on normal data and detect anomalies that vary over time. RVAE is the structure of combining seq2seq with VAE, whose encoder and decoder consist of the auto-regressive model. RVAE utilizes recurrent neural networks (RNN) instead of a multilayer perceptron (MLP) or convolutional neural network (CNN) to generate sequential output. As such, it not only takes the current input into account while generating but also its neighborhood. RVAE uses a Gaussian prior distribution similar to VAE. The last hidden state is used as the mean and variance of multivariate Gaussian in latent space. The latent variable is employed as the initial hidden state of the decoder. A more detailed discussion of RVAE can be found in Kim et al. [7].

III. System Model

The proposed RVAE model will be trained on sequential time series data generated from a CPS. To detect anomalies we must consider our CPS, where the attack can be launched, and where the anomaly will manifest. Our specific CPS will be a distributed power system, with a controller that collects measurements from sensors, and sends control commands to the actuators to maintain stable system operations. The system is subjected to various disturbances, such as measurement noise, actuation biases, setpoint changes, etc. We will apply the proposed anomaly detection approach to the automatic generation control (AGC) of this power system [12]. AGC is a critical component of power grids whose complexity is representative of real-world CPS control problems.

Fig. 1 shows that sensors are used by AGC to control the generators. AGC maintains the grid frequency at its nominal value by adjusting the input mechanical power setpoints of the generators. AGC also maintains the net power interchanges among neighboring areas at scheduled values. Here, an area is a part of the grid and is usually operated by a utility. Separate areas are connected and share power through tielines.

AGC, located in the grid control center, receives measurements of grid frequency deviations (from the nominal frequency) and the i^{th} area's power export from their respective setpoints (which are denoted by $\Delta\omega_i$ and ΔP_{Ei}). It then computes the area control error (ACE) as $ACE_i = \alpha_i \Delta\varphi_i + \beta_i \Delta p_i$ where α_i and β_i are two constants. The control center sends ACE signals to the area's power plants over the communication network. The controller updates ACE every AGC cycle which is often two to four seconds, and sends it to generators to determine their setpoints.

We use a 37-bus system with three areas as a case study. Its scale corresponds to a small-/mid-scale grid in real life. According to Lou et al., a major fraction of 130 national grids consist of fewer than 37 buses [2]. The AGC-based 37-bus system is simulated in PowerWorld [13], an industry strength power system simulator used by grid operators.

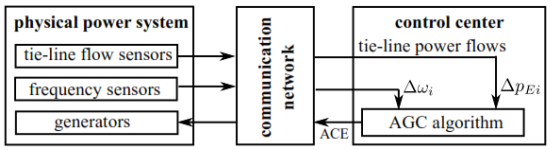


Fig. 1: **Overview of AGC**

A. Threat Model

In AGC, the sensor measurements and ACE signals are transmitted in long-range communication networks that are susceptible to cybersecurity threats. In this paper, we focus on the delay attacks against transmission of ACE signals. However, our approach can be readily applied to delay attacks

on sensor measurements or both ACE signals and sensor measurements. The delay attack can be conducted by jamming communication channels or using a compromised router to delay control commands from the AGC controller to the actuators. This effectively adds a buffer in the control signal communication line where packets are stored for some delay before they are released in order. In this paper, we assume that the clocks of the controller and the actuator are not synchronized, so timestamps cannot be compared to detect time delay attacks.

B. Attack Impact and Objective

Usually, the impact of the delay attack progresses gradually, which can be seen as sensor measurements and system frequency deviate from their normal state. A sustained delay attack causes deviations to diverge and may lead to an unsafe system. A system is $unsafe$ if its state (grid frequency) violates a specified range([a_{min},a_{max}]) . Safety is naturally a key concern of system operators because devices are designed to function properly only within specified ranges. Crossing these ranges may damage the devices or cause system failures. Fig. 2 illustrates the trajectory of tieline sensor measurements and grid frequency before and after a time delay attack with a magnitude of 8 cycles in the power control loop of an IEEE 37 bus system. It can be observed that the deviation in the tieline and frequency at the early stages after the attack are similar to the inherent (natural) fluctuations and within the safety limits. This makes it difficult to detect these attacks at the early stages. We can observe that the undetected attack remains in the system which makes the system unsafe at time 400s. If the attack attributes could be identified earlier, the control system can adopt a proper mitigation policy to avoid any adverse disturbance in the whole system.

Different time delay attacks have a different impact on the system. Some attacks with a small magnitude or short duration do not affect the system. However, some attacks with a large magnitude or long duration may result in an unsafe system. In table I, we group the various time delay attacks based on their impact on the system frequency. The $negligible$ attacks have a minimal impact on the system with minor frequency deviations in the range of $[-0.25, 0.25]$ Hz. Since these attacks have such little impact, they are not considered in our evaluations. $Weak$ attacks have the potential to make the system unsafe and are still important to detect. Finally, $moderate$ and $strong$ attacks do result in an unsafe system. These attacks have a high magnitude and duration and must be detected as early as possible. Our ultimate objective in this paper is to detect time delay attacks before the system becomes unsafe and therefore allow for proactive mitigation.

IV. Methodology

To detect time delay attacks in power plants, we must first design simulations to accurately represent real power systems. The simulation data is then used to train and evaluate our proposed solution.

TABLE I: Categories of attack samples.

Attack Type	Frequency Impact Range (Hz)
Negligible	[-0.25, 0.25]
Weak	[-0.5, 0.5]
Moderate	[-0.75, 0.75]
Strong	[>-0.75, 0.75<]

A. Simulation Design

To generate a realistic power system dataset, we use PowerWorld [13] to simulate a three-area 37-bus model as described in section III. Default settings and constants are used where: a_{min} = -5 MW, a_{max} = 5 MW, α_i = 12, β_i = 100 MW/Hz, and the AGC gain $K_i = 10^{-4}$ [2]. To make simulations realistic, zero-mean Gaussian noises are introduced that simulate disturbances to the system.

To generate a comprehensive dataset, a total of 150,886 simulations were executed. Simulation duration was set to 560 seconds, or 140 AGC cycles, where each AGC cycle is 4s. The tieline sensor measurements from each simulation were grouped into sliding windows W_i with a length of 74 data points $(P_1, P_2, \ldots, P_{74})$ and a sliding step of 5 data points. To train the RVAE model, we use only normal data to learn the natural system's behavior. However, to evaluate the performance of the proposed model, we introduce time delay attacks on 15% of simulations. The magnitude of malicious delay is randomly selected between 1-10 AGC cycles. The attack start and attack end are randomly selected in the range of 40s-350s and 350s-560s, respectively. Within a simulation, a window is considered *true abnormal* if at least 10 percent of the window is under attack; otherwise, the window is considered *true normal*.

B. Anomaly Detection Framework

After our training dataset is generated, we may train our RVAE and use it within our anomaly detection framework, which can be visualized in Fig. 3. First, we extract important information from the power grid AGC to obtain a set of features. For example, in Fig. 2 we see that tieline measurements deviate quickly following a delay attack, suggesting they will make good features. Once features are selected, they are segmented into time windows and fed to the RVAE to learn the normal behavior of the system. After the system is trained, the RVAE can make real-time predictions using a sliding window of input features. Reconstruction error, $E(P)$, from the RVAE is then compared to a set of detection thresholds, T_u and T_l, where T_u is the upper threshold and T_l is the lower threshold. If the reconstruction error is above T_u, we classify the sample as *predicted abnormal*. If the error is below T_l, the sample is classified as *predicted normal*. However, if the error falls between the two thresholds ($T_l \leq E(P) \leq T_u$) then the sample is considered $borderline$, and additional K-means clustering is applied.

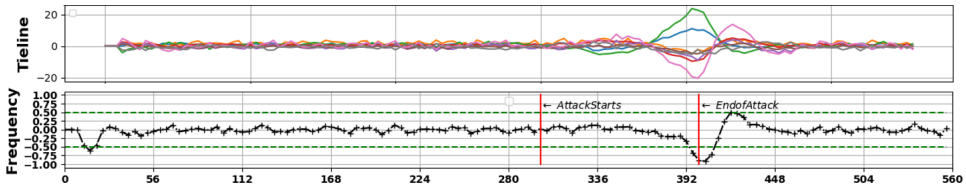


Fig. 2: **Impact of malicious time delay attack on system frequency and tielines profile. The attack starts at 300s and ends at 400s. The system goes to the unsafe zone at 396s.**

The two thresholds T_u and T_l are chosen based on our ROC curve to bound $borderline$ samples, which cannot be accurately classified based on reconstruction error. For our case, we choose T_u = 0.03241, where the true positive rate (TPR) is 1, and T_l =0.01928, where the false positive rate (FPR) is 0. Fig. 4 shows the reconstruction error distribution for borderline samples. For aesthetic purposes, we have scaled the error range from 0 to 1, however, the true error range is between T_l to T_u. The figure shows a significant overlap between the error of the *true normal* samples and *true abnormal* samples. This illustrates why we cannot use a reconstruction error threshold to detect attacks among borderline samples. To overcome this challenge, we apply our K-means clustering enhancement.

For our K-means clustering, we extract first-order statistical features, F, from tieline flows, P, and from the reconstructed signal, P'. We utilize four features: standard deviation, minimum, maximum, and mean. These features along with the reconstruction error, $E(P)$, create a feature vector of [$F(P)$, $F(P')$, $E(P)$]. This feature vector is then used by K-means clustering to classify the borderline samples as either *predicted normal* or *predicted abnormal*.

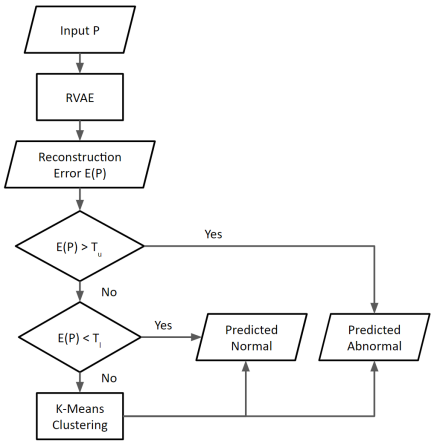


Fig. 3: **Overview of the anomaly detection framework. P represents the input features, E(P) represents the reconstruction error, T_l and T_u are the lower and upper thresholds. Samples with errors between the two thresholds are considered borderline and K-means clustering is used.**

C. Evaluation Criteria

We compare our proposed anomaly detection framework against that of conventional Recurrent AE (RAE) [7], RVAE

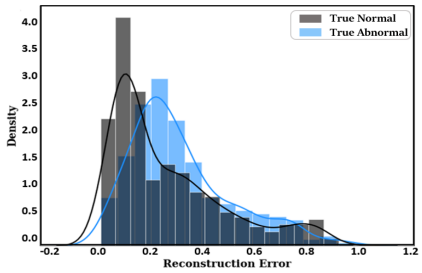


Fig. 4: **Distribution of reconstruction error among true normal and true abnormal borderline samples.**

[7], and a Gaussian-based Thresholding (GBT) approach. The RAE and $RVAE$ are based on a single threshold (T), where samples with large reconstruction error, above T, are flagged as anomalies. For the GBT, we fit independent but non-identical Gaussian distribution models to the features to learn the standard behaviors of the data. Then, we compute the Z-score for all features in the testing dataset and use the product of the average, standard deviation, and the maximum of the Z-scores as final scores for anomaly detection. Data points with a score above a threshold are considered anomalous.

For a fair comparison, the baseline RAE and RVAE share the same architecture and settings of the proposed RVAE. The RAE and RVAE approaches are implemented using Keras, a high-level open-source neural network library. For training the RAE and RVAE we use the same TensorFlow backend, a batch size of 128, and the stochastic optimizer Adam to minimize the reconstruction error (mean square error).

To compare performance we will use the AUC, TPR, and FPR. AUC represents the measure of separability, where higher AUC corresponds to a more accurate distinction between *true normal* and *true abnormal* samples.

V. RESULTS AND DISCUSSION

In the following subsections, we compare the performance of the proposed method to baseline models, then we evaluate our performance for the different groups listed in table I.

A. ROC Performance Evaluation

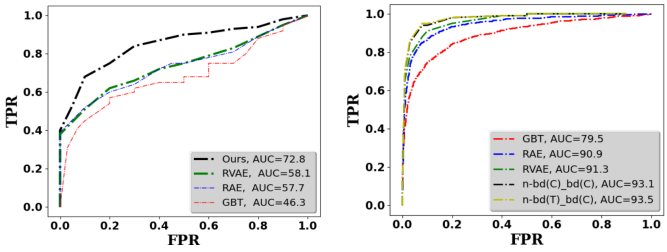
Fig. 5 illustrates the performance of the proposed method for borderline (5a) as well as overall (5b) cases. In section IV-B, we defined borderline cases as samples with reconstruction error in the range of $T_l \leq E(P) \leq T_u$, and perform enhanced clustering for such samples. It should be noted that

samples with *negligible* attacks are not considered for the results in this subsection.

Fig. 5a illustrates the performance of the baseline approaches for only borderline samples. We can observe that the clustering enhancement has improved AUC compared to RAE and RVAE by 15.1% and 14.7%, respectively.

Fig. 5b shows the performance of the proposed approach for both borderline and non-borderline samples. In our proposed approach (n-bd(T), bd(C)), we use the reconstruction error thresholds (T) for the non-borderline samples, and the clustering approach (C) for the borderline samples. We also try the clustering approach for both non-borderline and borderline samples (n-bd(C), bd(C)).

The results show that our clustering enhancement should only be applied to the borderline samples. Applying the clustering to non-borderline samples does not improve performance, because non-borderline samples contain strong attacks, which are more accurately detected based on reconstruction error. It can be observed that our proposed approach (n-bd(T), bd(C)) improves the overall AUC of RAE and RVAE by 2.6% and 2.2%, respectively. This performance improvement will depend on the percentage of borderline samples in the dataset.



(a) (b)

Fig. 5: **ROC curve of different approaches for (a) borderline samples, and (b) all types of samples.**

B. Granular Evaluation of the Proposed Method

Table. II illustrates the performance of the proposed method for the detection of types of time delay attacks described in Table I. Observe that detection performance is directly related to the strength of the attack. *Negligible* attacks are very difficult to detect for the same reason we do not count them, they have very little impact on the system. Our relatively high performance with *weak* attacks explains our success with borderline samples, as there is a high overlap between the two groups. These are important attacks that can harm the system if effectively targeted, and are typically challenging to detect.

VI. Conclusion

This paper presents an RVAE-based technique to detect time delay attacks in the control communication line of a power plant. In the proposed approach, the reconstruction error is first provided by a trained RVAE and compared to two thresholds to determine if the sample is normal or abnormal. If the error is between the thresholds, the sample is considered borderline, and a K-means clustering algorithm is used. To this end, the statistical features of the original sample, statistical features of the reconstructed sample, and reconstruction error are utilized to train the K-means clustering model. The proposed anomaly detection approach is compared with other baseline models. Comparison shows the proposed approach is more accurate than baseline models overall, with improvement among borderline samples. Compared to standard RVAE, our enhancement improves AUC for borderline samples by 14.7%.

TABLE II: Granular Performance of the Proposed Method

Attack Type	AUC	FPR	TPR
Negligible	43.2	0.484	0.211
Weak	71.4	0.210	0.811
Moderate	86.2	0.178	0.871
Strong	98.8	0.001	100

References

[1] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[2] X. Lou, C. Tran, R. Tan, D. K. Yau, and Z. T. Kalbarczyk, "Assessing and mitigating impact of time delay attack: a case study for power grid frequency control," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 207–216.

[3] A. Sargolzaei, K. K. Yen, and M. N. Abdelghani, "Preventing time-delay switch attack on load frequency control in distributed power systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1176–1185, 2015.

[4] S. Ghahremani, R. Sidhu, D. K Yau, N.-M. Cheung, and J. Albrethsen, "Defense against Power System Time Delay Attacks via Attention-based Multivariate Deep Learning," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), to be published.* IEEE, 2021, pp. 1–6.

[5] J. Albrethsen, S. Ghahremani, R. Sidhu, and D. K. Yau, "Managing Uncertainty in Deep Learning Predictions for Mitigating Time Delay Attacks," in *Submitted for publication*, 2021.

[6] Z. N. Anshuman, K. S. Sajan, and A. K. Srivastava, "ML-based Data Anomaly Mitigation and Cyber-Power Transmission Resiliency Analysis," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[7] J. Kim, A. Sim, J. Kim, and K. Wu, "Botnet Detection Using Recurrent Variational Autoencoder," *arXiv preprint arXiv:2004.00234*, 2020.

[8] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection," *arXiv:1903.06661 [cs, stat]*, Mar. 2019, arXiv: 1903.06661. [Online]. Available: http://arxiv.org/abs/1903.06661

[9] A. Laszka, W. Abbas, S. S. Sastry, Y. Vorobeychik, and X. Koutsoukos, "Optimal thresholds for intrusion detection systems," in *Proceedings of the Symposium and Bootcamp on the Science of Security*, 2016, pp. 72–81.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[12] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.

[13] "PowerWorld The visual approach to electric power systems." [Online]. Available: https://www.powerworld.com/