Deep Learning-Based Data Forgery Detection in Automatic Generation Control

Fengli Zhang, Qinghua Li
University of Arkansas
{fz002, qinghual}@uark.edu

Abstract - Automatic Generation Control (AGC) is a key control system in the power grid. It is used to calculate the Area Control Error (ACE) based on frequency and tie-line power flow between balancing areas, and then adjust power generation to maintain the power system frequency in an acceptable range. However, attackers might inject malicious frequency or tie-line power flow measurements to mislead AGC to do false generation correction which will harm the power grid operation. Such attacks are hard to be detected since they do not violate physical power system models. In this work, we propose algorithms based on Neural Network and Fourier Transform to detect data forgery attacks in AGC. Different from the few previous work that rely on accurate load prediction to detect data forgery, our solution only uses the ACE data already available in existing AGC systems. In particular, our solution learns the normal patterns of ACE time series and detects abnormal patterns caused by artificial attacks. Evaluations on the real ACE dataset show that our methods have high detection accuracy.

***Index Terms*—Power grid, AGC, data forgery attack, deep learning, attack detection.**

I. Introduction

Automatic Generation Control (AGC) is a critical control function of the power grid used to control the amount of power generation and maintain the balance between power generation and load, which keeps frequency at the scheduled value (i.e. 60 Hz in the U.S.). In AGC, a control center periodically monitors the power system's frequency and tie-line power flow between neighboring balancing areas (a balancing area is a part of the power system that balances the electric demand and supply within a geographic boundary), and adjusts the amount of power generation in each balancing area based on the collected information so that the change in generation can restore the frequency to the scheduled value. The required change in generation, called Area Control Error (ACE) is calculated based on the difference between measured frequency and scheduled frequency and the difference between measured tie-line power flow and the scheduled power flow. ACE is updated periodically every two or four seconds (which we call an *AGC cycle* for convenience) and then sent to the generators.

However, AGC is vulnerable to data falsification attacks [1-4]. The attacker can inject falsified frequency or tie-line power flow measurements to force miscalculation of ACE, which may deceive AGC to make some wrong control actions. For example, when ACE is positive, it means that the area is over generating and thus AGC will issue a command to decrease power generation. However, if the frequency or tie-lie power flow measurements are attacked resulting in negative ACE value, AGC will believe that the area is under generating and thus increase the power generation, which exaggerates the over generating situation. A few work [5, 6] have been done to detect such attacks. Those schemes use load forecast to predict the ACE and then compare the measured ACE value with the predicted ones to detect forged AGC measurements. However, load prediction is run every five minutes [7] (a much lower frequency than ACE calculation) and the prediction is never 100% correct [8], which will result in inaccurate ACE prediction and inaccurate attack detection.

The detection methods we propose in this paper only use the ACE time series data. Since ACE data is already available in current AGC systems, our detection methods can be easily deployed without interrupting service. Specifically, we propose two methods to do the detection. In the first method, we adopt Long Short Term Memory (LSTM) neural network to predict the ACE sequence pattern in the next detection window, compare with the corresponding ACE sequence pattern calculated from measurements, and then determine whether there is forged data in the sequence. The second method calculates the moving average of ACE time series and then converts the moving average data from time domain to frequency domain by using discrete Fourier transform. Then check if the data is normal in frequency domain. These two methods work well for different types of attacks. The two proposed methods are evaluated on the real ACE dataset obtained from company PJM, indicating their potential high performance in the real world.

This paper is organized as follows. Section II discusses related work about detecting data forgery attacks in AGC. Section III describes the system model and three attack models considered in this paper. Section

IV describes the proposed detecting methods. Section V shows the performance of the proposed methods on the real dataset. The last section concludes this paper.

II. Related Work

A few work have been done about attacks on AGC. In [9], the authors explore how to launch attacks to achieve expected effects in the shortest time, but no detection method is given. In [5], Sridhar et al. develop a model-based anomaly detection algorithm, in which the ACE values are predicted in 5-minute intervals for the next hour based on load forecast. The real-time value of ACE will be regarded as an anomaly if it is not in the forecasted range. This method heavily depends on load forecast. However, in the practical power system, load forecast have no high accuracy [8]. The work in [6] presents a two-tier intrusion detection system. The first tier forecasts the ACE value for the next time instance based on the current time instance. The measurement deviating from the prediction will be flagged as anomalous and then the flagged instance is passed to the second tier to verify anomaly by incorporating the overall system variable. However, the algorithm only depends on one previous observed value. If the previous data is abnormal or attacked, its prediction for the next time instance will be misled by the attacked data. Therefore, the attack in the next time instance may not be detected. The approach presented in [10] adopts a security game model to choose the best response strategies against attackers, but it does not give an approach to detect the attacks.

III. Preliminary and Attack models

A. AGC System

For a balancing area, AGC is used to adjust power generation to maintain the frequency at the scheduled value. AGC is an automated control system as illustrated in Fig. 1. It periodically (every 2-4 seconds) receives the measurements of frequency in this balancing area and tie-line power flows between this area and neighboring areas from field devices, and calculates the ACE according to the equation $ACE = (P_{tielie} - P_{sch}) + B(f - f_{sch})$, where $P_{tieline}$ and f_{sch} are the scheduled tie-line power flow and the scheduled frequency respectively. B is the frequency bias factor, which is constant and is estimated annually. Then AGC adjusts the power generation of generators according to the obtained ACE.

B. Attack Models

One stealthily forged data measurement may not be enough to introduce significant impact to the power grid. According to [9], the shortest time to stealthily mislead the system frequency to breach the safety condition without triggering AGC suspension is at least 10 AGC cycles, which means that in order to achieve expected

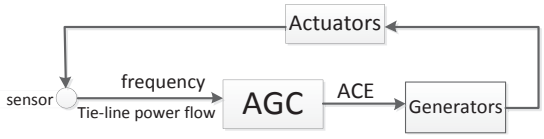

Fig. 1. AGC system

effects, the attacker needs to inject a series of false data to indirectly control the generator for a period.

In this work, we mainly consider three attack models explored in [5, 11]: *scaling attack*, *ramp attack*, and *random attack*. In the attack models, the attacker keeps launching attacks until achieving expected results. Let T_a represent the attack period, t represent time, t_0 represent the time point when the attacker starts an attack, $y(t)$ represent the true measurement (which could be either frequency or tie-line power flow as discussed later) value without attacks, and $y^*(t)$ represent the measurement value with possible attacks. The three attack models can be described as follows.

Scale Attack: This attack modifies the measurements by scaling up or down with a scaling parameter λ_s.

$$y^*(t) = \begin{cases} y(t), & \text{for } t \notin T_a \\ y(t) + \lambda_s \cdot y(t), & \text{for } t \in T_a \end{cases}$$

Ramp Attack: This attack modifies the measurements gradually with the addition $\lambda_r(t - t_0)$. λ_r is a ramping parameter. This type of attack is more difficult to detect because it has very small and unnoticeable changes at the beginning of the attack period.

$$y^*(t) = \begin{cases} y(t), & \text{for } t \notin T_a \\ y(t) + \lambda_r \cdot (t - t_0), & \text{for } t \in T_a \end{cases}$$

Random attack: This attack aims to add some random positive values in a range with lower bound a and upper bound b to the real measurements during the attack period.

$$y^*(t) = \begin{cases} y(t), & \text{for } t \notin T_a \\ y(t) + rand(a,b), & \text{for } t \in T_a \end{cases}$$

These attacks directly change the sensor measurements. It can be derived that ACE is proportional to tie-line power flow and frequency deviation. According to [12], we have

$$\Delta P_L = -(\Delta f)\sum_{i=1}^{n} \frac{G_{Ri} / f_0}{GD_i}$$

f_0, G_{Ri} and GD_i is the scheduled frequency, the rated generation capacity and the governor droop of generator i. All the three variables are constants for each power system. Assuming $K = -\sum_{i=1}^{n} \frac{G_{Ri}/f_0}{GD_i}$, then $\Delta P_L = K\Delta f$ and K is also a constant. Then, we can obtain:

$$ACE = \Delta P_L + B\Delta f = \Delta P_L + \frac{B}{K}\Delta P_L = (1 + \frac{B}{K})\Delta P_L$$

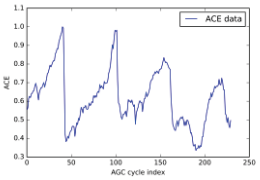

Fig.2. ACE data pattern

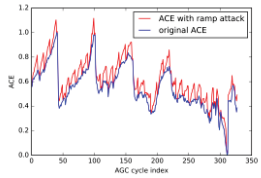

Fig. 3. ACE with Ramp Attack

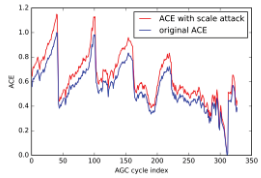

Fig. 4. ACE with Scale Attack

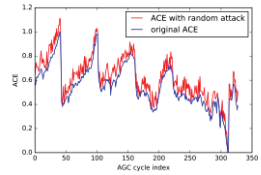

Fig. 5. ACE with Random Attack

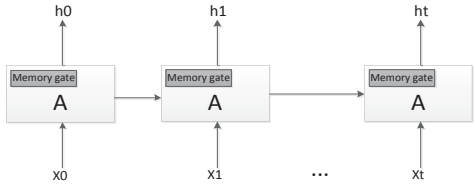

Fig. 6. Structure of LSTM

From the above formula, it can be known that ACE is linear to tie-line power flow and frequency deviation. Therefore, even though these attack models directly modify the sensor measurements such as tie-line power flow and frequency, they also have same modification trend on ACE. For example, if the attacker launches scale attack on tie-lie power flow or frequency, it can also be regarded as scale attack on ACE.

C. Dataset

The real ACE dataset is from PJM (PJM Interconnection) [13], an electric regional transmission organization (RTO). The dataset includes four years' ACE data, from the year 2012 to 2015, with about 2 million data records. Each record provides the ACE value and its date and time.

This dataset is normal data without any attack. Figure 2 shows the real ACE data pattern of 250 cycles. From the figure, it can be seen that the ACE data has some specific patterns. For example, the data sequence pattern from cycle 0 to 49 is similar to the sequence pattern from cycle 50 to 100. To generate attacked ACE data, we add attacks to the normal data based on the above attack models. The ACE data with the ramp attack (λ_r= 0.02) is shown in Fig. 3. The ramp attack is launched periodically every 10 cycles and each attack lasts for 10 cycles. The blue line shows the original data without attack, and the red one is the data with ramp attack. It can been seen that the ACE data sequence pattern has been totally changed. Figure 4 and Figure 5 show the ACE data with scale attack when λ_s = 0.1 and random data with rand (0, 0.1) respectively.

IV. Detection Methods

A. LSTM-based Detection

Based on the dataset observations, the ACE time series data of a balancing area have some patterns determined by the physical configuration of the AGC system (e.g., how ACE responds to load changes). If an attacker injects artificial data into AGC, the resulted ACE patterns will be different. Therefore, we can detect attacks through checking whether the ACE data patterns deviate from the normal patterns. Following this idea, we use neural network to learn the normal pattern of ACE time series and use it to detect attacks. To determine whether the pattern of the current data sequence is normal or has appeared before, the neural network model must have the ability to link the current observations with the past observations. However, traditional neural networks just focus on the current input and they cannot connect the current task with the previous information. Long Short Term Memory networks (LSTM), a special kind of recurrent neural network, is designed to make such connections.

The structure of LSTM is shown in Fig 6. x_t is the input, h_t is the output and A is neural network. This chain allows the past information to be passed from one step to the next. In each step, it has a memory part to remember useful information for a long period of time. It is well-designed to learn from past experiences and connect the previous data with the current [14]. LSTM is able to find which previous sequence pattern that the current sequence resembles or is similar to. Then it can predict the next data sequence pattern based on the resembled pattern as well as the current inputs.

Our experiments on the PJM dataset show that the LSTM model learned from past ACE time series data is able to make very accurate predictions for the next ACE sequence. We split the PJM ACE data into two parts, training data and testing data. Then training dataset (size: 1 million records) is used to train the model and then the testing dataset is fed into the model to do the prediction for each data point. The prediction results are shown in Fig. 7. The red ones are predicted data which fits the real data (blue ones) very well.

Since the LSTM model has very high prediction accuracy, we can compare the predicted ACE sequence with the measured ACE sequence to detect abnormal measurements. In each comparison, we calculate the distance between a predicted ACE data sequence with the corresponding measured ACE sequence by using Manhattan Similarity.

In particular, the LSTM model is first trained with historical ACE data and can be updated dynamically (e.g., every day or every week) to include the newly generated ACE data. Then the current ACE data sequence is fed into the trained model to do the prediction. The length of the input data sequence can be

adjusted based on the datasets to achieve better prediction results. Suppose the input length is m, and x_i is the i^{th} ACE measurement. We use the input data sequence $(X_{i+1}, X_{i+2}, \dots, X_{i+m})$ to predict X_{i+m+1}. When predicted data sequence with n values is available, it is compared with the measured data sequence at the same time points to check whether the measured data sequence deviates too much from it. The detailed steps of the method are shown as follows which are run every n ACE cycles.

Step 1: predict the next data sequence using the trained model.

$$X_{i+1}, X_{i+2}, X_{i+3}, \dots, X_{i+m} \rightarrow \hat{X}_{i+m+1}$$
$$X_{i+2}, X_{i+3}, X_{i+4}, \dots, X_{i+m+1} \rightarrow \hat{X}_{i+m+2}$$
$$\dots$$
$$X_{i+n}, X_{i+n+1}, X_{i+n+2}, \dots, X_{i+n+m} \rightarrow \hat{X}_{i+m+n}$$

Step 2: use Manhattan Similarity to compute the distance between the predicted sequence and the measurements.

$$d = \frac{1}{n} \sum_{j=1}^{n} |X_{i+m+j} - \hat{X}_{i+m+j}|$$

Step 3: compare the distance with the threshold Θ. If it is larger than the threshold (d >Θ), it is regarded as attacked data. Otherwise, it is normal data.

B. Fast Fourier Transform based Detection

The prediction accuracy of LSTM model mainly depends on data sequence pattern, but it is not very sensitive to data's value. Since the scale attack just scales the data's value up or down and does not change the data sequence patterns, LSTM model cannot detect such attacks very well.

However, if the data sequence values change unexpectedly, the change will be reflected on its average. We can calculate the moving average [15] of normal ACE data and attacked ACE data to observe their differences. Figure 8 shows the moving average. The blue line shows the moving average without any attack and the red one is the moving average with the scale attack (λ_s= 0.2) and the length of one attack period is 10 AGC cycles. It can be seen that the moving average of the data with scale attacks is more fluctuated than the normal data. Then we use Fast Fourier Transform (FFT) [16] to convert the moving average data from time domain to frequency domain to explore the fluctuation. For each 10-data sequence, we use FFT to convert its moving average to frequency domain and get the minimum transformed value (MTV) of each sequence. As shown in Fig. 9, the MTV of normal moving average is around 0.0. However, the MTV of moving average with scale attack is around -0.2. The MTVs of data under scale attack and normal data have very obvious differences. Then we can set a threshold. If a data sequence's MTV is larger than the threshold, it is normal data. Otherwise, it is regarded as attacked data. Such threshold can be set by observing the differences between MTVs of attacked data and normal data.

V. Evaluations

A. LSTM-based Detection

The data used in the experiments are as described in Section III.C. We split the data into two parts: 67% as training data and 33% as testing data. The training dataset is used to train the LSTM model, which has a hidden layer with 6 neurons and an output layer to make the prediction. The sigmoid activation function is used for the LSTM neurons. Here n is set as 10 because the shortest attacked sequence which can negatively influence the system is 10. The input data sequence size m is set as 5, which can be adjusted based on different datasets. The attacked data are generated by adding the attack periodically into testing data every 10 cycles. To test the model's performance, we feed the attacked data into the model to check the True Positive (TP) detection rate which is defined as the fraction of attacks successfully detected. We also feed the normal data without attacks into the model to see the False Positive (FP) detection rate which is defined as the fraction of normal data sequences falsely detected as attacked data.

The setting of the threshold Θ is critical. If the threshold is too low, some normal data sequences will be detected as attacked data. If the threshold is too high, the attacked data sequences may not be detected. The FP rates of different threshold settings are shown in Table 1. The higher is the threshold and the lower is the FP rate. In the following, we set the threshold as 0.3, which has a FP rate of less than 5%.

When Θ =0.3, the detection results for ramping attacks are shown in Fig. 10. The results show that when λ_r (lambda) is higher, the TP detection rate is also higher. This is because higher λ_r means the attacks have more significant modifications on ACE data and thus such attacks are easier to be detected (note that these attacks also have higher impact to the power grid). When $\lambda_r \leq 0.015$, the impact of attack is very small under this parameter setting, but our algorithm can still detect most of attacks.

In the random attack, the attacker tries to inject positive data to increase the ACE rather than decreasing it. Thus, we set the lower bound a as 0 which means the injected data are positive values which range in $(0, b)$. The TP detection rates for random attacks are shown in Fig. 11. The results have similar trending with ramp attack. Higher upper bound b means larger amount of modifications. Thus the detection rate is higher. When $b \geq 0.1$, the detection rate is above 92%.

Table 1. FP rates for different thresholds

Threshold	0.5	0.4	0.3	0.25
FP rate	0.2%	0.7%	4.5%	8.1%

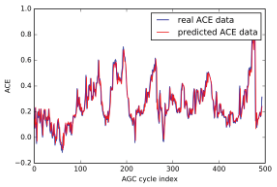

Fig. 7. Prediction with LSTM

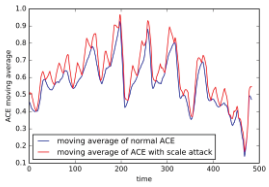

Fig. 8. Moving average of scale attacks

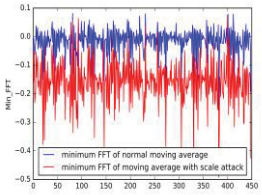

Fig.9. FFT of Moving Average with scale attacks

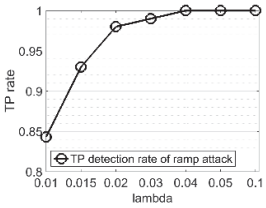

Fig. 10. TP rate for ramp attack with LSTM-based method

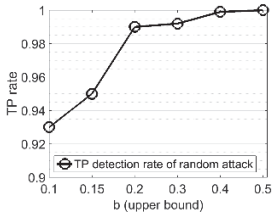

Fig.11. TP rate for random attack with LSTM-based method

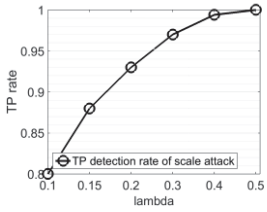

Fig. 12. TP rate for scale attack with FFT-based method

B. FFT-based Detection Method

We first calculate the moving average of the ACE data and divide them into subsequences. Each subsequence has 10 data points. Then we use FFT to transform each subsequence to frequency domain and get the MTVs for each subsequence. Then a threshold is set to separate the MTVs of attacked data from the ones of normal data. By observing the MTVs' distribution, the threshold is set as -0.08. If MTV is less than -0.08, the subsequence is regarded as attacked data. Otherwise, it is normal data. The FP rate is 5% when the threshold is -0.08. The TP rates are shown as Fig.12. When $\lambda_s > 0.2$, more than 90% attacks can be detected.

VI. CONCLUSIONS

In this work we proposed two methods, LSTM-based method and FFT-based method, to detect data forgery attacks in AGC. We test our methods on the real dataset and these methods achieve high detection performances. Both LSTM-based and FFT-based methods can detect about 90% attacks with less than 5% FP detection rate.

ACKNOWLEDGMENT

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000779.

REFERENCES:

[1] P.M. Esfahani, M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson. "A robust policy for Automatic Generation Control cyber attack in two area power network," In proc. of IEEE Conference on Decision and Control, 2010.

[2] Mohajerin Esfahani, P., et al. "Cyber attack in a two-area power system: Impact identification using reachability." American Control Conference (ACC), 2010. IEEE, 2010.

[3] Sridhar, Siddharth, and G. Manimaran. "Data integrity attacks and their impacts on SCADA control system." (2010): 1-6.

[4] Vrakopoulou, Maria, et al. "Cyber-attacks in the automatic generation control." Cyber Physical Systems Approach to Smart Electric Power Grid. Springer Berlin Heidelberg, 2015. 303-328.

[5] S. Sridhar, and M. Govindarasu. "Model-based attack detection and mitigation for automatic generation control." Smart Grid, IEEE Transactions on 5.2 (2014): 580-591.

[6] Ali, Muhammad Qasim, et al. "Two-tier data-driven intrusion detection for automatic generation control in smart grid." Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.

[7] Trudnowski, Dan J., Warren L. McReynolds, and Jeffery M. Johnson. "Real-time very short-term load prediction for power-system automatic generation control." IEEE Transactions on Control Systems Technology 9.

[8] Eli, T. "Iceman,(2012), Power grid operations." (2012).

[9] Tan, Rui, Hoang Hai Nguyen, and Hoay Beng Gooi. "Optimal False Data Injection Attack against Automatic Generation Control in Power Grids." In 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS), pp. 1-10. IEEE, 2016.

[10] Law, Yee Wei, Tansu Alpcan, and Marimuthu Palaniswami. "Security games for risk minimization in automatic generation control." Power Systems, IEEE Transactions on 30.1 (2015): 223-232.

[11] Huang, Yu-Lun, Alvaro A. Cárdenas, and Shankar Sastry. "Understanding the physical and economic consequences of attacks on control systems." International Journal of Critical Infrastructure Protection 2, no. 3 (2009): 73-83.

[12] Ebrahim Vaahedi. "Practical power system operation ebrahim." (2014)

[13] http://www.pjm.com/markets-and operations/etools/oasis/system-information/historical-area-control-error-data.aspx.

[14] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[15] https://en.wikipedia.org/wiki/Moving_average.

[16] Brigham, E. Oran, and Elbert Oran Brigham. *The fast Fourier transform*. Vol. 7. Englewood Cliffs, NJ: Prentice-Hall, 1974.