Two-tier Data-Driven Intrusion Detection for Automatic Generation Control in Smart Grid

Muhammad Qasim Ali[†§], Reza Yousefian[‡], Ehab Al-Shaer[†], Sukumar Kamalasadan[‡], Quanyan Zhu[♯]
[†]Department of Software and Information Systems
[‡]Department of Electrical and Computer Engineering
University of North Carolina Charlotte, Charlotte, NC
[♯]Princeton University, Princeton, NJ
[§]Symantec Corporation, Mountain View, CA
Email: {mali12, ryousefi, ealshaer, skamalas}@uncc.edu, quanyanz@princeton.edu

***Abstract*—Legacy energy infrastructures are being replaced by modern smart grids. Smart grids provide bi-directional communications for the purpose of efficient energy and load management. In addition, energy generation is adjusted based on the load feedback. However, due to the dependency on the cyber infrastructure for load monitoring and reporting, generation control is inherently vulnerable to attacks. Recent studies have shown that the possibility of data integrity attacks on the generation control can significantly disrupt the energy system. In this work, we present simple yet effective data-driven two-tier intrusion detection system for automatic generation control (AGC). The first tier is a short-term adaptive predictor for system variables, such as load and area control error (ACE). The first tier provides a real-time measurement predictor that adapts to the underlying changing behavior of these system variables, and flags out the abnormal behavior in these variables independently. The second tier provides deep state inspection to investigate the presence of anomalies by incorporating the overall system variable correlation using Markov models. Moreover, we expand our second tier inspection to include multi-AGC environment where a behavior of one AGC is validated against the behavior of the interconnected AGC. The combination of tier-1 light-weight prediction and tier-2 offline deep state inspection offers a great advantage to balance accuracy and real-time requirements of intrusion detection for AGC environment. Our results show high detection accuracy (95%) under different multi-attack scenarios. Second tier successfully verified all the injected intrusions.**

I. Introduction

Smart grids have been replacing the legacy power infrastructure as they provide efficient energy management by utilizing the bi-directional communications. Bi-directional communications enable the smart grid to take different sensor measurements using cyber infrastructure in order to control the power generation, transmission and distribution effectively and in real time. The bi-directional communications are associated with the supervisory control and data acquisition system (SCADA). An important task of SCADA is automatic generation control which is responsible for adjusting the power generation according to the load in the area.

Several threats have been targeted towards the SCADA system due to its dependency on the cyber infrastructure. According to a recent Bipartisan policy center report, a Washington D.C. think tank, more than 150 cyber attacks targeted energy sector in 2013 [1]. There can be several entry points for an attacker to enter the SCADA system and/or control center including malware attachment in the email, malware on the storage device and WiFi enabled system in SCADA and/or control center. Moreover, SCADA systems and control centers are connected to the corporate offices using virtual private network, therefore, anybody having access to the corporate office can access the system. Attacks can be launched by two types of attacker i.e., naive and experienced/knowledged. Naive attackers lack the working knowledge of the smart grid system. On the contrary, experienced attackers may manipulate the generation control measurements such that it still satisfies the smart grid environment and look benign/normal. Although bad data detection algorithms provide some security to identify data integrity attacks, recent studies have shown that these algorithms can be bypassed by experienced attackers [15]. Moreover, attacks having attack vector after state estimation i.e., AGC, can not be detected by these detection algorithms.

To this end, we present a data-driven two-tier intrusion detection approach. The first tier is an online short-term adaptive predictor for both the load and Area Control Error (ACE), which are system variables in AGC. Load measurements are taken by the field sensors. However, ACE is calculated, in AGC, using the frequency and tie-line flow measurements. Generation control takes these measurements every few seconds. The basic hypothesis is that both the load and ACE have different behavior at different times of the day, therefore, at short intervals they exhibit a certain level of temporal dependence which can be used to predict their future behavior. The proposed predictor has the ability to adapt to the change in behavior of the variables. Since load and ACE forms the basis of calculation of set points and lowering/raising the generation, respectively, we use a data-driven approach to predict these variables. We show the prediction accuracy of the proposed predictor under normal conditions in a well known and widely used two-area power system model. Since the predictor shows high accuracy under normal conditions, therefore, deviations from the prediction can be flagged as anomalous.

Prediction is done independently and does not take into account the other AGC system variables. Therefore, we build a Markov model of AGC using its system variables in the second tier of the intrusion detection system. The model incorporates the system-wide knowledge to detect anomalies. It observes the state transitions, where state is defined using multiple system variables, and calculates the individual variable probability given the system state. If the probability does not fall in the probability range learnt from the normal data, it raises an alarm. Two-tier approach provides the benefit of timeliness, using light-weight online prediction, and accuracy, by reducing the false positives offline using multiple AGC system variables. We also extend the second-tier to interconnected multi-AGC scenario. In this scenario, state of one AGC is verified by incorporating the other AGC's knowledge. This helps in identifying that which particular AGC is under attack and causing the system de-stabilization overall. We conduct multiple attack case studies which include tampering the system frequency and the load in order to mimic the integrity attacks by knowledged attackers. Tampered parameters still satisfy the generation control equations. Our approach detected all of the integrity attacks successfully.

The remainder of the paper is organized as follows: Section II discusses the related work. Background of smart grid and motivation is described in Section III. Analysis and adaptive predictor is presented in Section IV followed by the anomaly verification module in Section V. Section VI shows the evaluation results along-with the attack model and Section VII concludes the work.

II. Related Work

Since we propose AGC parameters' prediction based intrusion detection approach, we discuss work related to prediction of AGC parameters and intrusion detection. There is no data-based prediction approach that can predict all the AGC parameters. However, there are approaches that can predict short term future load, a AGC parameter, [9], [18]. The work [18] presents a Kalman filter based load prediction approach. The approach does not take into account the temporal dependence in AGC data. Furthermore, it assumes that the introduced noise in AGC data is white, which may not be the case in real life. On the contrary, our approach takes into account the temporal dependence in the data and it is specifically designed to classify anomalous behavior in the data. Another approach that uses a hybrid model for adaptive load prediction is presented in [9]. It uses computationally expensive machine learning tools like support vector machine (SVM) and self organizing maps, and requires supervised learning. Due to its complexity, it does prediction on hourly intervals. Our approach is an online and real-time light weight predictor.

Recent literature discusses attacks on AGC [7], [8], [11], [15], however, no AGC specific intrusion detection approach exists. The work [15] presents integrity attacks on AGC by a knowledged attacker. The parameters are manipulated in an acceptable range thus attack is not detected by embedded data verification modules and causes imbalance in power generation. Cyber attack's impact assessment on a two area power system is presented in [7]. The work also shows how an attacker can cause undesirable behavior under the conditions established using reachability methods by interrupting the AGC signals. The work is followed by [8] where it is assumed that the attacker has partial information about the system parameters to launch an attack. Protective measures to minimize potential risks were developed using game theoretic framework in [11]. Our work infers the attack model from [15].

Intrusion detection techniques for smart grid have been proposed recently [2], [20]. While these techniques address cyber security challenges in Smart Grid, none of these techniques is designed specifically for AGC. A distributed intrusion detection technique in a multi-layer architecture of smart grid is proposed in [19]. In [2], we presented an intrusion detection approach for advanced metering infrastructure. In [16], an anomaly detection technique for the smart grid has been proposed. It focuses on the cyber security of substations and does not cater for AGC parameters. In [5] an approach for anomaly detection in SCADA system has been proposed. It considers the SCADA measurements that are used for state estimation in control center. However, attacks that target control center such as AGC that takes state estimation as an input, will not be detected. Since AGC is the last mile to raise or lower the generation, we work with variables related to AGC.

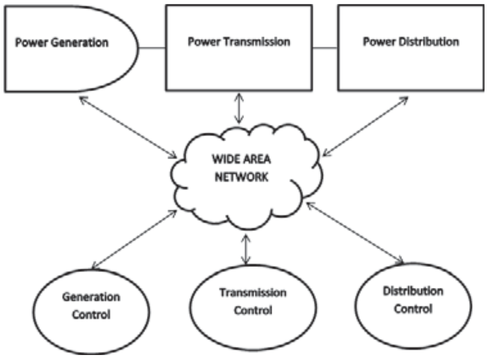


Fig. 1. Schematic diagram of Smart Grid

III. Background & Motivation

Different communication networks are connected to the power grid for sensing measurements and sending control commands. These networks are associated with the supervisory control and data acquisition (SCADA) system for its real-time operation. SCADA system connects the generating stations, substations, corporate offices and control center. SCADA is mainly responsible for monitoring and obtaining data from remote equipment; and for controlling the equipment remotely either by the operator or automatically based on the data acquisition. Though the connectivity has several advantages, it uses the readily available communication infrastructure from Internet Service Provider that makes the system inherently vulnerable to cyber threats.

A basic schematic diagram of smart grid is shown in Figure 1. It can be noticed that all the three control components i.e., generation, transmission and distribution are connected to the power infrastructure using cyber infrastructure. Control centers take the sensor measurements and other data from the power network in order to analyze and send the control commands using the same infrastructure. This operation is part of the SCADA schematic. Substations have the power generation ability for their area and it communicates with the generation control center to adjust the power generation to a required load.

Figure 2 shows a high level diagram of generation control. AGC takes measurements from the field that also goes to state estimation. However, there are some measurements which are AGC specific and calculated based on different field measurements. The main functionality of AGC is to calculate ACE based on the tie line flow (P_{tie}) and frequency (f) of the tie line. In our case, a tie line is connected to two areas for the inter-area power flow.

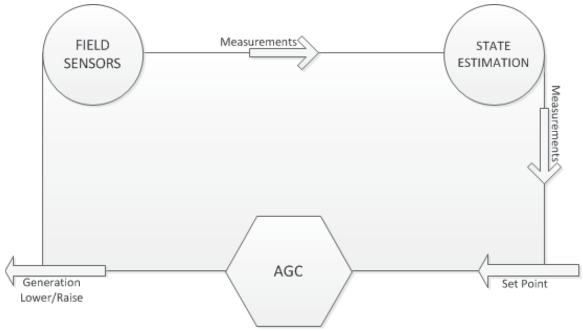


Fig. 2. Automatic Generation Control Loop

$$P_{ACE}^{(t)} \quad = \quad (f^{(t)} - f_{ref}).\beta + (\sum_{i\in\tau} P_{itie} - P_{sch}) \qquad (1)$$

where τ, β, f_{ref}, P_{itie} and P_{sch} represent all tie lines, frequency bias factor, base reference frequency, tie line flow for line i and the scheduled tie line flow, respectively. The first part of Equation 1 considers the system's frequency deviation from standard frequency, and second part considers the deviation of power flow in each tie line from the schedule flows. Thus, ACE is an error between the scheduled and actual values in the system, which is transmitted to each local generation control in order to adjust the generation accordingly.

Once the composite area-wide error is calculated, it is used along-with the economic dispatch (such as demand-based pricing or other) to define the generation schedule. It is done at intervals of 1 to 15 minutes. In case of continuously changing generation demand, allocation of generation must be made instantly. This task is performed through allocation control logic by means of base points and participation factor algorithms. After calculations, each control action i.e., raise or lower pulse signal, is transmitted to the generation unit at remote generation station for changing the generating unit's load reference point.

$$P_{isch}^{(t)} = P_{ibase}(k) + pf_i \times \Delta P_{total}^{(t)} \qquad (2)$$

$$\Delta P_{total}^{(t)} = P_{actual}^{(t)} - \sum_{i\in\xi} P_{ibase}(k) \qquad (3)$$

where ξ represents all the generators, P_{ibase} is the base point set by unit commitment, pf_i is participation factor and P_{actual} is actual generation demand. These variables such as P_{ibase} are generation control specific and are input to the system as shown in Figure 2, denoted by 'set point'. The output of the overall AGC system is the raise/lower generation command. Therefore, attacks on these variables or the variables that constitute them such as load, after the state estimation can not be detected by intrusion detectors working at state estimation or SCADA field measurements.

Generators are generally modeled by swing equation (Equation 4). The equation considers the physical rotor angle and speed as states of the system. Primary controllers, Governor, Automatic Voltage Regulator (AVR) and Power System Stabilizer (PSS) are designed to make these states in their equilibrium values by controlling mechanical and electrical torques.

$$\Delta P_{mech} - \Delta P_{elec} = M\frac{d\Delta\omega}{dt} \qquad (4)$$

where M is the angular momentum of the machine. The generator calculates the difference in change in electrical and mechanical power being generated. These balancing equations (Generator and Scheduling) can be modeled using model-based estimation approaches such as Kalman Filter based approach [18]. However, the estimation model does not take into account the temporal dependence in the data. We propose a hybrid approach in which the first tier is a light-weight online data-driven predictor that leverages the temporal dependence in the data. The second tier builds the system model by incorporating the system-wide knowledge using AGC system variables. This makes the approach practical since the online prediction module (first tier) is light weight and flags any deviation in the data. The output of first tier is given to the second tier offline verification module which utilizes the system-wide knowledge to verify the presence of anomaly. Thus, the proposed approach benefits from temporal dependence in the data along-with the system model.

TABLE I. SAMPLE LOAD AND ACE VALUES

	t_0	t_1	t_2	t_3	t_4
ACE	0.059	0.069	0.080	0.091	0.103
Load	0.950	0.952	0.955	0.957	0.959

Since ACE raises/lower the generation, we change the frequency f which is used to calculate the ACE in order to mimic the attack on ACE. It can be simulated by unbalancing the Equation 4 since that is used to model the generators. To mimic the attack on load we manipulate the active and reactive power by changing R and L of the power system. Since load is used as an input to define the base point and scheduled flow subsequently (Eq. 2), and tie-line flow and frequency deviation reflects in ACE, we use these variables for prediction as first tier of our approach. In the second tier, we model the system using system variables including scheduled flow, tie line and frequency along-with load and ACE to determine the AGC state for anomaly verification. Probability of each system variable is verified against all the system variables to verify the anomaly presence. We further extend it to multi-AGC scenario by conditionally comparing the states of one AGC given the state of interconnected AGC in order to mimic a scenario where two different power companies share the power resources.

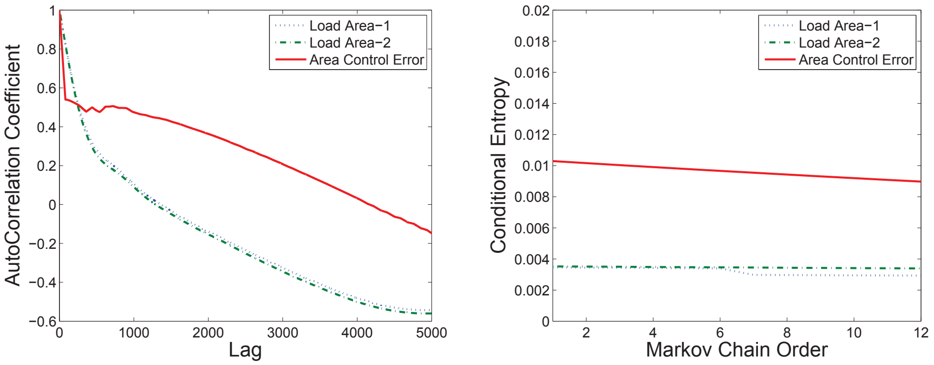
IV. ANALYSIS AND ADAPTIVE PREDICTION

In this section we discuss the analysis of load and ACE data followed by the adaptive prediction algorithm.

A. *Analysis*

To reveal the load and ACE behavior, we conduct statistical analysis on the 24 hours data. The data was generated using two-area Kundur power system model [10] which is well known and widely used in the power community [7], [8], [11]. Sample load and ACE data is shown in Table I, where t_x represents the value realized at time instance x. It can be intuitively argued that, as long as the load and ACE values are produced by the benign events, the values observed should exhibit a certain level of temporal dependence. In case of an anomalous behavior, perturbations in this dependence can be flagged as anomalies. Therefore, the level of temporal dependence can serve as an important metric for the modeling.

Autocorrelation measures the on-average temporal dependence between the random variables in a stochastic process at



(a) AutoCorrelation

(b) Conditional Entropy

Fig. 3. AutoCorrelation and Conditional Entropy of the Load for both the areas

different points in time. For a given lag k, data values window shift, the autocorrelation function of a stochastic process X_n (where n is the time index) is defined as:

$$\rho[k] = \frac{E\{X_0 X_k\} - E\{X_0\}E\{X_k\}}{\sigma_{X_0}\sigma_{X_k}}, \tag{5}$$

where $E\{.\}$ represents the expectation operation and σ_{X_k} is the standard deviation of the random variable at time lag k. In our case, load and ACE values are the realization of the random variable X. For example, X_0 represents the ACE or load value at time 0 as shown in Table I. The value of the autocorrelation function lies in the range $[-1, 1]$, where $\rho[k] = 1$ means perfect correlation at lag k (which is obviously true for $k = 0$) and $\rho[k] = 0$ means no correlation at lag k.

Figure 3(a) shows the autocorrelation function plotted against the load and ACE values at different lags. For both the load and ACE, a certain level of temporal dependence can be easily observed at small lags. This correlation decays in time and eventually drops down to a negligible value. Temporal dependence is present because load and ACE values are similar in a short time interval. It decreases when the behavior of the load and ACE is changed and it increases when similar values are observed.

It is well-known that a decaying temporal dependence structure can be accurately modeled using Markov chains [12]. Therefore, to identify the Markov chain order, we conduct analysis on different Markov chain orders. We define a Markov chain based stochastic model as follows: Let the load and ACE value at discrete time instance n represents the realization of a random variable derived from a stochastic process X_n. This process is a Markov chain if it satisfies the Markov property i.e., the probability of choosing a next state is only dependent on the current state.

Each unique realization of X_n for ACE and load is assigned to a unique bin among multiple non-overlapping bins. Therefore, the number of bins will be dependent on the number of unique values. Each bin then represents a state of the Markov chain, while the set of all bin indices ψ is its state space. Based on this state representation, we can define a 1-st order Markov chain, $X_n^{(1)}$, in which each bin represents a state of the random process. Probability of each state i can be calculated by counting the number of times state i occurred and dividing it by the total occurrences of all the states in the Markov chain model X_n. Similarly, an l-th order Markov chain, $X_n^{(l)}$, can be defined in which each state is an l-tuple $\langle i_0, i_1, \ldots, i_{l-1}\rangle$ representing the values taken by the random process in the last l time instances. In this case the occurrences of l-load values will be counted. This will increase the size of state space ψ since different combinations of l-tuple can be observed.

Conditional entropy, $H(B|A)$, of two discrete random variables A and B characterizes the information remaining in B when A is already known. If A and B are highly correlated, most of the information about B is communicated through A and $H(B|A)$ is small. On the other hand, if A and B are quite different then $H(B|A)$ assumes a high value, which means that most of the information about B is not given by A.

The transition probability matrix of the 1-st order Markov chain $P^{(1)}$ can be computed by counting the number of times the state i is followed by state j. The resulting $|\psi^{(1)}|$ histograms can be normalized to obtain the state-wise transition probability mass functions as the rows of $P^{(1)}$.

We can find the conditional probability of the 1-st order Markov chain as:

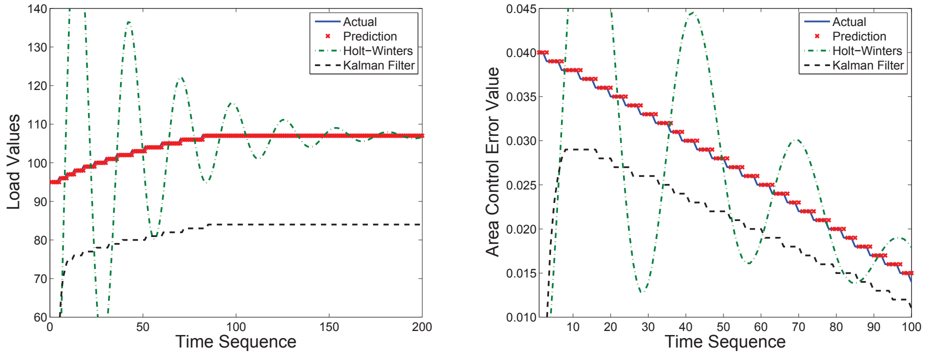
$$H^{(1)} = -\sum_{i\in\psi^{(1)}} \pi_i^{(1)} \sum_{j\in\psi^{(1)}} p_{j|i}^{(1)} \log_2\left(p_{j|i}^{(1)}\right), \tag{6}$$

where $\pi_i^{(1)}$ is the average probability of being in state i which is computed by counting the total number of times each state is visited and then normalizing the frequency histogram.

It can be clearly seen in Figure 3(b) that load and ACE values exhibit a very low conditional entropy at all the orders. There is a slight decaying trend in conditional entropy, if observed closely, however, the decay is not significant enough. Moreover, conditional entropy at all the Markov chain orders is very small for both the parameters i.e., < 0.012. Therefore, prediction can be done using the first order Markov chain.

B. Prediction Algorithm

Based on the analysis and results in the previous section, we now propose a simple predictor based on the Markov chain. This prediction algorithm is in essence a variant of the adaptive thresholding algorithm proposed in [3]. The proposed algorithm along-with the prediction is also capable of adapting to the underlying variation observed in the load and ACE behavior. Below we discuss the algorithm and its prediction accuracy under normal conditions i.e., no attack.



(a) Load Prediction

(b) ACE Prediction

Fig. 4. Prediction Accuracy of the algorithm

The data was generated using well known two-area power system as discussed earlier. Let us subdivide the data values into ψ equal sized bins. Since multiple values will fall into each bin, thus the prediction will give us the range of expected values for the future time instance. The number of total bins ψ can then be calculated by dividing the minimum and maximum allowed value by the size of each bin. Since allowed values are defined by the capacity of the network, we do not expect to see a smaller value than minimum or greater value than maximum, thus prediction is not needed for those values.

Since the conditional entropy was very low for the load and ACE values at all the Markov chain orders, we decide to use the first order Markov chain in order to keep the complexity of the algorithm low. Let $P^{(t)}$ denote the transition probability matrix at time t, where $P^{(t)}_{i,j}$ denote the $i-th$ row and $j-th$ column of the matrix. Also, l^t_p and l^t_o denote the predicted and observed values, respectively, at time t. The working of the algorithm is shown in Algorithm 1. The input to the algorithm is the observed value at the current time instance t and the output is the predicted value for the next time instance $t+1$. The algorithm first calculates the error in the prediction and the observed value for time instance t. If the difference is greater than a particular threshold, we update the transition probability matrix, in order to be adaptive, by giving a higher weight (β) to the transition from $l^{(t-1)}_o$ to $l^{(t)}_o$. However, if the difference is not greater than a threshold α, we update the transition probability matrix by a regular weight ω, where $\omega < \beta$ and α, β and ω are tunable parameters. Once the transition probability matrix is updated, it is used to make a prediction for the next time instance $t+1$ based on the current value l^t_o. Since each row is a pmf for a given state, the column with the highest probability i.e., the bin representing the range of values is selected as the predicted range of values for the next time instance $t+1$. Since it is an adaptive online prediction algorithm, it adapts and gives higher weight to learn the prediction error greater than the threshold α. This makes the algorithm learn the anomalous behavior as well on the run-time. However, it keeps flagging those instances until the behavior is completely learnt and has higher probability than observing any other state. This helps in reducing the false alarms in case the deviation was caused due to a legitimate change in load behavior.

The prediction accuracy of the proposed predictor for load and ACE values is shown in Figure 4(a) and (b), respectively. We show the prediction accuracy comparison with a well-known Kalman Filter based power system load predictor [18] and general purpose predictor Holt-Winters [17]. Although Kalman Filter based predictor was originally proposed for load prediction only, we predict the ACE values too since our detection approach requires monitoring both the data feeds. We do not compare with the hybrid predictor [9] since it does prediction on hourly basis only, however, we are predicting in an online manner for each data value. The solid blue line denotes the actual values observed while red cross, dotted green and black dashed line denotes the prediction using Algorithm 1, Holt-Winters and Kalman Filter, respectively. It can be clearly seen that even in the constant load increase and ACE decrease trend, prediction algorithm was highly accurate and outperformed Kalman Filter and Holt-Winters. Moreover, the prediction followed the complete trend until the values were stabilized. We show a subset of data for clarity, however, similar results were obtained for the entire 24 hour data. Therefore, this prediction algorithm can be a good measure in order to predict the short term future load and ACE values for the purpose of anomaly detection.

Algorithm 1: Load Prediction Algorithm

```
Data: Observed values $l$
Result: Prediction for the value
$\varepsilon = |l^t_p - l^t_o|$ ;
if $\varepsilon > \alpha$ then
    $P^{(t+1)}_{l^{t-1}_o, l^t_o} = \beta \times P^{(t)}_{l^{t-1}_o, l^t_o}$ ;
end
else
    $P^{(t+1)}_{l^{t-1}_o, l^t_o} = \omega \times P^{(t)}_{l^{t-1}_o, l^t_o}$ ;
end
$l^{(t+1)}_p = P^{(t+1)}_{l^t_o, max(l_k)}$ ; where $l_k \in \psi$
```

V. Anomaly Verification in AGC

Prediction serves as the first tier in identifying anomalies. Since prediction is done for short intervals and considers only the temporal dependence in the single variable without the complete knowledge of the system, it may yield false positives. To this end, as a second tier, we build a model for the AGC system. AGC can be seen as a control system taking finite inputs and generating finite outputs. The main inputs are P_{tie}, f and l, where l is for all the areas participating in an AGC. Similarly, the outputs or calculations it does are ACE

and P_{sch}, where P_{sch} is for all the participating generators in an AGC. We consider two different cases: 1) single AGC and 2) multiple AGC for two area power system. Single AGC is the environment where a power company focuses on its own AGC without any collaboration with any other power company's AGC. However, multi-AGC scenario considers multiple power companies sharing power resources for power generation and control.

A. Single AGC

Since AGC has finite set of input and output variables, the state of a single AGC can be encoded using the following characteristic function:

$$\sigma : P_{tie} \wedge f \wedge l \wedge ACE \wedge P_{sch} \rightarrow \{true, false\} \tag{7}$$

The function σ encodes the state of the AGC by evaluating to *true* whenever the parameters used as input to the function correspond to the values observed in the system. If the AGC observes x unique combinations, then exact x assignments to σ function will evaluate to $true$. We use these assignments to learn the markov model for the AGC. Since conditional entropy does not show an exponential decay on higher order markov chains, in analysis, we use the first order.

A Labeled Markov Chain (LMC) is a quintuple $M = \{Q, \Sigma, \pi, \tau, L\}$, where Q is a finite set of states, π is an initial probability distribution τ is the transition probability function and L is a labeling function. Atomic propositions AP are assigned to states by a labeling function using $\Sigma = 2^{AP}$. Each state is assigned a unique label derived from σ i.e., s, which is used to define the state. A probability distribution for sequence of states can then be defined using Markov chain. Suppose we have sequence $S = s_1, s_2, \ldots, s_n,\ s_i \in Q$. A finite state machine having directed graph can be learned from the given sequence S.

To learn the Markov model, we initialize an empty graph and then starts observing the sequence S from AGC. It utilizes a sliding window approach where window slides at instance i by one entry i.e., s. If s_i already exists in graph then a directed edge from s_{i-1} to s_i is created, if the directed edge does not exist already. However, if s_i does not exist in graph, then a node is also created for s_i. This process keeps repeating until S is empty. Once the state machine is created, the transition probability matrix is calculated using the frequency of transitions observed while building finite state machine.

Since the proposed model is based on Markov chain and exhibits a temporal dependence, we define properties in Linear time Temporal Logic (LTL) [4]. Unlike traditional model checking, stochastic model checking allows you to check that with what probability the property is satisfied by the model. These probabilities can be thresholded in order to accommodate the unseen behavior up to a certain extent. The probabilistic LTL can be defined as:

$$\phi ::= P_{\bowtie p}(\varphi), \quad \bowtie \ \in \ \{\geq, >, \leq, <, =\}; p \in [0, 1]$$

where φ is an LTL formula. Since the states are defined using measurement/calculation variables, properties can be written in the form of conditional probability. For example, given the AGC is in a state having some values for the variables under consideration, what is the probability of seeing the current value of any variable? That will determine the state transition at discreet time interval.

$$\phi ::= P_{\geq min \& \leq max}(ACE_{i+1} | P_{tie_i}, f_i, l_i, ACE_i, P_{sch_i}); \quad i \in \Upsilon; \tag{8}$$

where Υ is the sequence of measurements in time domain. The above property checks if the probability of current ACE value observed, given the system was in a particular state, is less-or-equal- and greater-or-equal-than the minimum and maximum probability thresholds, respectively. These probability thresholds are learnt from the data collected at AGC under normal conditions. This identifies whether the system behaves as expected or not. Moreover, it also identifies that which particular variable is causing the anomalous behavior. The thresholds are derived from models built under different operating conditions, like different load levels and hours of day. We define the properties for all the variables, mentioned in Equation 7, in the similar fashion.

B. Multiple AGC

One feature of the smart grid is its large-scale distributed generation networks. In interconnected AGCs, power imbalance in one area can be caused by the attack on the interconnected AGC. To this end, we extend the single AGC framework to the case of multiple AGCs, which reflects the scenarios where multiple power companies cooperate for power generation and control. We assume that participating organizations communicate state variable data. The state for each AGC can be calculated in the similar fashion as described earlier.

$$\sigma_{agc_j} : P_{tie} \wedge f \wedge l \wedge ACE \wedge P_{sch} \rightarrow \{true, false\} \tag{9}$$

where σ_{agc_j} represents the characteristic function for AGC j. Note that P_{tie} and f are the same at a given time for both the AGCs. The common state information is shared between two AGCs that are connected to a tie-line. Both the participating AGCs will share the characteristic function output with each other in order to define the overall state of the entire system i.e., two AGCs. The shared information allows each AGC to check its own state variables and detect local anomalies. The model will be built using the state transitions for both the AGCs. Consider the sequence of states is:

$$S = (\sigma^{t_0}_{agc_1}, \sigma^{t_0}_{agc_2}), (\sigma^{t_1}_{agc_1}, \sigma^{t_1}_{agc_2}), \ldots, (\sigma^{t_n}_{agc_1}, \sigma^{t_n}_{agc_2}) \tag{10}$$

where $\sigma^{t_k}_{agc_j}$ represents the state of AGC j at time k. Thus, the finite state graph can be learnt the same way as described previously. However, in this case the state function comes from both AGCs instead of one AGC. Similarly, intrusion can be identified by checking the probability of one AGC characteristic function given the other AGC's characteristic function instead of individual variables, in order to incorporate the knowledge of both the AGCs i.e., entire system. Thus, it can be written in conditional probability form for the case of two connected AGC as an example.

$$\phi_1 ::= P_{\geq min \& \leq max}(\sigma^{t+1}_{agc_1} | \sigma^{t}_{agc_2}); \tag{11}$$

where the minimum and maximum probabilities are learnt from the model built for both the AGCs. Similarly we calculate the probability of AGC 2 state given the state of AGC 1.

$$\phi_2 ::= P_{\geq min \& \leq max}(\sigma^{t+1}_{agc_2} | \sigma^{t}_{agc_1}); \tag{12}$$

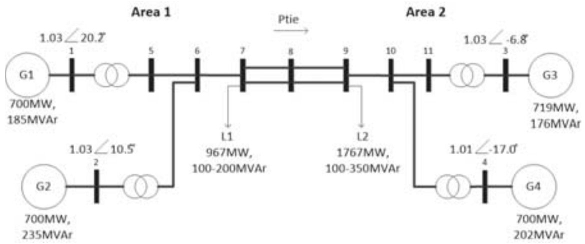


Fig. 5. Two-area power system

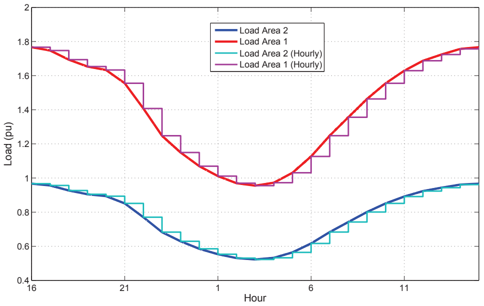


Fig. 6. Load Profile

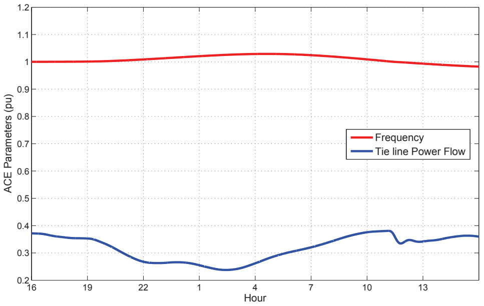


Fig. 7. ACE Parameters

The iterative process yielded by Equation 11 and 12 provides a distributed and scalable detection for large scale AGC networks. At each time t, two AGCs exchange their state variables. In order to verify anomaly in an interconnected AGC scenario, we verify the state of one AGC given the state of the other connected AGC using the above equations. This verifies the overall behavior of both the AGCs together by incorporating the system-wide knowledge. This helps in identifying whether both the AGCs behave normal as learnt. It also helps in identifying that which AGC is deviated away from the normal behavior since the property for that AGC would not be validated with respect to the other AGC. This process can be applied to any pair of two connected AGCs.

VI. Experimentation and Evaluation

In this section we discuss the experimentation, attack models and evaluation.

A. Experiment

In our work, we adopt a well-known two-area Kundur's power system model [10]. Figure 5 shows one-line diagram of the model. This system has been modeled in a power simulation software, PSCAD, which is widely used by the power system professionals [14]. The system consists of 2 areas, 11 buses, 4 generation units and 2 tie lines. All generation units are frequency dependent and the balance is initially restored locally due to the load that varies with frequency. Generator governors change generator output in response to the frequency changes. The two tie lines are identical in this case study, which can be considered as one equivalent tie line connecting area 1 to area 2. In this simulation, loads are modeled in PSCAD as varying resistance and inductance. Load profile data is derived from September 1, 2010 at NYISO's LONGIL [6]. In this case, the hourly demand curve for NYISO's LONGIL is scaled and used at all the load buses of the test system with time intervals of 5 sec to meet the SCADA data transfer rate as shown in Figure 6. In addition, frequency and tie line power flow (ACE parameters), with respect to time of the day at normal condition, are provided in Figure 7. It can be observed that a power of $0.4p.u.$ flows from area 1 to area 2 under normal conditions (legitimate change in behavior without attack) at maximum load demand, which is hour 16. This high load demand also brings down the systems frequency from $1p.u.$ to $0.99p.u.$ According to our assumption, all the generators respond to change in load, and their participation factor is related to their capacity of generation. The value of β, which is frequency deviation coefficient in eq. 1, is set to 1.9. We show experimental results on a 24 hour dataset. However, the load and ACE can be modeled for a longer duration (e.g., weeks) using their respective base values. Prediction algorithm has been implemented in Java which collects the measurements from PSCAD and predict accordingly. If an alarm is raised, it is passed on to the anomaly verification module, which has been implemented in PRISM [13]. PRISM supports building the Markov model and allows probabilistic property verification.

B. Attack Model

Control centers are generally connected to two networks i.e., corporate and control network. Corporate and control network are separated using added layer of security generally through firewall. Corporate network is also connected to the internet through a security layer using firewalls. Therefore, for an attacker to reach control network, he/she first has to reach corporate network by bypassing first layer of security. Then the attacker has to gain access to the control network in order to manipulate AGC. Another entry point for an attacker could be a WiFi network in the control center. The intrusions addressed by the work are manipulation of AGC parameters for power imbalance as shown in [15]. We introduce two different types of integrity attacks for two different parameters related to AGC and generation allocation logic. All the parameters used in this work affect the control logic unit. We reiterate that the attack point is the control center, after the state estimation in the control flow as shown in Figure 2. These attacks will not be detected by SCADA specific intrusion detection systems focusing on SCADA data or state estimation.

1) Attack on Load: First, we change the load of the system by injecting a false load which satisfies the generation control equations. By manipulating R and L of the loads, we can control the active and reactive power load. The false load directly impacts the generation allocation control to generate the power as per the new load observed, since base point is calculated on the load. However, it indirectly affects the AGC parameters as well i.e., P_{tie} and f. Three cases have been studied in this attack model.

- Case1: $0.15p.u.$ change in total system's load

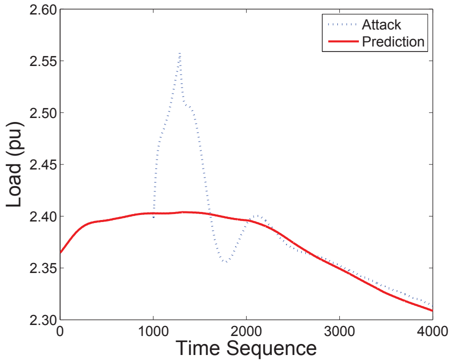


Fig. 8. Attack on Load

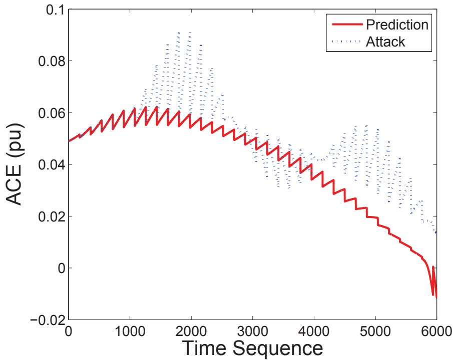


Fig. 9. Attack on ACE Parameter, Frequency

TABLE II. MEASUREMENTS AT NORMAL AND ATTACK SCENARIOS

Parameter	Normal	Attack Case1	Attack Case2	Attack Case3
P_{load1}	0.844	0.919	0.944	0.844
P_{load2}	1.558	1.633	1.558	1.658
f	0.9902	0.9881	0.9865	0.9867
P_{tie}	0.3578	0.3513	0.3413	0.3849
ACE area 1	0.0492	0.0387	0.0257	0.0696

- Case2: $0.1p.u.$ change in area 1's load
- Case3: $0.1p.u.$ change in area 2's load

As it can be seen, in Figure 8, the load value is increased by $0.15p.u.$ from the normal condition at time $16hr$, which has the max load and minimum frequency. $Psch.tie$ is set to $0.29p.u.$ derived from economic dispatch. Reference for f is $1p.u.$ Detailed normal and attack scenario measurements are provided in Table II. Moreover, to mimic a sophisticated attacker who has consistent access to the data, we introduce instances of stealthy attacks such that $0.1p.u.$ change commutatively in 100 time instances.

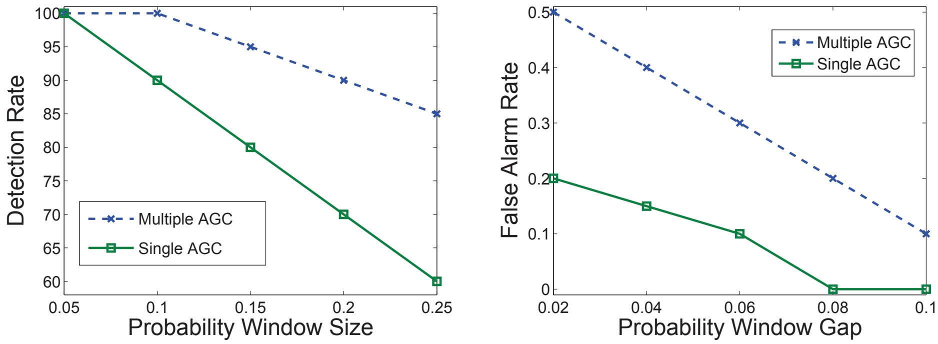
2) Attack on ACE: In the second case study, we change frequency f which affects the value of ACE. This task is simulated by unbalancing the swing equation of generator 3. A sudden $0.2p.u.$ mechanical power change is simulated at 16 hr, which directly causes a frequency deviation from its reference (Eq. 4). This gives a false estimate of the frequency f of generator 3 and total system, thus resulting in the disruption of ACE, as shown in Figure 9. It can be observed that the error has been increased, thus destabilizing the system since AGC adjusts according to the error for power generation. We also introduce the stealthy attack instances as described earlier for load. Please note that all the values changed still satisfies the AGC equation.

C. Accuracy Evaluation

We evaluate accuracy for both the single and multiple AGC scenario. Since prediction is done individually for AGC system variables, it is the same in both the case studies. Multiple attack instances were introduced as mentioned earlier and data was collected using PSCAD tool. Prediction algorithm was employed on the data which included attack instances for both the load and ACE values. It can be clearly seen in Figure 8 and 9 that the prediction follows the trend learnt from the normal data and do not deviate with the sporadic changes observed in the load and ACE values. Perturbations in the ACE values as a result of attack are clear, which ultimately causes the destabilization of the power system. All the attack instances were successfully flagged by the prediction algorithm i.e., deviated away from the prediction, however, one percent false alarm was observed in a 24 hour dataset. These flagged instances were then passed to the anomaly verification module.

Anomaly verification module is evaluated in two fashions i.e., single and multiple AGC. Single AGC is a centralized (or composed) verification of two-area power system as a whole, and the multiple AGC is the distributed verification of the two-area system. In case of single AGC, all the attack instances were successfully detected i.e., 100% detection rate with no false alarm. Similarly, for multiple AGC, we also observed 100% detection rate but with a false alarm rate of 0.1%. Since the properties are probabilistically verified using the model, we change the probability verification threshold and show its effect on the accuracy for both single and multiple AGC in Figure 10. Specifically, we change the minimum and maximum verification threshold, thus increasing or decreasing the window gap between probability verification thresholds. It can be explained by the fact that if the window gap is 1 (maximum gap possible), i.e., the minimum threshold is 0 and the maximum is 1, the property will always be validated successfully. Therefore, it will not detect any intrusions and hence there is no false alarm. Similar trend can be observed in Figure 10(a) that as the probability window gap increases, detection rate decreases. It can be noticed that multi-AGC scenario observed higher detection rate, as compared to single AGC scenario, for the same probability verification window gap. The underlying reason is that multi-AGC case is more sensitive to intrusions since it checks its local system state given the system state of the other AGC. Intuitively, since it involves all the system variables (in conjunction form) from both the AGCs, all the system variables are verified against all the other variables. Thus, it is a strict property to satisfy and can be thought of as a strict threshold for verification and it is well known that strict thresholds yield higher detection and false alarm. Hence, it is more sensitive to even slight deviation from the normal behavior. On the other hand, in single AGC case, the probability of each system variable is checked against only the entire system state of that particular AGC, thus it is not as sensitive as the multi-AGC scenario. Consequently, multi-AGC observes a higher false alarm rate as shown in Figure 10(b). Although the multi-AGC false alarm rate is higher than single AGC, it is still negligible and overall higher accuracy was observed in all the cases.

Though multi-AGC scenario provides higher detection accuracy, it is not an obvious choice. Multiple AGCs requires sharing the system state data among cooperating AGCs which



(a) Detection Rate

(b) False Alarm Rate

Fig. 10. Comparison of Single AGC and Multiple AGC scenario

may be owned by different companies. Thus, it may raise technical and non-technical issues. Moreover, delay in sharing the system state data may cause a wrong estimate of the system state, hence it requires reliable and efficient communication. Thus, sharing the system state data introduces extra overhead on the system. In such cases, single AGC scenario is a better suited option. However, it is not as sensitive to attacks as multiple AGC. Therefore, the choice depends on the trade-offs.

VII. Conclusion

The paper presents a two-tier intrusion detection system for AGC. The first tier is an online short-term adaptive predictor for load and ACE variables independently, which are the key input and output variables in AGC. We show that both, load and ACE parameters, exhibit a temporal dependence which can be modeled in order to make future short-term predictions accurately. The second tier provides offline probabilistic model checking of the overall system by incorporating the system-wide knowledge. Markov models have been used to represent the system state. Two case studies were conducted i.e., single- and multi-AGC. Single AGC verifies the behavior of each system variable with respect to the system-state. On the other hand, multi-AGC verifies the state of one AGC given the state of other AGC to verify anomaly. Anomaly verification reduces the overall false alarm rate by incorporating the system-wide knowledge. Since measurements are collected every few seconds, model checking at run-time is not computationally feasible. Therefore, the first tier does online prediction for individual variables, as a result the flagged instances are then passed to an offline anomaly verification module which incorporates the complete knowledge of the system in order to verify the anomaly presence. The prediction algorithm exhibits high prediction accuracy ($> 95\%$) under normal conditions. Multiple attack scenarios, inspired from [15], have been implemented. All the malicious measurements have been found to deviate from predicted values, thus being successfully detected. One false alarm was observed by the prediction on 24 hour data. Second tier successfully verified all the anomalies present with a negligible false alarm rate i.e., 0% for single AGC and 0.1% for multi-AGC, thus increasing the overall accuracy of the approach.

References

[1] Cybersecurity and the north american electric grid: New policy approaches to address an evolving threat. *Bipartisan Policy Center*, 2014.

[2] M. Q. Ali and E. Al-Shaer. Configuration-based IDS for advanced metering infrastructure. In *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[3] M. Q. Ali, E. Al-Shaer, H. Khan, and S. A. Khayam. Automated anomaly detector adaptation using adaptive threshold tuning. *ACM Transactions on Information and System Security (TISSEC)*, 2013.

[4] C. Baier and J. P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[5] J. Bigham, D. Gamez, and N. Lu. Safeguarding scada systems with anomaly detection. In *Computer Network Security*, 2003.

[6] M. Doostizadeh and H. Ghasemi. Day-ahead scheduling of an active distribution network considering energy and reserve markets. *European Transactions on Electrical Power*, 2012.

[7] P. M. Esfahani, M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson. Cyber attack in a two-area power system: Impact identification using reachability. In *American Control Conference (ACC), 2010*, pages 962–967. IEEE, 2010.

[8] 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 *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5973–5978. IEEE, 2010.

[9] S. Fan and L. Chen. Short-term load forecasting based on an adaptive hybrid method. *Power Systems, IEEE Transactions on*, 21(1):392–401, 2006.

[10] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.

[11] Y. W. Law, T. Alpcan, M. Palaniswami, and S. Dey. Security games and risk minimization for automatic generation control in smart grid. In *Decision and Game Theory for Security*, pages 281–295. Springer, 2012.

[12] M. Merhav, M. Gutman, and J. Ziv. On the estimation of the order of a markov chain and universal data compression. *IEEE Transactions on Information Theory*, 1989.

[13] Probabilistic symbolic model checker PRISM. http://www.prismmodelchecker.org/.

[14] Power system simulation tool. https://hvdc.ca/pscad/.

[15] S. Sridhar and G. Manimaran. Data integrity attacks and their impacts on scada control system. In *Power and Energy Society General Meeting, 2010 IEEE*, 2010.

[16] C. Ten, J. Hong, and C. Liu. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, 2011.

[17] H. L. V. Trees. *Detection, estimation and modulation theory: part I.* Wiley-Interscience, 2001.

[18] D. J. Trudnowski, W. L. McReynolds, and J. M. Johnson. Real-time very short-term load prediction for power-system automatic generation control. *Control Systems Technology, IEEE Transactions on*, 9(2):254–260, 2001.

[19] Y. Zhang, L. Wang, W. Sun, R. Green, and M. Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Transactions on Smart Grid*, 2011.

[20] B. Zhu and S. Sastry. SCADA-specific intrusion detection/prevention systems: A survey and taxonomy. In *First Workshop on Secure Control Systems (SCS)*, 2010.