ASSESSING MACHINE LEARNING ALGORITHMS FOR INDUSTRIAL NETWORK ANOMALY DETECTION

Dayaker P, Assistant Professor, Department of MCA Loyola Academy Degree & PG College, Hyderabad, Telangana, India
P V Naga Lakshmi, Assistant Professor, Department of MCA Loyola Academy Degree & PG College, Hyderabad, Telangana, India

Abstract: *Industrial Control Systems (ICSs) cyber-physical security is a real and valuable research issue. This study compares and assesses various machine learning (ML) techniques to identify anomalies in industrial control networks. Using datasets taken from the Secure Water Treatment (SWaT), a testbed designed to mimic a scaled-down genuine industrial plant, we examine supervised and unsupervised machine learning-based anomaly detection techniques. Our tests highlight the advantages and disadvantages of the two ML-based industrial network anomaly detection techniques.*

Keywords—Machine Learning, Anomaly Detection, Industrial Control System, Cyber-Physical System

1. Introduction: Contemporary Supervisory Control and Data Acquisition (SCADA) systems keep an eye on the events, processes, and physical components of industrial plants—such as actuators and sensors. Although ICSs' widespread and growing interconnection capabilities enhance performance, they also create new opportunities for cyber-physical threats. With very minimal human interaction, device-to-device or device-to-computer connections predominate. These days, Cyber-Physical Systems (CPSs) serve as the basis for a large number of ICS applications that require actuators and sensors to carry out monitoring. Refineries, power plants, nuclear power plants, and water distribution systems are a few instances of ICS. Many malware attacks on intrusion detection systems (ICSs) have been reported in the past few years. The most well-known is Stuxnet [1], while more recent ones draw attention to the security flaws in ICSs like Triton [2] and BlackEnergy [3]. To safeguard these systems, it is crucial to research and create cutting-edge cyber-physical security techniques. Issues with ICSs might have disastrous effects. Furthermore, because some ICSs are classified as vital infrastructures, harming or eliminating them could hurt both the environment and the populace. Large volumes of data are produced by industrial infrastructures these days. It is difficult to filter, analyse, and make operational and security choices for ICSs because of this enormous volume of traffic. Artificial intelligence and machine learning have been used in decision-making processes to address these issues. In actuality, these methods make it possible to evaluate enormous volumes of data and produce precise conclusions that analysts otherwise would not be able to afford. This sector has seen a great deal of activity in recent years, with remarkable advancements in research. When anomaly detection systems are combined with machine learning and artificial intelligence, performance and accuracy can be increased beyond what can be achieved with conventional methods. Thus, to create and evaluate innovative, accurate anomaly detection systems to prevent possible damages to ICS.

Here is a summary of our contributions:

- We analyse ML-based supervised and unsupervised anomaly detection techniques.
- We test the discovered ML-based anomaly detection techniques using the SWaT testbed dataset.
- We talk about how ML-based methods can be useful in identifying risks that impact ICSs.

A widely researched issue in ICSs is anomaly detection.

2. Literature Review:

The lack of readily available datasets for evaluations plagues the scientific community. Lemay and Fernandez [4] offer a dataset of a SCADA system that has been sandboxed for simulation. Several data collections have been made in this case. Their study contains a detailed explanation of the simulated scenario. In this work, hostile activities are introduced and exploited with the use of penetration testing tools like Metasploit. One of the most common problems with the dataset used is that it uses Metasploit as a source of assaults; while there are several Modbus-based attacks described in [5], none of them are included in the dataset. Nonetheless, extensive research was done using the Lemay dataset; for instance, in [6], the authors contrasted supervised and unsupervised methods and demonstrated the superiority of the former. Using actual ICS testbeds and creating ad hoc offensive and defensive strategies, such as the anomaly detection systems described in [7], [8], is another way to look into ICS cyber-physical security.

This paper examines the SWaT testbed dataset [9] and will pay specific attention to the physical data analysis. There are thirty-six distinct cyberattacks in the SWaT dataset. This dataset has been the subject of numerous experiments to evaluate the efficacy of various ML detection algorithm types. The authors of [10] identified attacks on physical dataset sensors using unsupervised techniques. The authors of [11] presented a window-based anomaly detection technique in which a neural network model forecasts the values of the data features in the future based on past values.

An anomaly detection system can handle its data in one of two ways. The first involves analysing each component separately, while the second involves grouping the components in time series and examining the components that have collapsed. Although the latter does not permit the identification of individual harmful elements, it does permit the use of temporal properties derived from the union of several elements. We used the first analysis approach in this work to find every single harmful element. Additionally, in contrast to the aforementioned publications, we compare supervised and unsupervised methodologies to highlight the key distinctions.

3. ICS TESTBED AND DATASET

A. Data availability problem for ICS security research

Cyber-attack datasets are necessary for the security community to identify, test, and assess novel approaches for detection and prevention to comprehend trends and consequences of physical and cyberattacks on ICSs. However, conducting novel security research is challenging since there are few datasets available that contain assaults in ICS networks. The scarcity of accessible datasets derives from privacy issues and the impossibility of testing cyber-physical attacks on vital real-world systems. While professional testbeds [12] or inexpensive ones [13] allow some research groups to explore new security techniques, most of the time there is a barrier to the implementation and assessment of cyber-physical security measures and cyber-physical threats. To overcome this problem, research groups provide useful datasets of their testbed communication. One example is the SWaT testbed with its dataset.

B. The Secure Water Treatment (SWaT) testbed

To provide a realistic ICS environment for the safe testing of offensive and defensive cyber-physical techniques, the Singapore University of Technology and Design (SUTD) constructed the SWaT testbed. Figure 1 describes the procedures for operating the testbed. The testbed is a miniature version of an actual water treatment facility that generates clean water.

From process P1 to process P6, the water passes through a six-step filtering process, with a specific number of sensors and actuators at each level. For additional details regarding the SWaT testbed, please see [14].

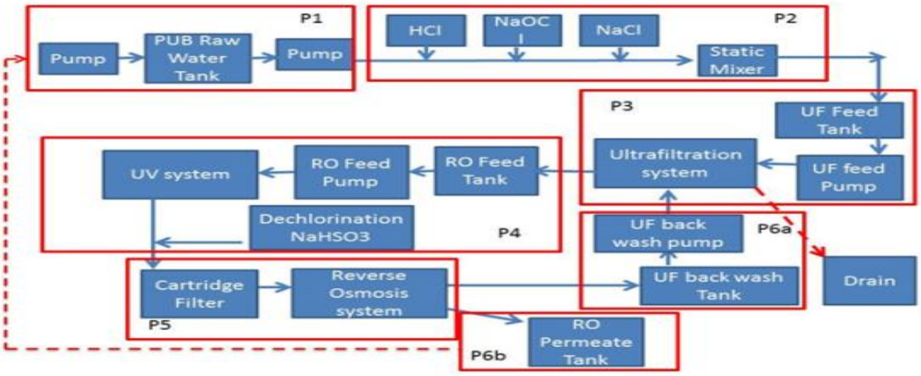


Fig 1. SWaT tested

C. The dataset SWaT

The SWaT dataset includes four days of data recording in addition to seven days of data recording during normal circumstances.

where thirty-six attacks were carried out in total. There are 946722 elements in the dataset overall, and each one is classified as either attack or normal. Targeting a single step of the process or several phases at once are two possible attack strategies. The dataset documentation includes a comprehensive table with every attack, its related times, and its results.

One limitation of the dataset is that, as the authors note in [9], there are instances of overlap when multiple rows have the same timestamp but indicate separate activities because the data was collected at a per-second interval. Similarly, the data was labelled according to the start and finish times of the attacks without making a distinction between malicious components and those that were regular, based on the attack logs. Consequently, these issues have an impact on the integrity of the data and could jeopardise the analysis's conclusions.

We have selected attack number three for our analysis in this work out of the 36 attacks that were carried out. The file under consideration includes the appropriate label along with both benign and harmful data. Scalar values captured by sensors at regular intervals make up the dataset. During this time, the attack is carried out by raising the water level by one millimetre per second. Tank overflow as a result damages the P-101 sensor. In a real-world situation, these implications might be disastrous. We chose to split the dataset into the following sections because it contains a large number of sensors.

ds1: This dataset includes all of the testbed's sensor data collected throughout the attack phase. This dataset is examined to determine whether the modification of one sensor (LIT-101) affects the other sensors and, as a result, offers more data for the detection.

ds2: this dataset solely includes information from the LIT-101 sensor, which is the target of the attack. Out of the 2701 entries in each of the derived datasets, 383 are malicious, or approximately 14.18% of the total. Figure 2 shows a normal and harmful number of components.

IV. METHODS FOR ML-BASED ANOMALY DETECTION

We describe the ML-based algorithms that we utilised to carry out our evaluation in this Section. We examine the advantages and disadvantages of the two strategies by contrasting the supervised (Section IV-A) and unsupervised (Section IV-B) methods.

Part A: Supervised Code

a) Support Vector Machine: The Support Vector Machine (SVM) was first presented by Boser et al. in 1992 [15]. A supervised machine learning technique called SVM is useful for both regression and classification. The primary goal of this framework is to establish a divider—the so-called large margin classifier—between two data sets so that every element is as far away from the divider as possible. The training algorithm receives a set of m instances (xi) with labels (yi) in the following format.

$$(x_i, y_i) \quad i = 1, \ldots, m, \tag{1}$$

where each y_i is equal to:

$$y_i = \begin{cases} 1 & \text{if } x_i \text{ belongs to the first class} \\ -1 & \text{if } x_i \text{ belongs to the second class} \end{cases}, \tag{2}$$

Once the training phase is over, the decision function, namely *signum function*, is defined as follows:

$$z_i = sgn(w, x_i - b), \tag{3}$$

The output vector, z, represents the attribution to one of the two classes, while b is the offset from the hyperplane. SVMs can be applied in one of two scenarios: either the instances can or cannot be separated by a linear function. If the instances cannot be divided linearly, a kernel trick [16] transformation is used. Using this technique, the input space is nonlinearly mapped into a higher dimensional feature space, allowing the algorithm to produce a linear divider.

b) Random Forest: Made up of a union of Decision Trees [17] that are filled during the training phase, the Random Forest (RF) is a supervised learning technique used for both regression and classification tasks. A root node, internal nodes—also referred to as split nodes—and leaf nodes make up RF. Every class that the algorithm predicts has a corresponding leaf node. Through a majority vote, an element is assigned to a class. When faced with noise or overfitting, which are frequent issues in machine learning, RF works admirably.

c) k-Nearest Neighbour: The k-Nearest Neighbour (KNN) algorithm is a non-parametric technique [18] that may be applied to regression and classification issues alike. The KNN method assumes that similar objects are located nearby. Stated differently, similar objects are located close to one another. In actuality, the categorization is predicated on the properties of objects in the vicinity of the object under consideration, generally as measured by the Euclidean distance or other particular distance metrics. The definition of the Euclidean distance is as follows:

$$D = \sqrt{\sum_{i=1}^{n}(x_i - p_i)^2}, \qquad (4)$$

In n-dimensional vectors, the components are denoted by xi and pi. The class of the considered point is determined as the most frequent among the first k points with the smaller distance after the distance between the considered point and every other point is calculated.

B. Unsupervised Algorithms

a) One-Class SVM: An example of a special instance of the conventional SVM is the One-Class Support Vector Machine (OCSVM) [19]. OCSVMs are trained with a single class that reflects typical behaviour. The OCSVM is a One-Class Classification (OCC) method specifically. By setting up a decision boundary, OCCs attempt to distinguish objects belonging to a particular class from all other things. To accomplish this, a set of components that solely contain the objects of the designated class is used to train the OCSVM model. As a result, having every training element labelled the same is the same as having none at all. The OCSVM is regarded as an unsupervised learning model for these reasons [20], [21].

The OCSVM model can predict which elements do not belong to the normal class by inferring the properties of the normal elements once it has been trained with the normal class. Outliers are elements that do not fit into the typical class.

b) Autoencoder: Fully linked neural networks trained to reconstruct their input are called autoencoders (AEs) [22] training materials. By first compressing the input (encoder), the AEs automatically learn a "compressed representation" of the input (which might be an image, text sequence, etc.) and decode it back to match the original input by decompressing it. One input, one output, and one (or more) hidden layer make up an AE network. The size of the hidden layer can be adjusted based on the use case, but the input and output layers must always have the same size. This type of neural network has gained prominence in the past several years for solving practical issues, such as anomaly identification. In actuality, an AE that has been trained on test set X can reconstruct previously undiscovered occurrences from the same data distribution. Reconstruction-based detection is the term for this type of detection [23]. We anticipate the reconstruction to have a significant error rate if an instance does not fit inside the ideas gained from X. The Root Mean Squared Error (RMSE) between x¯ and the reconstructed output y¯ can be used to calculate the Reconstruction Error (RE) of the instance x¯ for a given AE. RMSE between two vectors can be expressed as follows:

$$RMSE(\bar{x}, \bar{y}) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}, \qquad (5)$$

where n denotes the input vectors' dimension. One of the most important stages in the development of AE is threshold selection. A threshold that is set too high could lead to more false negative results, and a threshold set too low could result in more false positive results. Since the AEs in our instance are only trained with normal classes, the threshold selection strategy is based on that which was employed in [24]. Following neural network training, a 20% validation set with only normal components is taken from the training set, and the maximum of the validation set's REs is used as the threshold value.

V. EVALUATION AND RESULTS

This section covers the evaluation criteria used to compare the ML-based anomaly detection algorithms in Section V-B and the implementation specifics of the ML methods in Section V-**A. Next, we go over the findings from the studies in Section V-C.**

To conduct the assessment, we divided every dataset into 80% for the detection test and 20% for training. For the unsupervised algorithm, we take advantage of a semi-supervised learning technique used in the instruction. This indicates that a subset of the training set's normal elements is used to train the models. We use the Scikit-learn library [25] for Python to build the SVM, RF, KNN, and OCSVM algorithms. All of the estimator classes in the Scikit-learn library implement a fit (Xtrain, ytrain) method to fit the model given a training set (Xtrain) and the matching labels (ytrain). For unsupervised algorithms, like the OCSVM, this method takes only the Xtrain value.

The Scikit-learn library uses the predict(Xtest) function to classify the unlabeled observations Xtest and returns the anticipated labels y^test. We utilise the Keras [26] library to create the AE neural network. A primary challenge in neural network implementation is the issue of fine-tuning parameters. An input layer and an output layer with the same dimension of the feature number make up the implemented AE model. Half of the neurons in the input and output layers are found in the hidden layers. s. In the case of ds2, where there is only one feature, the number of neurons is set to 1.

We evaluate a laptop with the following specifications:

- CPU: Intel(R) Core (TM) i7-3537U 2.00GHz x 4.
- RAM: 10GB DDR3.
- Operative System: Ubuntu 18.10 64-bit.

B. Metrics for Evaluation

The following metrics are used to assess the performance of the machine learning-based anomaly detection algorithms:

• **Accuracy:** this is the percentage of the model's predictions that are accurate. The accuracy in the binary classification situation is defined as follows in terms of positives and negatives:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (6)$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

• **F1-Score:** is a metric used to evaluate a classification by taking into consideration both precision and recall as follows:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \qquad (7)$$

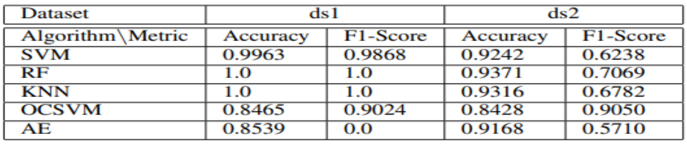
where:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}. \quad (8)$$

C. Results

The outcomes of the ML-based anomaly detection algorithms for the datasets under investigation are shown in Table I. All of the recordings are contained in the dataset ds1 the sensor data; in contrast, ds2 only includes the data from the targeted sensor, Lit-101. The findings demonstrate that in the case of ds1, which is the derived dataset characterised by all sensor readings, all algorithms-aside from the AE—perform better. This indicates that the other sensors' conditions are impacted by the attack on the LIT101 sensor as well, which helps to identify the malicious activity. Additionally, RF and KNN get a detection score of 1.0 on ds1, indicating the high efficacy of their detection strategies on this dataset. SVM does well on this dataset as well; it is unable to identify a single element, allowing the data to be separated from a hyperplane. OCSVM demonstrates its discrete nature by outperforming supervised algorithms but outperforming AE. Once more, the outcome highlights the data's divisibility by the hyperplane, which is superior to the RE-based method in terms of data classification. The F1-score of AE on ds1 is null, indicating that no harmful items have been identified. The performances are typically poorer in the situation of ds2, that is, in the dataset when only the targeted sensor is taken into consideration. This is because the data gleaned from the compromised sensors is insufficient to pinpoint the malicious activity. Figure 3 displays the graphical representation of the Accuracy for each algorithm, and Figure 4 displays the F1-Score representation.

TABLE I
THE RESULTS OF THE ML-BASED ANOMALY DETECTION ALGORITHMS ON THE DATASETS.



Dataset	ds1		ds2	
Algorithm\Metric	Accuracy	F1-Score	Accuracy	F1-Score
SVM	0.9963	0.9868	0.9242	0.6238
RF	1.0	1.0	0.9371	0.7069
KNN	1.0	1.0	0.9316	0.6782
OCSVM	0.8465	0.9024	0.8428	0.9050
AE	0.8539	0.0	0.9168	0.5710

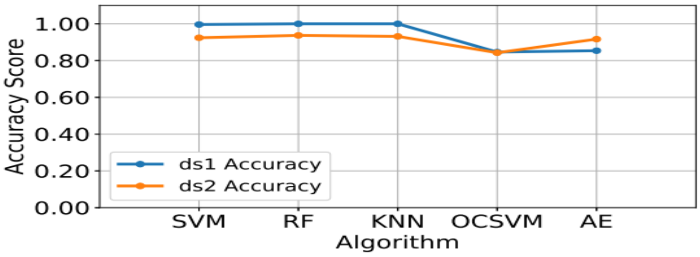


Fig. 3. Accuracy representation for ML-based anomaly detection algorithms.

We examine the RE of AE on this dataset to look into the lower scores that all of the algorithms on ds2 obtained in comparison to ds1. The distance between the test set's elements and the compressed representation of the training set that was committed to memory during the AE training phase can be easily visualised through the examination of the RE. Figure 5 presents the findings. When a prediction is made on the test set, elements that are similar to the normal ones will have a low RE value, while abnormal elements will have a high RE value because AE was trained only with elements that were labelled as normal.

Figure 5 demonstrates that elements are classified as harmful even when the device LIT-101's physical condition is still regarded as normal because it hasn't started to change. Consequently, the dataset's poor labelling is what led to the low accuracy score. This results in a low F1-score even though the procedure was applied correctly. The reason for this issue is that the malicious labelling's beginning point was set at the moment the attack began rather than the moment the data began to fluctuate. While it is evident on AE, this issue also arises for the other algorithms when analysing ds2.

D. Discussion on ML-based anomaly detection algorithms for ICS security

In light of the outcomes, we concluded that algorithms that employ supervised learning typically outperform unsupervised ones in terms of performance. This is a result of the supervised algorithms' use of a priori knowledge obtained from the dataset's labelling. In contrast, no information of any kind is used by the unsupervised algorithms. instead, they search the data for latent information to identify recurring themes. One fundamental drawback of the supervised technique is that model training necessitates labelling the dataset. We discover that there is a labelling issue with the dataset following a more thorough examination of the AE findings (the issue also arises in the other algorithms). Data that are equal to normal are also classified as anomalous, which results in a low F1-Score and a high number of FN elements.

The trials demonstrate the potential power of machine learning techniques in identifying anomalies. Communications between various devices in ICS systems are continuous and repetitive because they are defined by polling time. When managing vast amounts of data, the application of intelligent and self-adaptive approaches makes it possible to discover abnormalities even in situations where humans are unable to.

VI. Conclusion

In this work, we assessed machine learning (ML)-based methods for industrial control networks anomaly detection. Specifically, we used datasets taken from the SWaT testbed to analyse supervised and unsupervised methods. The experiments conducted demonstrate the potentials and constraints of supervised and unsupervised algorithms used for anomaly identification on datasets from industrial control systems networks. For future work, we will implement the ML-based approaches identified on ad hoc ICS simulation networks and real ICS testbeds to deepen the analysis and develop effective anomaly detection systems.

REFERENCES

[1] N. Falliere, L. O. Murchu, and E. Chien, “W32. stuxnet dossier,” White paper, Symantec Corp., Security Response, vol. 5, no. 6, p. 29, 2011.

[2] R. M. Lee, TRISIS: Analyzing Safety System Targeting Malware. DRAGOS, 2017.

[3] R. M. Lee, M. J. Assante, and T. Conway, “Analysis of the cyber-attack on the ukrainian power grid,” SANS Industrial Control Systems, vol. 23, 2016.

[4] A. Lemay and J. M. Fernandez, “Providing {SCADA} network data sets for intrusion detection research,” in 9th Workshop on Cyber Security Experimentation and Test ({CSET} 16), 2016.

[5] T. Morris and W. Gao, “Industrial control system traffic data sets for intrusion detection research,” in International Conference on Critical Infrastructure Protection. Springer, 2014, pp. 65–78.

[6] S. D. Anton, S. Kanoor, D. Fraunholz, and H. D. Schotten, “Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set,” in Proceedings of the 13th International Conference on Availability, Reliability and Security. ACM, 2018, p. 41.

[7] H. R. Ghaeini and N. O. Tippenhauer, “Hamids: Hierarchical monitoring intrusion detection system for industrial control systems,” in Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. ACM, 2016, pp. 103–111.

[8] E. E. Miciolino, R. Setola, G. Bernieri, S. Panzieri, F. Pascucci, and M. M. Polycarpou, “Fault diagnosis and network anomaly detection in water infrastructures,” IEEE Design & Test, vol. 34, no. 4, pp. 44–51,2017.

[9] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, “A dataset to support research in the design of secure water treatment systems,” in International Conference on Critical Information Infrastructures Security. Springer, 2016, pp. 88–99.

[10] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, “Anomaly detection for a water treatment system using unsupervised machine learning,” in 2017 IEEE International Conference on Data Mining
Workshops (ICDMW). IEEE, 2017, pp. 1058–1065.

[11] M. Kravchik and A. Shabtai, “Detecting cyber-attacks in industrial control systems using convolutional neural networks,” in Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. ACM, 2018, pp. 72–83.

[12] G. Bernieri, E. E. Miciolino, F. Pascucci, and R. Setola, “Monitoring system reaction in cyber-physical testbed under cyber-attacks,” Computers & Electrical Engineering, vol. 59, pp. 86–98, 2017.

[13] G. Bernieri, F. Del Moro, L. Faramondi, and F. Pascucci, “A testbed for integrated fault diagnosis and cyber security investigation,” in 2016 International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2016, pp. 454–459.

[14] A. P. Mathur and N. O. Tippenhauer, “Swat: a water treatment testbed for research and training on ics security,” in 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater). IEEE, 2016, pp. 31–36.

[15] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM, 1992, pp. 144–152.

[16] V. Vapnik, “Universal learning technology: Support vector machines,” NEC Journal of Advanced Technology, vol. 2, no. 2, pp. 137–144, 2005.

[17] L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp.5–32, 2001.

[18] N. S. Altman, “An introduction to kernel and nearest-neighbour nonparametric regression,” The American Statistician, vol. 46, no. 3, pp.175–185, 1992.

[19] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. ¨Williamson, “Estimating the support of a high-dimensional distribution,” Neural computation, vol. 13, no. 7, pp. 1443–1471, 2001.

[20] L. A. Maglaras and J. Jiang, “Intrusion detection in scada systems using machine learning techniques,” in 2014 Science and Information Conference. IEEE, 2014, pp. 626–631.

[21] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, “A geometric framework for unsupervised anomaly detection,” in Applications of data mining in computer security. Springer, 2002, pp. 77–101.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California University San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[23] D. Zimmerer, S. A. Kohl, J. Petersen, F. Isensee, and K. H. Maier-Hein, “Context-encoding variational autoencoder for unsupervised anomaly detection,” arXiv preprint arXiv:1812.05941, 2018.