Deep Learning based Covert Attack Identification for Industrial Control Systems

Dan Li*, Paritosh Ramanan*, Nagi Gebraeel*, and Kamran Paynabar*

*School of Industrial and Systems Engineering

Georgia Institute of Technology Atlanta, USA

{dli352, paritoshpr}@gatech.edu

{nagi.gebraeel, kamran.paynabar}@isye.gatech.edu

Abstract—Cybersecurity of Industrial Control Systems (ICS) is drawing significant concerns as data communication increasingly leverages wireless networks. A lot of data-driven methods were developed for detecting cyberattacks, but few are focused on distinguishing them from equipment faults. In this paper, we develop a data-driven framework that can be used to detect, diagnose, and localize a type of cyberattack called covert attacks on smart grids. The framework has a hybrid design that combines an autoencoder, a recurrent neural network (RNN) with a Long-Short-Term-Memory (LSTM) layer, and a Deep Neural Network (DNN). This data-driven framework considers the temporal behavior of a generic physical system that extracts features from the time series of the sensor measurements that can be used for detecting covert attacks, distinguishing them from equipment faults, as well as localize the attack/fault. We evaluate the performance of the proposed method through a realistic simulation study on the IEEE 14-bus model as a typical example of ICS. We compare the performance of the proposed method with the traditional model-based method to show its applicability and efficacy.

Index Terms—cybersecurity, industrial control system, smart grid, LSTM

I. Introduction

Industrial control systems are collections of complex networks of interactive control and automation systems that are widely used for control and monitoring in chemical, energy, and manufacturing industries [1]. With the development of sensor technology in the past few decades, modern industrial control systems are generating a massive amount of data in real-time, leveraging the system operations and monitoring to wireless data communication. Although these technologies bring novel insights to the optimization of system operations, they also raise concerns about cybersecurity [2], [3]. Unlike the common information systems, industrial control systems belong to cyber-physical systems in which the cyber layer is interactive with the physical layer. The computation and the data transformation on the cyber layer enforces the operation and management of the physical layer, which in turn generates data that is fed back to the cyber layer. Because of these operational control interactions, malicious attacks that impact data integrity can cause physical damage to the system and thus be particularly dangerous. For example, the outbreak of the malware *Stuxnet* [4] in 2010 draws public attention to data integrity attacks. The event showed such cyberattacks can truly happen and cause severe damage to the critical infrastructures.

Therefore, data integrity becomes a very important factor for cyber-physical systems [5].

Data integrity attacks often take over the programmable logic controllers (PLC) and smart sensors, and manipulate sensor measurements and/or control actions, while bypassing the traditional detection schemes [6]. For example, *false data injections* [7] refer to the attacks where the attacker only manipulates the sensor measurements to mislead the state estimation and further alter the control action. *Replay attacks* refer to the attacks where the attacker manipulates the control action and replays the sensor measurements recorded from normal operations to disguise the malicious control. In this paper, we focus on a more sophisticated type of integrity attack called *covert attack* [8], where the attacker manipulates the control action and disguises the malicious behavior by replacing the sensor measurements with the simulated data based on the original control input. The underlying assumption includes the following: 1) The attacker has access to manipulate the control actions and sensor measurements; 2) The attacker possesses knowledge of the system behavior so that the simulated data could represent the normal behavior of the system and can be used to mask the malicious manipulation. The covert attack has been proved to be able to be undetectable when the attacker has access to all the sensors and full knowledge of the system [8].

Aside from the difficulty in detecting these integrity attacks like covert attacks, there are two challenges in cyberattack identification for ICSs. The first one is the false alarms triggered by natural equipment faults [9]. Although various methods can successfully detect the cyberattack, few of them consider the possibility that a natural equipment fault could trigger false alarms. These false alarms can bring the unnecessary cost of inspections and even system shutdowns. The other challenge is the localization [10]. Industrial control systems are networks of multiple subsystems, which serve as the nodes in the network. These nodes may operate individually but are physically connected. Typical examples of ICS are found in power networks, where a control center may be monitoring multiple substations and power generation plants that are geographically distributed, while being physically connected by the transmission lines. Once a cyberattack is detected by the control center, the next step is to determine the location of the attack. Because of the complex interconnectivity among

these subsystems, the localization of the attack is nontrivial. For example, a covert attack might impact multiple nodes such that more than one will raise up alarms for detecting an attack. Therefore, a diagnosis method needs to be developed to localize the attack.

In this paper, we demonstrate the application of deep learning towards the detection, diagnosis, and localization of covert attacks. We focus on generic networked industrial control systems and propose a data-driven framework that combines an autoencoder, an RNN, and a DNN. We use the RNN to characterize the system behavior under normal operations. The output of the RNN together with the sensor measurements are fed to a DNN classifier to detect, diagnose, and localize the anomaly. We use the autoencoder to extract features that represent the system status, as well as the spatial correlation among the nodes, in an unsupervised manner. The RNN captures the temporal behavior of the features extracted by the autoencoder, and the DNN helps detect anomalies in the system as well as diagnose whether it is an attack or fault. By considering both the spatial and temporal behavior of the system, this DL framework helps reduce false alarms triggered by natural faults as well as localize the attack by extracting the features that distinguish anomalies at different locations and between attack and faults.

II. RELATED WORK

The literature on cyberattack detection for industrial control systems can be divided into two groups: model-based detection and data-driven detection.

Model-based methods rely on the engineering knowledge of the physical rules to establish a parameterized model of the normal sensor measurements [7], [11]–[14]. In [11], [12], and [13], the observed sensor measurements are compared with the estimated sensor measurements from these models, and the residuals (i.e., the difference between the observations and model estimations) are monitored to detect the anomalies. The models are often established such that the expectation of the residuals is approximately 0 under normal operations. During monitoring, a large residual means a large discrepancy between the model estimation and the observations, which indicates anomalous behavior of the system. The χ^2 detector, for example, tests the sum of squared residuals (SSR) and triggers alarms whenever the SSR is above the threshold defined by a χ^2 distribution. The drawback of the model-based methods is the complexity in establishing an accurate physical model, especially when the system consists of multiple subsystems with complex connections and correlations, both spatially and temporally. For complex systems such as power systems, the system model is usually represented by the steady-state power flow equations, which takes the observations as independent inputs and does not consider the system dynamics over time.

Compared to model-based methods, data-driven methods are more flexible. A number of applications of machine learning techniques have been developed for detecting cyberattacks. However, most of them focus on designing intrusion detection systems based on the network traffic data [15]–[17].

If such intrusion detection systems fail, there is a lack of backup detection scheme that detects cyberattacks based on the physical process. On the other hand, unlike the behavior of network traffic, the physical process has a relatively stable and consistent dynamic behavior that does not change over time. Therefore, a recurrent neural network can be used to characterize the system behavior in a non-parametric manner, especially when the complexity of the system obstructs the establishment of a physics-based model. To take into consideration network traffic data and the sensor data, in [18], the authors proposed a multi-model data fusion and adaptive deep learning method based on a convolutional neural network to characterize the normal system behavior. The framework then detects cyberattacks as well as physical intrusions in a single ICS pertaining to water infrastructure. However, due to the complexity of the proposed method, it would not be applicable to a network of infrastructures, where a utility operates multiple equipment (subsystems) with interactions at the same time. Meanwhile, the establishment of such model requires thorough understanding of the system, and it is hard to generalize it to other systems.

III. SYSTEM MODEL

We consider a networked control system consisting of k subsystems as shown in Fig. 1, where the subsystems are not necessarily homogeneous, meaning there is not a single model that can represent the behaviors of all the subsystems. For example, a smart grid is a network of generation buses, representing the power plants, and load buses, representing the substations, and different models should be used to represent the power plants and the substations. We denote the state of the subsystem i at time t by a vector x_t^i, where $x_t^i \in R^{n_i}$, and n_i is the minimum number of variables needed to uniquely define the system state of site i. Denote the history of state of subsystem i till time t as $x_{\{t\}}^i = \{x_0^i, x_1^i, ..., x_t^i\}$. The full system state vector at time t, x_t, is given by a concatenation of all the subsystem states. i.e, $x_t^T = [x^{1T}, ..., x^{kT}] \in R^N$ where $N = n_1 + n_2 + ... + n_k$. Note that x is not directly observed, but inferred from the measurements from the M ($M > N$ for system observability) sensors distributed throughout the system. Similarly, we have $y_t^T = [y^{1T}, ..., y^{kT}] \in R^M$ where $M = m_1 + m_2 + ... + m_k$, and m_i is the number of sensors in subsystem i. Denote the control action as $u_t^T = [u^{1T}, ..., u^{kT}] \in R^P$, where $u_t^i \in R^{p_i}$ and $P = p_1 + ... + p_k$. Similarly, we denote the history of control actions and measurements till time t as $u_{\{t\}} = \{u_0, u_1, ..., u_t\}$ and $y_{\{t\}} = \{y_0, y_1, ..., y_t\}$, respectively.

In general, the dynamics of the networked system can be represented by

$$x_t = \mathcal{F}(x_{\{t-1\}}, u_{\{t-1\}}), \tag{1}$$
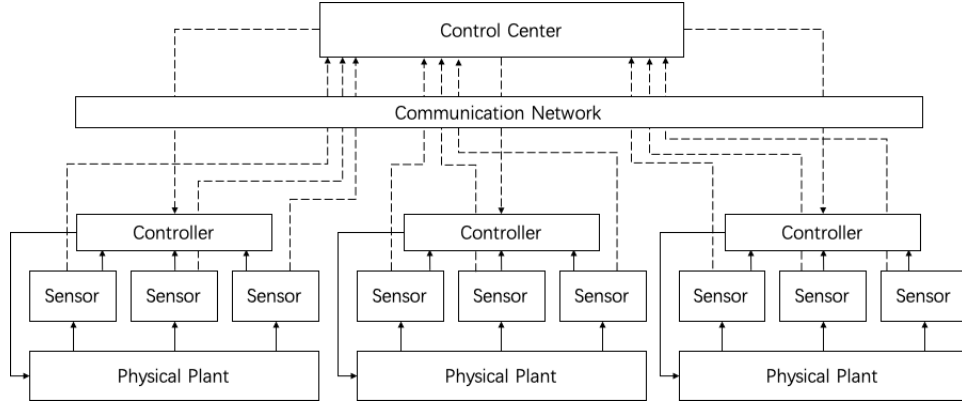$$y_t = \mathcal{G}_i(x_t), \tag{2}$$

Fig. 1. Networked Industrial Control System

and a single subsystem i can be represented by a recursive function of x_t^i as follows:

$$x_t^i = f_i(x_{\{t-1\}}, u_{\{t-1\}}) \tag{3}$$
$$y_t^i = g_i(x_t) \tag{4}$$

Systems with linear dynamics where the Markovian property holds, can be represented by a linear state-space model. For an individual subsystem i which operates independently, we have:

$$x_t^i = A_i x_{t-1}^i + B_i u_{t-1}^i \tag{5}$$
$$y_t^i = C_i x_t^i \tag{6}$$

When the subsystems are correlated with each other, the full system can be modeled by

$$x_t = A x_{t-1} + B u_{t-1}, \tag{7}$$
$$y_t = C x_t, \tag{8}$$

where A, B, and C are functions of A_i's, B_i's, and C_i's. In model-based methods, typically the above model is used to build a state estimator (e.g., a Kalman filter) that estimates $\hat{x}_t$ based on the observed measurement y_t and the previous estimation $\hat{x}_{t-1}$. The estimated $\hat{x}$ is then substituted back into the system model to predict the next measurement $\hat{y}_{t+1}$, which is compared with the observed y_{t+1} to obtain the residuals:

$$r_t = y_t - \hat{y}_t.$$

When accurately parameterized, the above model is robust and works well for simple systems. However, one disadvantage of these physics-based models is the estimation of all the parameters requires a large amount of data, and might cause the identifiability issues. Therefore, We assume the interconnectivity of these systems are complex to represent using a physics-based model, this is true especially when the system is nonlinear and the subsystems are interconnected.

Another disadvantage of the physics-based models is that sometimes they do not consider the dynamic behavior of the system or the temporal correlation in the data. In most systems, the control actions are calculated based on a period of historical data, or on a state estimation from the previous time step. In this case, the temporal correlation is implicit, and a steady-state estimation does not capture this type of correlation. For example, in a power system, the state vector x contains the control actions u_t as well.

We note that the control actions can be represented by the power generation setpoints. These setpoints can be determined by solving an operational planning problem subject to the system demand profile. Therefore, in this paper, we utilize the well known Mixed Integer Unit Commitment (MIUC) [19] as a means to compute our operational setpoints. The MIUC is widely used in the power industry for operational planning and can be formulated as follows (9):

$$\min_{\alpha, \theta} \quad c^\top \alpha + d^\top \theta \tag{9a}$$
$$\text{subject to} \quad Q\theta + R\alpha = E \tag{9b}$$
$$F\theta = H \tag{9c}$$

In Problem (9) α represents a binary vector of length $|\mathcal{G}| \times T$ indicating if generators are turned on or turned off across each time epoch for each generator in the network, where $|\mathcal{G}|$ is the number of generators. Similarly, θ represents a real-valued vector of dispatch variables specifying the level of production on generators as well as the electric phase angles on separate buses of length $(|\mathcal{B}| + |\mathcal{G}|) \times T$, where $|\mathcal{B}|$ is the number of buses.

Constraint (9b) ensures that the commitment and production decisions are coupled along with ramp up (Q) and ramp down (R) constraints found in unit commitment. Constraint (9c) enforces flow constraints (F) subject to the phase angle values as well as the transmission line capacities (H).

As mentioned earlier, the implicit temporal correlation generated by calculating the control actions in a history-dependent manner cannot be easily represented by any physics-based model. On the other hand, the occurrence of an attack might be easily captured by analysing the temporal behavior of the system, while the steady state does not show any anomaly. Therefore, data driven method, specifically RNN, well fits this situation where temporal correlation needs to be captured via data-driven techniques.

A. Covert Attack Model

The covert cyberattack was first proposed in [8] for a single linear system as represented by functions 5 and 6. Specifically, the attacker knows the values of B_i and C_i. Recall that a covert attacker is assumed to possess access to control action and the sensor measurements as well. Under these assumptions, the attacker implements the covert attack by the following steps:

- First, the attacker manipulates the control actions using the following equation

$$\tilde{u}_t^i = u_t^i + a_t, \tag{10}$$

where $\tilde{u}_t^i$ is the manipulated control action at time t, a_t is the attack signal added to the original control action u_t^i. According to 5, this manipulation will alter the system state at $t+1$ by $B_i a_t$. That is,

$$\tilde{x}_{t+1}^i = x_t^i + B_i a_t \tag{11}$$

The consequent sensor measurements will be biased by $C_i B_i a_t$. That is,

$$\tilde{y}_{t+1}^i = y_{t+1}^i + C_i B_i a_t. \tag{12}$$

- Then, the attacker manipulates the sensor measurements by subtracting the above bias. i.e.,

$$\breve{y}_{t+1}^i = \tilde{y}_{t+1}^i - \gamma, \tag{13}$$

where $\gamma = C_i B_i a_t$. In this way, the manipulated measurement $\breve{y}_{t+1}^i$ is equal to the expected measurement y_{t+1}^i without attack. Hence, the covert attack can be successfully disguised, as the measurement is the only output of the system, which means all the data inference is conducted based on y.

In this work, we generalize the covert attack to nonlinear systems represented by Equations (3) and (4). We assume the attacker gains knowledge of the dynamics of subsystem i. This could be taken as the attacker obtains an estimation of the local functions $\hat{f}_i$ and $\hat{g}_i$ in Equations (3-4), which can serve as the simulator of system i. With this knowledge as well as the access to the control actions and sensor measurements, the attacker conducts a covert attack using the following steps:

- First, the attacker reads the original control action u_t and simulates the expected sensor measurements using the knowledge of subsystem i. That is,

$$\hat{x}_{t+1}^i = \hat{f}_i(x_{\{t\}}, u_{\{t\}}^i), \tag{14}$$

$$\hat{y}_{t+1}^i = \hat{g}_i(\hat{x}_{t+1}^i). \tag{15}$$

- Then, the attacker manipulates the control actions as in Eq. (10). According to Eq. (3), this manipulation will alter the system state at $t+1$ as well as the sensor measurements. That is,

$$\tilde{x}_{t+1}^i = f_i(x_{\{t\}}, \tilde{u}_{\{t\}}^i,) \tag{16}$$

$$\tilde{y}_{t+1}^i = g_i(\tilde{x}_{t+1}^i). \tag{17}$$

- Finally, the attacker replaces the consequent sensor measurements with the simulated one. i.e.,

$$\breve{y}_{t+1}^i \leftarrow \hat{y}_{t+1}^i. \tag{18}$$

Notice that we assume the attacker's access to the sensors is limited to subsystem i. This means the attacker is not capable of compensating for the impact of attacking subsystem i on other subsystems. This fact lays the foundation for our detection and localization framework. However, detecting the attack in this case is nontrivial because the sensors that are most informative of the attack are manipulated, while the attack's impact on other sensors does not have a clear indication of the occurrence of an attack as well as its location.
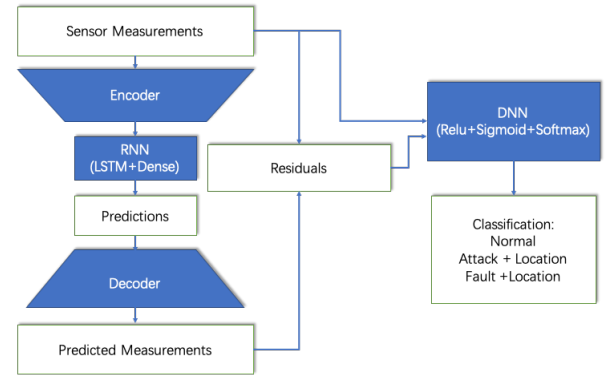
IV. PROPOSED DETECTION FRAMEWORK

Fig. 2. The proposed detection framework (AEN + RNN + DNN)

The structure of the proposed framework is shown in Fig.2. The autoencoder is used for unsupervised feature extraction. Essentially, it projects the sensor measurements to a lower dimensional layer that filters out the noise and better represent the system status, which corresponds the steady-state state estimation in traditional frameworks. We use an RNN with an LSTM [20] layer to capture the nonlinear temporal dependency and the correlation among all the variables and extract the residuals. The RNN is used as a predictor, which corresponds to the particle filters or Kalman filters in the traditional frameworks. The residuals together with the original sensor measurements are fed into a DNN for detection as well as diagnosis, and the DNN corresponds to the detection and diagnosis schemes in the traditional frameworks.

Since the measurement $y \in R^m$ is obtained by the measurement function $\mathcal{G}(\cdot)$, which is a mapping from x in the lower dimensional space R^n. The correlation among the sensor measurements is actually defined by the correlation among the state variables. Therefore, we use an autoencoder (AEN) to reduce the dimension of the sensor measurements, and for generality, the code size (output size of the encoder) is chosen as the dimension of the state vector, n. In our experiments, the autoencoder (and decoder) consists of 2 dense hidden layers with leaky ReLU activation functions. The encoded sensor measurements is taken as the input to the RNN (in our

experiments, consisting of an LSTM layer followed by a dense layer). The output of the RNN is then decoded to reconstruct the predicted sensor measurements. The residuals are calculated as the difference between the observed measurements and the predicted ones. Then, we concatenate the residuals with the observed measurements, and the concatenated data are fed into a DNN. The DNN used in our experiments contains 3 dense hidden layers with ReLU, Sigmoid, and Softmax activation functions in sequence. The output of the DNN determines the type and location of the attack, designated by "normal", "attack+location", and "fault+location".

By the nature of covert attacks, the attacker has to obtain accurate system knowledge to conduct a successful attack. However, in reality, the complex subsystems are often geographically far apart from each other that makes it very unlikely that an attacker can obtain the knowledge necessary for attack, and get access to more than one subsystems. Therefore, in this work, we assume that the attacker only attacks one node, and it is less likely that faults on different nodes happen to occur at the same time. Hence, there are at most $2k + 1$ independent possible conditions of the system, including "normal", {"attack on node i, $i = 1, ..., k$"}, and {"fault on node i, $i = 1, ..., k$"}.

The autoencoder and the LSTM are trained using only the data from normal operations. The DNN can be trained in a supervised manner using labeled data from simulation or historical data.
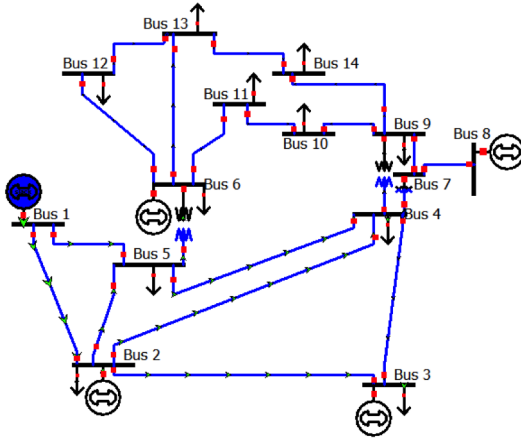
V. PERFORMANCE EVALUATION

A. Data Extraction

Fig. 3. The IEEE 14-Bus System[1]

We generate data via a simulation study on a smart grid. We use the IEEE 14-bus (Fig. 3) to represent the smart grid at the transmission level to generate the time series of sensor measurements under different conditions. The model contains 9 load buses (buses 4, 5, 7, and 9-14), representing 9 substations; 4 generation buses (buses 2, 3, 6, and 8),

representing 4 power generation plants; and 1 slack bus (bus 1), which is used to balance the active and reactive power in the system and also serves as a reference for all other buses. The model has 20 edges, representing the 20 transmission lines connecting the load and generator buses. The input to the simulation is the load profiles of all the load buses and the power generation plans of all the generation buses. We obtain the load profile of each substation by aggregation the load profiles of a random number of hourly residential power consumption profiles extracted from Pecan Street [21]. The generation plan is constructed using the method mentioned in Section III. The simulation in Matlab uses Matpower [22] 7.0 to solve the power flow equation and add measurement noise. The output of the simulation is the hourly time series of the 39 sensor measurements for the simulated period. The 39 sensor measurements include the active power flow on each of the 20 transmission lines, the power generation of each generator bus as well as the slack bus, and the voltage of all the 14 buses.

Recall that the mechanism of a covert attack is to alter the system state by manipulating the control actions. Since in the transmission system we are considering in this simulation, most of the control happens in the power generation plants, we only consider the covert attacks on the generator buses. During the attack, the attacker decreases the generation level by a specific portion. The reason we choose to decrease the generation is from the attacker's objective: compared to generating more power than needed, decreasing the generation will cause possible blackouts and overloading of other generators, which is likely to cause more damage to the system. We simulate the covert attacks on each of the 4 generator buses at 5 levels of severity, where the attacker decreases the power generation by level 1: 20%, level 2: 40%, level 3: 60%, level 4: 80%, and level 5: 100% of the planned generation. We assume the attacker obtains access to all the sensors related to the attacked generator and manipulates the sensor measurements by replacing the original values with the ones obtained from simulation such that it shows the attacked generator bus generates the same amount power as planned. For comparison, we simulate the faults as the decrease of power generation by the same amount caused by equipment malfunctioning. The biggest difference between a fault and a covert attack is there is no sensor data manipulation.

B. Numerical Results

We compare our proposed method with two benchmarks. The first one uses the traditional state estimation (SE) residuals for detection (Fig. 4). The state estimation is implemented by solving the (nonlinear) static power flow equations using Newton's method [23], which is implemented in Matpower 7.0. The second method uses the same RNN model as in the proposed model, but does not include the autoencoder (Fig. 5).

For the proposed model and the "RNN + DNN" model mentioned above, we first calculate the first order difference of observations to ensure the stationarity, and then standardize the differences based on the standard deviation for each sensor. In
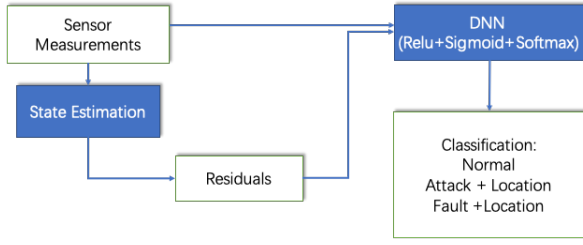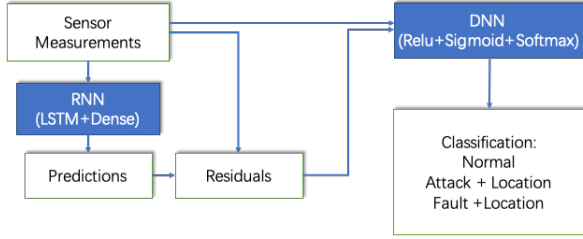
Fig. 4. The SE + DNN Structure

Fig. 5. The RNN + DNN Structure

this case, we use $n = 14$ as the code size of the autoencoder. Usually for an AC k-bus system the state space has dimension $2(k-1)$, consisting of the magnitudes and phase angles of voltages at each bus. Since here we only care about the active power and do not consider the phase angle, we use $k-1 = 13$ as the code size. The stateful LSTM uses 10 lags to predict the next encoded observation. We train the autoencoder and the RNN with 80% of the normal data, and keep them fixed when generating the residuals for attack and fault series. Therefore, the autoencoder and the RNN are only exposed to the normal data, which can be used to represent the system behavior under normal operations. The autoencoder and the RNN can be viewed as a non-parametric substitute of the physics-based models.

Since there are 39 sensors, the concatenation of residual and observation data is of length 78. Since there are 4 generator buses considered, the output of DNN is a multi-class classification of 9 labels listed in Tab I. The numbers refer to the bus number where the attack/fault occurs.

We train the model with 80% of data and show the classification performance of the DNN by testing the model on the rest 20% of data. The performance of the method is evaluated by precision, recall, and the F1-score, which are shown in Tab.I and Tab.II, where Tab.I shows the classification performance of DNN among the 9 labels (normal, 4 attacks, and 4 faults), and Tab.II shows the classification among the 3 classes (normal, attack, and fault) as well as the localization performance within the attack and fault classes. The precision, recall, and F_1-score are calculated based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) using the following equations:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Tab.I and Tab.II show that for most of the cases, the proposed method outperforms the other two methods in terms of detection as well as localization. In general, the SE+DNN model has similar but slightly better performance than the RNN+DNN model. However, both of them has a very low recall for attack on bus #2. This is because in the network topology, bus #2 has a high degree – it has four neighbors, which is the highest among the four generator buses. According to the proposed generalization of covert attack in Section III, the attacker manipulates all the sensors measuring the power flow on the four edges. Intuitively, in this case the attacker has the highest coverage of sensor accesses, which best covers the attack. Since the state estimation only refers to the data at current step and does not consider temporal correlation, it does not give a good estimation of the underlying truth. On the other hand, since the RNN is taking all the sensor measurements as input, the prediction of RNN is relatively inaccurate due to the noise in the sensor data. However, since the RNN considers the temporal correlation, the performance of RNN + DNN in this case is slightly better than SE + DNN. In contrast, the proposed method uses an autoencoder for unsupervised feature extraction, which helps filter out the noise in the data when training the RNN. In this case, the RNN gives more precise prediction of the data. Therefore, the residuals in this case could better capture the anomaly caused by the attack.

We also show the F_1 scores of the three methods under different levels of attacks and faults in Fig.6. In general, the proposed method has a better performance than the other two, especially for attacks on bus #2. Another inspection is that the F_1 score increases as the level (severity) of the attack increases. This is a validation that the covertness (the ability to stay undetected) of the attack decreases as the the severity of attack increases, meaning the distinction between normal data and the data under attack becomes clearer as the severity of attack increases, leading to a higher detection power and diagnosis accuracy. Moreover, it can be seen that the performance of all three methods generally depends on the connectivity of the attacked node – Attacks on bus #2 and #6 has lower detection rates because they have more neighbors.

To evaluate the performance of the proposed method when encountering attacks that are not in training, we use a subset of the attack levels to train the DNN and test it on the other levels. We test the method by selecting l ($l = 1, 2, 3, 4$) levels among the five levels simulated. For each l, we replicate the training and testing 20 times. Within each replication, the attack levels for training is randomly selected. The boxplot of testing accuracy is shown in Fig. 7. The result shows that the proposed method has a relatively high accuracy when the model is trained on more than 2 levels of attacks. Meanwhile, as more attack levels are used in training the model, the testing accuracy increases, and the variance of the accuracy decreases. When the testing levels are out of the range of the training

TABLE I
PRECISION, RECALL, AND F SCORE FOR DNN CLASSIFICATION

Model	State Estimation + DNN			RNN + DNN			Autoencoder + RNN + DNN		
Label	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Normal	0.7330	0.9699	0.8350	0.7114	0.9803	0.8245	0.9980	0.9284	0.9619
Attack #2	0.8761	0.1943	0.3180	0.8361	0.2203	0.3487	0.9998	0.9813	0.9904
Attack #3	0.9985	0.9983	0.9984	0.9403	0.9921	0.9655	0.9773	0.9998	0.9884
Attack #6	0.9998	0.9920	0.9959	0.9867	0.8146	0.8924	1.0000	0.8640	0.9271
Attack #8	0.8645	0.9771	0.9173	0.9762	0.9046	0.9391	0.8918	0.9869	0.9369
Fault #2	0.9999	1.0000	1.0000	0.9947	0.9995	0.9971	0.9999	0.9987	0.9993
Fault #3	0.9925	0.9994	0.9959	0.9962	0.9774	0.9867	0.9997	0.9964	0.9981
Fault #6	0.9308	0.9806	0.9551	0.8971	0.8792	0.8881	0.7320	0.9977	0.8444
Fault #8	0.9572	0.9005	0.9280	0.9287	0.8930	0.9105	0.9729	0.8781	0.9231

TABLE II
PRECISION, RECALL, AND F SCORE FOR DETECTION AND LOCALIZATION

Model		State Estimation + DNN			RNN + DNN			Autoencoder + RNN + DNN		
Label		Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Normal		0.7330	0.9699	0.8350	0.7114	0.9803	0.8245	0.9980	0.9284	0.9619
Attack		0.9778	0.7199	0.8292	0.9512	0.6817	0.7942	0.9900	0.9667	0.9782
Fault		0.9776	0.9885	0.9830	0.9683	0.9565	0.9624	0.9180	0.9896	0.9524
Attack	#2	0.9962	0.9468	0.9709	0.9685	0.9914	0.9798	0.9999	0.9815	0.9906
	#3	0.9999	1.0000	1.0000	0.9913	1.0000	0.9956	0.9819	0.9999	0.9908
	#6	0.9999	1.0000	1.0000	0.9991	0.9869	0.9929	1.0000	1.0000	1.0000
	#8	0.9460	0.9961	0.9704	0.9951	0.9565	0.9754	1.0000	1.0000	1.0000
Fault	#2	0.9999	1.0000	1.0000	0.9998	0.9999	0.9998	1.0000	1.0000	1.0000
	#3	1.0000	0.9999	1.0000	0.9996	0.9998	0.9997	1.0000	0.9999	1.0000
	#6	1.0000	1.0000	1.0000	0.9998	0.9992	0.9995	1.0000	1.0000	1.0000
	#8	1.0000	1.0000	1.0000	0.9973	0.9980	0.9977	0.9997	1.0000	0.9998

levels, the performance is worse (e.g., the outliers for $l = 4$ are corresponding to the replications where the model is trained with levels 2-5 and tested on level 1). This is because if the data is trained on a stronger attack, the data from a weaker attack would lie in between the clusters of normal data and the attack data, and hence the DNN would have difficulty classifying these data.

VI. CONCLUSION

In this paper, we proposed a generic data-driven framework for detecting, diagnosing, and localizing covert attacks on industrial control systems. The proposed framework uses an autoencoder for unsupervised feature extraction from sensor measurements, and then uses an RNN to capture the temporal correlations among the encoded sensor measurements. The prediction of the RNN is decoded and compared with the input sensor measurements to get the residuals, which help detect the anomaly. The residuals and the sensor measurements are processed with a DNN to determine whether an observation is representing normal conditions, an attack, or a fault, and to identify the location of the attack/fault. The proposed framework was compared with the model-based state estimation technique, as well as a modification of itself by removing the autoencoder. The results showed that the autoencoder helps extract the important features from the data, as well as reduce the dimension of the input to the RNN. This significantly

helps improve the classification accuracy of the DNN. It is worth noticing that the RNN does not provide a more accurate estimation of the state compared to the model-based state estimation. However, since the RNN considers the temporal behavior of the system, which is not considered by the mode-based SE, the residuals obtained from the decoded RNN prediction could better capture the anomalous characteristics of the data when the system is under the attacks/faults. The reason model-based SE does not perform well under covert attack is because the objective of SE is to minimize the residuals. This leverages the estimation of the state to normal conditions, which does not represent the underlying truth, especially when the attacker has access to more sensors. The simulation study and performance evaluation validated the proposed method. Since this method is model-free, it is easily generalizable to other networked industrial control systems.

In this work, we only trained and tested the model on the known attack/fault types. A future direction is to extend the method to anomaly-based detection, which can detect novel attacks and faults. Another direction is to combine the method with graphical network topology, as well as the correlation structure of the data.

REFERENCES

[1] K. Stouffer, J. Falco, K. Scarfone, Guide to industrial control systems (ics) security, NIST special publication 800 (82) (2011) 16–16.
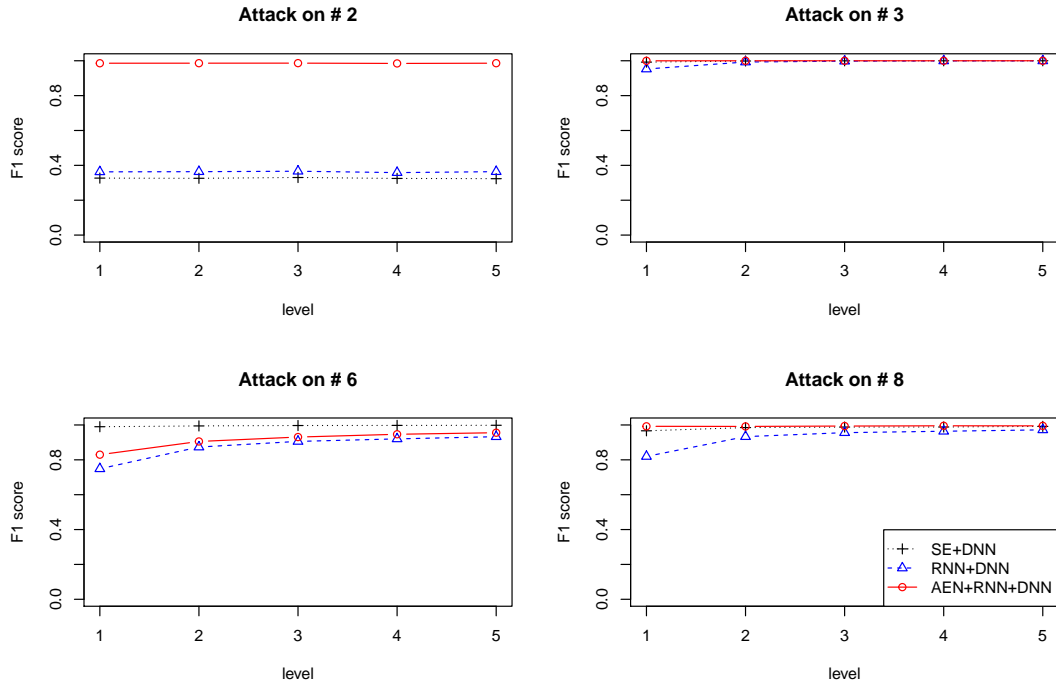
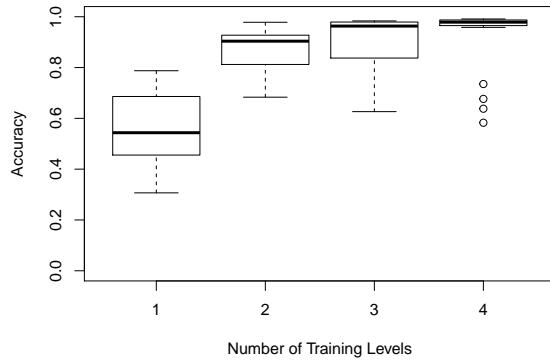Fig. 6. F1 score under different levels of attack for the three models

Fig. 7. The boxplots of accuracy under different number of training levels

[2] Z. DeSmit, A. E. Elhabashy, L. J. Wells, J. A. Camelio, An approach to cyber-physical vulnerability assessment for intelligent manufacturing systems, Journal of Manufacturing Systems 43 (2017) 339–351.

[3] S. Sridhar, A. Hahn, M. Govindarasu, Cyber–physical system security for the electric power grid, Proceedings of the IEEE 100 (1) (2011) 210–224.

[4] D. Kushner, The real story of stuxnet, ieee Spectrum 3 (50) (2013) 48–53.

[5] S. Sridhar, G. Manimaran, Data integrity attacks and their impacts on scada control system, in: IEEE PES general meeting, IEEE, 2010, pp. 1–6.

[6] Y. Mo, R. Chabukswar, B. Sinopoli, Detecting integrity attacks on scada systems, IEEE Transactions on Control Systems Technology 22 (4) (2014) 1396–1407.

[7] Y. Liu, P. Ning, M. K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Transactions on Information and System Security (TISSEC) 14 (1) (2011) 13.

[8] R. S. Smith, A decoupled feedback structure for covertly appropriating networked control systems, IFAC Proceedings Volumes 44 (1) (2011) 90–95.

[9] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications surveys & tutorials 18 (2) (2015) 1153–1176.

[10] T. R. Nudell, S. Nabavi, A. Chakrabortty, A real-time attack localization algorithm for large power system networks using graph-theoretic techniques, IEEE Transactions on Smart Grid 6 (5) (2015) 2551–2559.

[11] D. Van Long, L. FILLATRE, I. NIKIFOROV, Sequential monitoring of scada systems against cyber/physical attacks, IFAC-PapersOnLine 48 (21) (2015) 746–753.

[12] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, S. Sastry, Attacks against process control systems: risk assessment, detection, and response, in: Proceedings of the 6th ACM symposium on information, computer and communications security, ACM, 2011, pp. 355–366.

[13] T. Huang, B. Satchidanandan, P. Kumar, L. Xie, An online detection framework for cyber attacks on automatic generation control, IEEE Transactions on Power Systems 33 (6) (2018) 6816–6827.

[14] S. Li, Y. Yılmaz, X. Wang, Quickest detection of false data injection attack in wide-area smart grids, IEEE Transactions on Smart Grid 6 (6) (2014) 2725–2735.

[15] Y. Yilmaz, S. Uludag, Mitigating iot-based cyberattacks on the smart grid, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2017, pp. 517–522.

[16] A. Terai, S. Abe, S. Kojima, Y. Takano, I. Koshijima, Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile, in: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2017, pp. 132–138.

[17] M. Caselli, E. Zambon, J. Amann, R. Sommer, F. Kargl, Specification mining for intrusion detection in networked control systems., in: USENIX Security Symposium, 2016, pp. 791–806.

[18] N. Bakalos, A. Voulodimos, N. Doulamis, A. Doulamis, A. Ostfeld, E. Salomons, J. Caubet, V. Jimenez, P. Li, Protecting water infrastructure from cyber and physical threats: Using multimodal data fusion and adap-

tive deep learning to monitor critical systems, IEEE Signal Processing Magazine 36 (2) (2019) 36–48.

[19] M. Carrión, J. M. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Transactions on power systems 21 (3) (2006) 1371–1378.

[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[21] P. Street, Dataport: the world's largest energy data resource, Pecan Street Inc.

[22] R. D. Zimmerman, C. E. Murillo-Sánchez, R. J. Thomas, Matpower: Steady-state operations, planning, and analysis tools for power systems research and education, IEEE Transactions on power systems 26 (1) (2010) 12–19.

[23] F. C. Schweppe, J. Wildes, Power system static-state estimation, part i: Exact model, IEEE Transactions on Power Apparatus and systems (1) (1970) 120–125.