Enhanced Cyber-Physical Security through Deep Learning Techniques*

Mayra Macas and Wu Chunming

Department of Computer Science, Zhejiang University, No.38, Zheda Rd, Zhejiang 310000, PR China
{mayramacas11,wuchunming}@zju.edu.cn

Abstract. Nowadays that various aspects of our lives depend on complex cyber-physical systems, automated anomaly detection, as well as attack prevention and reaction have become of paramount importance and directly affect our security and ultimately our quality of life. Recent catastrophic events have demonstrated that manual, human-based management of anomalies in complex systems is not efficient enough, underlying the importance of automatic detection and intelligent response as the recommended approach of defence. We proposed an anomaly detection framework for complex systems based on monitored data storage and Statistical Correlation Analysis for different pairs of constituent time series of a multivariate time series segment, and unsupervised deep learning to intelligently distinguish between normal and abnormal behavior of the system. Experimental results demonstrate that the proposed model is much better than baseline methods, and it can model (inter)correlation and temporal patterns of multivariate time series effectively.

Keywords: Anomaly detection · Critical Infrastructures · Deep Learning.

1 Introduction

Cyber-Physical Systems (CPS) comprise a new generation of sophisticated systems whose normal operation depends on robust communications between their physical and cyber components. Such systems have become vital for several industrial sectors, including water treatment and distribution plants, electrical power grids, public transportation systems, oil refineries, and many more. As the deployment of Internet of Things (IoT) is undergoing an exponential increase, a rise in CPS applications for a large variety of tasks is also observed, resulting in many systems and devices communicating and working autonomously over networks. At the same time, CPS and IoT also increase the likelihood of cyber-security vulnerabilities and incidents, as pointed out in the annual statements issued by the European Agency for Network and Information Security (ENISA) [1] and the Industrial Control Systems Cyber Emergency Response

*M. Macas is supported by Chinese Government Scholarship (15027268)

Team (ICS-CERT) [2], where exploits of the heterogeneous communication systems in charge of managing and controlling complex environments are presented and discussed. From the cybercriminals' perspective, the use of CPS constitutes a unique opportunity to cause maximum damage with minimum effort [3]. In recent years, many attempts to stealthily exploit CPS of important sectors have occurred, such as the attack on the power grid in Ukraine [4], the Maroochy water breach [5], the Stuxnet worm in Iranian nuclear plant [6], the Triton malware on the Saudi oil company [7], and a growing number of attacks on energy networks [3].

Considering that the services provided by such systems are important for the well-being of the community, CPS can be classified as Critical Infrastructures (CI) [3]. Consequently, their flexibility and resilience against cyber-attacks has been a primary concern. Indeed, attacks that could corrupt or disrupt the rendered services would have a negative impact in the context of public safety and order, financial losses and environmental damage. Therefore, the ability to detect sophisticated cyber-attacks on the increasingly heterogeneous nature of the CPS that is amplified by the arrival of IoT has become a crucial task. In this paper, we focus on an unsupervised machine-learning based anomaly detection approach, based on which we attempt to detect anomalous behavior of the system at the physical level.

Popular solutions for anomaly detection such as Statistical Process Control (SPC) [8] methods like cumulative sum (CUSUM), Exponentially Weighted Moving Average (EWMA) and Shewhart charts are not able to cope with the increasingly heterogeneous nature of the CPS with the arrival of IoT. As a result, researchers have moved beyond specification or signature-based techniques and have begun to leverage both supervised and unsupervised machine learning techniques to develop more intelligent and adaptive methods for big data, in order to identify anomalies or intrusions [9]. However, even with the use of machine learning techniques, the detection of anomalies in time series remains a demanding task. First, while the supervised techniques require a sufficient amount of labeled normal data and anomaly classes to learn from, anomalies are typically scarce in a real environment. Second, most of the existing unsupervised methods, such as distance/clustering methods [10] and temporal prediction methods [13], may still not be able to effectively recognize anomalies due to the following reasons: (i) The existence of temporal dependencies in multivariate time series. Distance/clustering methods (e.g., k-Nearest Neighbor (kNN) [10]) and classification methods (e.g., One-Class SVM [11,12]) cannot capture temporal dependencies across different time series. (ii) Multivariate time series data often contain noise in real environment applications. When the noise grows moderately severe, it can affect the generalization ability of temporal prediction models (e.g., LSTM-RRN [13]), causing the false positive detection rate to increase. Fig. 1. illustrates a high-level overview of the proposed anomaly detection framework for complex systems that aims to address the aforementioned challenges. Based on monitored data storage and Statistical Correlation Analysis for different pairs of constituent time series of a multivariate time series segment,

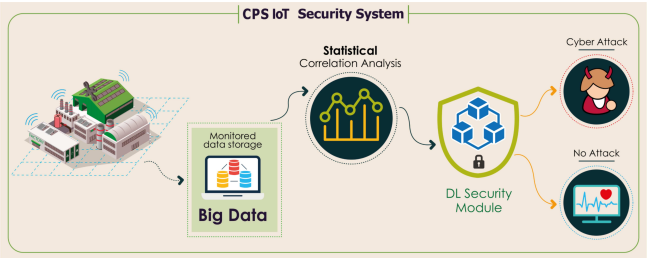

Fig. 1: Anomaly detection representation

we employ unsupervised deep learning to intelligently distinguish between "normal" and "abnormal" behavior of the system. More precisely, we propose an unsupervised deep learning approach based on Spatial-Temporal Encoder-Decoder scheme for Anomaly Detection in a complex multi-process CPS that builds upon the trained Convolutional Neural Network Autoencoder (CNN-AE) and Convolutional LSTM EncoderDecoder (ConvLSTM-ED) models. In greater detail, we first construct correlation matrices to characterize the system status. Next, a convolutional encoder is employed to encode the patterns of the correlation matrices, whereas a ConvLSTM-ED model captures the underlying temporal dependencies. Finally, the convolutional decoder is used to reconstruct the correlation matrices and is leveraged in order to detect anomalies. The central idea is that the model will be trained only with normal data and will learn to accurately reconstruct the respective matrices. When given an anomalous instance, it is not expected to reconstruct it equally well, and this will result to higher reconstruction errors compared to the ones of normal instances. Therefore, these errors can be used to separate normal from abnormal behaviors. Our primary contributions are the following: (i) We design an intelligent system, which is trained to detect anomalies in complex multi-process cyber-physical systems and is built upon Convolutional Neural Network Autoencoder and Convolutional LSTM Encoder-Decoder. (ii) We conduct extensive performance evaluation on the Secure Water Treatment (SWat) testbed [14]. Our preliminary results demonstrate the superior performance of the proposed model compared to state-of-the-art baseline methods.

2 Related works

Unsupervised learning techniques aim to identify the hidden structure of unlabeled data. Given that these techniques can handle a large dataset in addition to their simplicity, they have been extensively employed in the most recent studies on CPS intrusion detection [16]. The SVM-based one-class (OCSVM) classifier [11,12] and k-means clustering algorithms are used in [10]. Nevertheless, distance/clustering methods and the OCSVM classifier ignore the temporal dependencies that exist between anomalous data points [15] and are vulnerable to false alarms. Goh et al. [13] used deep LSTM-RNN and Cumulative Sum

(CUSUM) to detect anomalies on the first stage of the SWAT dataset. However, this approach is not suitable for time series affected by external factors not captured by sensors, making them unpredictable [17]. Inoue et al. [11] conducted a study based on OCSVM. The research was carried out on all six stages of the SWaT dataset. The axuthors used a complex structure that treats sensors and actuators separately, in which the outputs of the LSTM layer are used to predict the outcome of the actuators. The predictions are combined with actual values and are fed into a fully connected hidden layer to predict the mean value and variance of the first sensor. This process is repeated for the remainder of the sensors, and then the sum of the log likelihoods of the actuator positions and sensor values gives the outlier factor used for anomaly detection. The proposed architecture is complex, challenging to understand and resource demanding.

Recently, Kravchik et al. [18] used two deep neural network models: 1D-convolutional (CNN) and recurrent neural network (LSTM), in order to detect cyber-attacks on all six stages of the SWAT dataset. The authors assert that the model with ensembled record reports rates of 86.7%, 85.4% and 86.0% for precision, recall, and F1 score, respectively. Nevertheless, the attack detection was performances at each stage separately, therefore the ways to learn inter-stage dependencies (including time dependencies) were not examined.

3 Secure Water Treatment (SWaT) testbed dataset

The Secure Water Treatment (SWaT) testbed [14] was designed to provide researchers with data collected from a realistic, complex CPS environment. The SWaT testbed is an operational small-scale water treatment plant that supplies purified water. The water purification method in the testbed is divided in six stages denoted P1 through P6. Each stage has a series of sensors and actuators. The P1 stage is for raw water supply and storage, and P2 is for pre-treatment where the water condition is evaluated. Undesired substances are then eliminated by ultra-filtration (UF) backwash in P3. The residual chlorine is eliminated during the Dechlorination process (P4). Subsequently, the water from P4 is pumped into the Reverse Osmosis (RO) system (P5) to decrease inorganic impurities. Finally, P6 stores the water that is suitable for distribution and consumption. The sensors and the actuators at each phase are connected to the corresponding PLC (programming logic controller), and the PLCs are connected to the SCADA (Supervisory Control and Data Acquisitions) workstation.

The data from 51 sensors and actuators were recorded every second by the Historian Server. The SWaT dataset contains seven days of capturing under normal operating conditions and a four-day-long recording during which 36 attacks were carried out. The attack model used in the experiment simulated a system that was already affected by attackers, who proceed to interfere with normal system operation and spoof the system state to the PLCs, thus causing incorrect commands by modifying the network traffic in the level 1 network, raising the sensors' values and issuing fake SCADA commands. The dataset includes attacks that aim at a single stage of the system, as well as attacks targeting

simultaneously multiple stages. Furthermore, similar varieties of sensors (or actuators) tend to react to attacks in similar fashions. The above observations suggest we should assume a multivariate approach during model formulation, instead of considering each sensor or actuator in the CPS as an independent data source (univariate approach). The underlying correlation between the sensors and actuators could be applied to accurately recognize irregularities in the system.

4 Proposed Framework

In this section, we first introduce the problem we aim to study, and then we describe the proposed model in detail.

4.1 Problem statement

Suppose we have the historical data of n time series, i.e., $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n)^\intercal = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L) \in \mathbb{R}^{n \times L}$, where L is the size/length of the time series. Under the assumption that there are no anomalies in the historical data, the model aims to detect anomalous events at certain time steps after L.

4.2 Statistical Correlation Analysis

Following the suggestions of many recent studies [20,21], we apply statistical correlation analysis between different pairs of time series in a multivariate time series segment to characterize the system. In particular, we construct a $n \times n$ correlation matrix based on Pearson's correlation coefficient. Given two time series $\mathbf{x}^i = \left(\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_L^i\right)^\top \in \mathbb{R}^L$ and $\mathbf{x}^j = \left(\mathbf{x}_1^j, \mathbf{x}_2^j, \cdots, \mathbf{x}_L^j\right)^\top \in \mathbb{R}^L$, the Pearson's correlation coefficient can be calculated as follows:

$$m_i^j = \frac{\sum_{g=1}^{L}(\mathbf{x}_g^i - \bar{\mathbf{x}}^i)(\mathbf{x}_g^j - \bar{\mathbf{x}}^j)}{\sqrt{\sum_{g=1}^{L}(\mathbf{x}_g^i - \bar{\mathbf{x}}^i)^2 \sum_{g=1}^{L}(\mathbf{x}_g^j - \bar{\mathbf{x}}^j)^2}} \tag{1}$$

where $\bar{\mathbf{x}}^i$ and $\bar{\mathbf{x}}^j$ represent the sample means of the two time series. In order to investigate the effect of characterizing system status in different scales (or different sequences length) during anomaly detection different lengths of sequences were tested in the experimental phase. Since the SWaT data were recorded every second, we built the correlation matrices with window lengths of $\ell = \{90, 120, 150\}$ (i.e., data collected within 1.5 2 and 2.5 minutes) at each time step. In this study, the interval between the starting time of two consecutive segments is $s = 10$.

4.3 Spatial-Temporal Encoder-Decoder scheme

The Spatial-Temporal Encoder-Decoder (ST-ED) scheme is adapted from the architecture proposed in [19]. The model combines a convolutional autoencoder

[22], which learns the spatial structure of each correlation matrix, with a ConvLSTM Encoder-Decoder [23] that captures the temporal dependencies from the learned spatial feature maps of every time step. Fig. 2. (left) depicts the core of our approach.

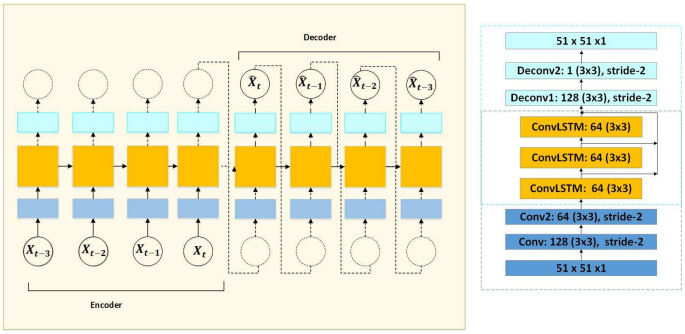

Fig. 2: (Left)the proposed Deep Learning Security Module with Input Subsequences of Length equal four. The blue and light blue denote convolution and deconvolution respectively. The orange boxes represent ConvLSTM. (Rigth) spatial view of our model.

Convolutional LSTM Units: the regular LSTM applies vector multiplications on the input elements. That is, it treats the input as vectors and it vectorizes the input feature map. The statistical correlation matrices, however, are composed of both spatial and temporal components. Given that no spatial information is considered by the LSTM, the results of such an application could be suboptimal. In order to conserve the spatiotemporal information, the fully connected multiplicative operations of the input-to-state and state-to-state transitions are substituted by convolutions in ConvLSTM [23], namely:

$$\mathbf{i}_t = \alpha(\mathbf{W}_{xi} * \boldsymbol{\mathcal{X}}_t + \mathbf{W}_{hi} * \boldsymbol{\mathcal{H}}_{t-1} + \mathbf{W}_{ci} \odot \boldsymbol{\mathcal{C}}_{t-1} + \mathbf{b}_i) \tag{2}$$

$$\mathbf{f}_t = \alpha(\mathbf{W}_{xf} * \boldsymbol{\mathcal{X}}_t + \mathbf{W}_{hf} * \boldsymbol{\mathcal{H}}_{t-1} + \mathbf{W}_{cf} \odot \boldsymbol{\mathcal{C}}_{t-1} + \mathbf{b}_f) \tag{3}$$

$$\boldsymbol{\mathcal{C}}_t = \mathbf{f}_t \odot \boldsymbol{\mathcal{C}}_{t-1} + \mathbf{i}_t \odot tanh(\mathbf{W}_{xc} * \boldsymbol{\mathcal{X}}_t + \mathbf{W}_{hc} * \boldsymbol{\mathcal{H}}_{t-1} + \mathbf{b}_c) \tag{4}$$

$$\mathbf{o}_t = \alpha(\mathbf{W}_{xo} * \boldsymbol{\mathcal{X}}_t + \mathbf{W}_{ho} * \boldsymbol{\mathcal{H}}_{t-1} + \mathbf{W}_{co} \odot \boldsymbol{\mathcal{C}}_{t-1} + \mathbf{b}_o) \tag{5}$$

$$\boldsymbol{\mathcal{H}}_t = \mathbf{o}_t \odot tanh(\boldsymbol{\mathcal{C}}_t) \tag{6}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ represent the input, forget, and output gates at time t respectively; $\boldsymbol{\mathcal{C}}_t$, and $\boldsymbol{\mathcal{H}}_t$ denote the cell outputs and the hidden states at time t; $\alpha()$ and $tanh()$ are the sigmoid and hyperbolic tangent non-linearities; $\odot$ denotes the Hadamard product; * expresses the convolution operation; $\mathbf{W}_{h\bullet}$ are the filter matrices connecting different gates, and $\mathbf{b}_{h\bullet}$ are the corresponding bias of filters. All the inputs $\boldsymbol{\mathcal{X}}_1, \cdots, \boldsymbol{\mathcal{X}}_t$, cell outputs $\boldsymbol{\mathcal{C}}_1, \cdots, \boldsymbol{\mathcal{C}}_t$, hidden state $\boldsymbol{\mathcal{H}}_1, \cdots, \boldsymbol{\mathcal{H}}_t$, and gates $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are $3D$ tensors whose last two dimensions are spatial dimensions

(rows and columns). Furthermore, this convolutional version also adds optimal *peephole* connections that enable the units to derive past information better. The advantages of ConvLSTM over regular LSTM are related to the advantages of convolutional layers compared to linear layers: they are suitable for learning filters, valuable for spatially invariant inputs, and they require less memory for the parameters. The memory needed is independent of the size of the input.

As shown in Fig. 2. (right), our model consists of three ConvLSTM layers and the Convolutional auto-encoder is used to reconstruct the correlation matrices. The employed loss function is the mean square error (MSE) between the prediction result and ground truth for N time steps: $\frac{1}{N}\sum_{t=1}^{N}\left[\mathbf{X}_t - \hat{\mathbf{X}}_t\right]^2$. We use mini-batch stochastic gradient method together with Adaptive Moment Estimation (Adam) method [27] to minimize the mean square loss function. After training the model, the neural network is used to infer the reconstruction correlation matrices of validation and test data. Finally, anomaly detection is performed build upon residual correlation matrices, which is presented in the next section.

5 EXPERIMENTS AND ANALYSIS

In order to evaluate the performance of the proposed framework, we carried out a comprehensive empirical study exploiting the infrastructures and the datasets of the SWaT testbed.

5.1 Experimental Setup

Dataset: The SWaT dataset contains data captured on a per second basis for 51 variables corresponding to sensors and actuators. Within the raw data, 496,800 records were collected under normal conditions, and 449,919 records were collected while performing various cyber-attacks in the system. The first 16,000 records of the training dataset were trimmed since it took around 5 hours to reach stabilization when the system was first turned on according to [14]. During our analysis, we divided the dataset captured under normal conditions into three parts: SN which comprises the 80% of the original normal dataset and is used for the training of the model, V_{N1} which comprises the 10% and is used for early stopping in order to avoid over-fitting and V_{N2} which comprises the remaining 10% and is used for determining the threshold along with the 10% of the dataset that contains anomalies denoted by V_{AB1}. The remaining 90% of the anomalous dataset S_{AB} is used for testing.

Baseline methods: We compare the proposed model with the following baseline methods: One-Class SVM [11] learns a decision function and classifies the test dataset as similar or dissimilar to the training dataset. LSTM-ED [17] represents the temporal dependencies of the training dataset and predicts the value

of the test dataset. In LSTM-ED model the average prediction error over the all-time series is considered as the anomaly score. The anomaly score of each time point is equal to the reconstruction error of the respective correlation matrix. If that value is larger than a given threshold which is determined empirically over different datasets, then we consider that an attack is taking place. Otherwise, we assume normal behavior. The above baseline methods are state-of-art anomaly detection algorithms that can be used for raw time series data. The proposed method is implemented in Python 3.5.6 with use of TensorFlow framework version 1.11 [24]. The baseline methods, i.e., One Class SVM and LSTM encoder-decoder, are likewise created in Python 3.5 using the Scikit-learn library [25] and TensorFlow framework version 1.11, respectively. Experiments are performed on a Linux server with 6 vCPUs and 15GB of memory.

Evaluation metrics: In order to evaluate the anomaly detection performance of each method, we use Precision, Recall and F1 scores defined as $Prec = \frac{tp}{tp+fp}$ $Rec = \frac{tp}{tp+fn}$, and $F_1 = \frac{2 \times precision \times recall}{precision+recall}$, where tp, fp and tn denote true positives, false positives, and false negatives, respectively. To detect anomalies, we determine a threshold $\tau = \beta \cdot max\{Val(t)\}$ where $Val(t)$ are the anomaly scores over the join of the sets V_{N2} and V_{AB1}, and $\beta \in [1, 2]$ is a constant tuned to maximize the F_1 Score over validation period Recall and Precision scores over the testing period are computed based on this threshold.

Other settings In order to avoid over-fitting, we used early stopping while training the model. Furthermore, Dropout [26] is employed with probability 0,4 in the recurrent layers. Furthermore, we fix the batch size at 128. The learning rate and epoch are set to 0.01 and 1000 respectively. The model used hyperbolic tangent non-linearity (tanh) as activation function. The proposed model uses an input and output length of four.

5.2 Experimental Results

We first demonstrate that the ST-ED scheme as the baseline method LSTM-ED can learn the system features and predict them with high precision. We note that the respective parameter settings and configurations are described in Section 4.3 for the ST-ED architecture and Section 5.1 for LSTM-ED. The classification baseline method OC-SVM is omitted here since traditional models behave differently compared to deep learning methods. As mentioned in Section 4.2, we used different scales or window lengths (ℓ) to characterize the system status. In other words, ST-ED and its variants were trained, validated, and tested on correlation matrix samples built under different windows lengths. For LSTM-ED, we used the raw time-series data from the SWaT dataset. We identified that given adequate computational capacity, the deep learning models were able to achieve an RMSE within the range of 0.00323 (ST-ED) to 0.09873 (LSTM-ED), as summarized in Table I. Fig. 3(a). Illustrates how the test error rate of ST-ED architecture changes as the employed window length varies. In particular,

the lowest error is achieved when the window length is equal to 120 (i.e., two minutes). In Fig. 3(a) and Table I, it can be seen that the ST-ED scheme achieves the best convergence, generating the lowest error

	Test RMSE ($\times 10^{-2}$)				Train epoch time (sec)				Test epoch time (sec)				Model size (KB)
	90	120	150	Raw data	90	120	150	Raw data	90	120	150	Raw data	
ST-ED	0.328	0.323	0.355	-	1004	692	717	-	248.7	217.5	220		12.014
LSTM-ED	-	-	-	9.873	-	-	-	57	-	-	-	13	2.329

Table 1: Training/Test Result.

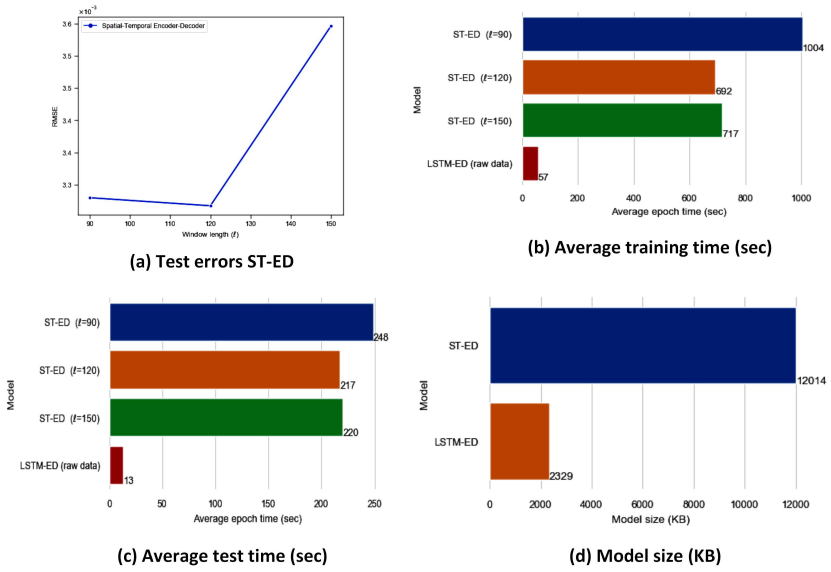

Fig. 3: Evaluation of the models

In the following, we compare the training and test times, as well as the different model sizes, which are presented in Fig. 3(b), Fig. 3(c) and Fig. 3(d), respectively. In particular, Figures 3(b)-3(c) demonstrate the average time/duration per epoch, as measured at the workstation machine described in Section 5.1 during training and testing. We found that the LSTM-ED model exhibits the fastest/shortest training and testing times due to its application to raw data, while also being the smallest model in terms of size. Another observation is that the ST-ED scheme can learn faster when $\ell = 120$. The results are summarized in Table I.

Subsequently, we evaluate the models' performance on the six stages of the SWaT dataset in terms of precision (Pre), recall (Rec), and F1 score. Experiments on the dataset are repeated five times, and the average results are reported for comparison, presented in Table 2. We observe that the classification method

(OC-SVM) perform worse than the prediction model (LSTM), indicating that the traditional methods cannot handle adequately the temporal dependencies that exist in the dataset. The LSTM-ED and the ST-ED with $\ell = 120$ architectures yield the largest precision. However, the ST-ED model achieves the largest recall and F1 score for all the employed window sizes. Hence, this verifies that the proposed spatial-temporal encoder-decoder is efficient at identifying anomalies or outliers. Next, we demonstrate how the performance of the ST-ED scheme varies with regard to the employed sequence window lengths $\ell = \{90, 120, 150\}$. In particular, ST-ED with $\ell = \{90, 120\}$ has better precision than ST-ED with $\ell = 150$, whereas ST-ED with $\ell = 120$ has better recall and F1 score compared to the other window lengths. Fig. 4 provides a visual representation of the ability of the ST-ED and LSTM-ED methods to detect anomalies.

Method	Raw time series data			$\ell = 90$			$\ell = 120$			$\ell = 150$		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
OC-SVM	0.210	0.423	0.281	-	-	-	-	-	-	-	-	-
LSTM-ED	0.951	0.627	0.756	-	-	-	-	-	-	-	-	-
ST-ED	-	-	-	0.929	0.695	0.795	0.949	0.705	0.809	0.927	0.686	0.788

Table 2: Anomaly detection results.

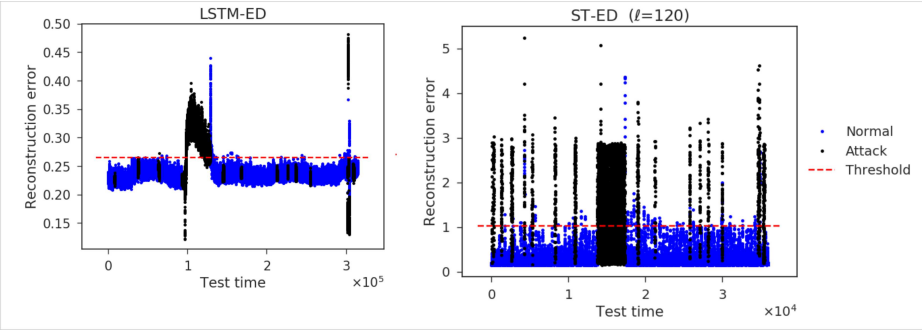

Fig. 4: Anomaly detection representation

6 Conclusion

In this paper, we have presented an anomaly detection model for the complex CPS networks based on the combination of a convolutional autoencoder, which learns the spatial structure of each correlation matrix, with a ConvLSTM Encoder-Decoder that captures the temporal dependencies from the learned spatial feature maps of every time step. We have demonstrated its improved performance compared to two baseline models in terms of Recall and F1 metrics. Moreover, the proposed model, contrary to study [18], is able to model both inter-sensor correlation and temporal dependencies of multivariate time series.

One limitation of the present work is the fact that the experiments were performed on one dataset from one type of industrial process. Various adversarial attacks can be carried out against the proposed model. One such attack can alter the training process by influencing and corrupting the training data. On the other hand, an exploratory attack can employ probing to discover information about the training set. The potential adversary cannot modify or manipulate the training data but can craft new instances based on the underlying data distribution. Therefore, it is necessary to explore reactive and proactive defense strategies in order to take countermeasures for adverse attacks. Apart from addressing the aforementioned issues, this research can be expanded in several directions: i) investigating the application of recent adversarial autoencoders as well as adversarial variational autoencoder to anomaly detection; ii) introducing an input attention mechanism to adaptively select the most significant input features; iii) exploring other methods to performance the correlation analysis that are robust to non-normality of the data; v) amplify the scope of the proposed model to anomaly diagnosis, i.e., identifying the most likely cause of an anomaly; iv) applying the proposed anomaly detection method to streaming data.

References

1. ENISA, "European Union Agency for Network and Information Security,https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends. Last accessed 7 Jan 2019.
2. ICS-CERT,TheIndustrialControlSystemsCyberEmergencyResponseTeam,https://ics-cert.us-cert.gov. Last accessed 7 Jan 2019.
3. STELLIOS, Ioannis, et al. A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. IEEE Communications Surveys & Tutorials, 2018, vol. 20, no 4, p. 3453-3495.
4. KHATOUN, Rida; ZEADALLY, Sherali. Cybersecurity and privacy solutions in smart cities. IEEE Communications Magazine, 2017, vol. 55, no 3, p. 51-59.
5. ABRAMS, Marshall; WEISS, Joe. Malicious control system cyber security attack case study–Maroochy Water Services, Australia. McLean, VA: The MITRE Corporation, 2008.
6. FALLIERE, Nicolas; MURCHU, Liam O.; CHIEN, Eric. W32. stuxnet dossier. White paper, Symantec Corp., Security Response, 2011, vol. 5, no 6, p. 29.
7. Blake Johnson, Dan Caban, Marina Krotofil, Dan Scali, Nathan Brubaker, and Christopher Glyer. 2017. Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure.https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploynew-ics-attack-framework-triton.html. Last accessed 7 Jan 2019.
8. MURGUIA, Carlos; RUTHS, Justin. Characterization of a cusum model-based sensor attack detector. En 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016. p. 1303-1309.
9. KRAVCHIK, Moshe; SHABTAI, Asaf. Detecting cyber attacks in industrial control systems using convolutional neural networks. En Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. ACM, 2018. p. 72-83.

10. MAGLARAS, Leandros, et al. Novel Intrusion Detection Mechanism with Low Overhead for SCADA Systems. En Security Solutions and Applied Cryptography in Smart Grid Communications. IGI Global, 2017. p. 160-178.
11. INOUE, Jun, et al. Anomaly detection for a water treatment system using unsupervised machine learning. En 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2017. p. 1058-1065.
12. LIN, Qin, et al. TABOR: a graphical model-based approach for anomaly detection in industrial control systems. En Proceedings of the 2018 on Asia Conference on Computer and Communications Security. ACM, 2018. p. 525-536.
13. GOH, Jonathan, et al. Anomaly detection in cyber physical systems using recurrent neural networks. En 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). IEEE, 2017. p. 140-145.
14. GOH, Jonathan, et al. A dataset to support research in the design of secure water treatment systems. En International Conference on Critical Information Infrastructures Security. Springer, Cham, 2016. p. 88-99.
15. CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection: A survey. ACM computing surveys (CSUR), 2009, vol. 41, no 3, p. 15.
16. ANTON, Simon Duque, et al. Two decades of SCADA exploitation: A brief history. En 2017 IEEE Conference on Application, Information and Network Security (AINS). IEEE, 2017. p. 98-104.
17. MALHOTRA, Pankaj, et al. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148, 2016.
18. KRAVCHIK, Moshe; SHABTAI, Asaf. Detecting cyber attacks in industrial control systems using convolutional neural networks. En Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. ACM, 2018. p. 72-83.
19. WANG, Lin, et al. Abnormal Event Detection in Videos Using Hybrid Spatio-Temporal Autoencoder. En 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018. p. 2276-2280.
20. SONG, Dongjin, et al. Deep r-th root of rank supervised joint binary embedding for multivariate time series retrieval. En Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018. p. 2229-2238.
21. HALLAC, David, et al. Toeplitz inverse covariance-based clustering of multivariate time series data. En Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017. p. 215-223.
22. LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. En Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 3431-3440.
23. XINGJIAN, S. H. I., et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. En Advances in neural information processing systems. 2015. p. 802-810.
24. ABADI, Martín, et al. Tensorflow: A system for large-scale machine learning. En 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). 2016. p. 265-283.
25. PEDREGOSA, Fabian, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research, 2011, vol. 12, no Oct, p. 2825-2830.
26. SRIVASTAVA, Nitish, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 2014, vol. 15, no 1, p. 1929-1958.
27. KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.