Adversarial Sample Generation for Anomaly Detection in Industrial Control Systems

Abdul Mustafa*, Muhammad Talha Khan *, Muhammad Azmi Umer †, Zaki Masood †, Chuadhry Mujeeb Ahmed ‡
*DHA Suffa University, Karachi, Pakistan
{cs172049, cs162021}@dsu.edu.pk
†Singapore University of Technology and Design, Singapore
{azmi_umer, zaki_masood}@sutd.edu.sg
‡Newcastle University, Newcastle, United Kingdom
mujeeb.ahmed@newcastle.ac.uk

***Abstract*—Machine learning (ML)-based intrusion detection systems (IDS) are vulnerable to adversarial attacks. It is crucial for an IDS to learn to recognize adversarial examples before malicious entities exploit them. In this paper, we generated adversarial samples using the Jacobian Saliency Map Attack (JSMA). We validate the generalization and scalability of the adversarial samples to tackle a broad range of real attacks on Industrial Control Systems (ICS). We evaluated the impact by assessing multiple attacks generated using the proposed method. The model trained with adversarial samples detected attacks with 95% accuracy on real-world attack data not used during training. The study was conducted using an operational secure water treatment (SWaT) testbed.**

***Index Terms*—Adversarial samples, Cyber physical systems, Industrial control system (ICS) security, Sensors and actuators, JSMA.**

I. Introduction

Industrial control systems (ICS) comprise a significant portion of any state or nation's critical infrastructure (CI). Examples of such systems include water treatment plants and electric power grids, where an ICS regulates the physical processes. The physical processes consist of two primary parts: monitoring and controlling. The monitoring part maintains processes and ensures they are operating properly by measuring various signals acquired from sensors. The controlling part handles processes and makes decisions that enable actuators to perform actions [1].

ICS and their modules were previously thought to be safe against cyber-attacks since they ran on proprietary hardware, software, and air-gapped networks that were not connected to the internet [2]. However, as connectivity with the internet provides online access and monitoring functionalities, it has led to the necessity of connecting ICS components to other networks, subsequently contributing to the digitalization of industrial systems [2].

Given their applicability, these systems have become a tempting target for attackers. Since these systems regulate real-world processes, cyber-attacks on them could have serious consequences for the ecosystems in which they are used, as well as for end-users [3]. As a result, it is evident that the security concerns of such systems are a serious global issue, and it is necessary to design robust systems to defend against cyber-attacks. Various security methods have been proposed for traditional IT systems, but applying them to ICS systems is complex since ICS devices have limited resources. They contain outdated systems and devices that do not support advanced safety mechanisms. Alternatively, security solutions such as passive monitoring of process data appear to be promising [4].

As a corollary, there has been a significant rise in research into ICS-tailored intrusion detection systems (IDS). These IDS monitor network or sensor data for attacks and anomalies that could impact ICS. Machine learning has seen a remarkable increase in use and integration with IDS due to its accuracy in detecting attacks [5]. Unfortunately, the emergence of such systems has opened up a new attack vector, i.e., trained models can be targeted as well. Adversarial Machine Learning (AML) involves launching attacks on machine learning-based IDS models by exploiting flaws in the trained models, such as "blind spots" among training examples. Specifically, by introducing minor perturbations to data points not seen during training, it is possible to cross decision boundaries and classify data into different classes. As a consequence, the model's performance may suffer when encountering previously unseen data points, leading to an increase in misclassifications. AML can be used to manipulate data received via actuators and other devices. In the context of ICS, this is done by introducing perturbations that cause attack data to be categorized as normal, thereby evading the IDS. This could result in late detection of an attack, information leakage, financial loss, etc.

In the current study, we assess the effectiveness of adversarial samples generated using machine learning against cyber-attacks on a water treatment plant [6]. This work uses the secure water treatment (SWaT) (December 2015) attacked dataset [7] to generate adversarial samples using Jacobian saliency map attack (JSMA). By utilizing JSMA, we can more accurately assess the effectiveness of the adversarial samples. Although the adversarial samples are generated using attack data, however, this attack data was never used directly in training the machine learning classifiers. Therefore, the result of such a trained model with high accuracy is an indication

of the usefulness of adversarial sample generation techniques, in attack detection. In this paper, we utilize a real-world SWaT attack dataset to provide a more realistic evaluation of our solution's performance under different machine learning classifiers with varying accuracy.

The main contributions of our paper are as follows:

- This paper presents an approach to generate synthetic adversarial samples using JSMA. It also highlights that the JSMA which was originally designed for image media, could be extended to generate adversarial samples for time series data. In particular, our approach is significant in understanding the potential weaknesses in current ML algorithms, particularly in security-sensitive applications.
- To validate our approach, we conducted various experiments demonstrating the practical effectiveness of synthetic adversarial samples against ML-based IDS. Our results show a significant decrease in the detection rate of the IDS when exposed to these adversarial samples, underscoring the critical need for enhanced security measures in ML-based approaches.

The remaining sections of this paper are organized as follows. Section II highlights the related work. Section III gives a brief overview of the water treatment control systems. Section IV presents our proposed ML framework for anomaly detection in the SWaT system. Then, we apply our framework to a real-world SWaT system. We evaluate the performance of ML models and showed the results in Section V. Finally, Section VI concludes the paper and offers insights for future research.

II. Comparison with Related Work

The majority of research on adversarial machine learning has focused on the computer vision area. However, we believe it is important to extend prior work to other domains, including cyber-physical systems, which are vulnerable to real-world attacks [8]. A study reported in [9] used machine learning to generate attack patterns for an operational water treatment plant. They used the same SWaT attack dataset that we have used in our proposed study. They employed Association Rule Mining (ARM) to generate a large number of attack patterns for SWaT. These attack patterns can later be used as a dictionary for signature-based anomaly detection. However, our proposed study creates perturbations in the attack data. The perturbed attack data is used to create more robust supervised machine learning models for anomaly detection. The literature related to attacker models for ICS and model-based tools for SWaT risk assessment is described in the subsequent subsection.

A. Attacker Models for ICS

According to [10], adversarial samples were generated using three distance metrics: L_0, L_2, and L_∞. These samples were created using limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), fast gradient sign method (FGSM), and JSMA, utilizing the modified National Institute of Standards and Technology (MNIST) database and Canadian Institute for Advanced Research (CIFAR) datasets. In a recent study, the authors generated adversarial examples using an upgraded projected gradient descent (PGD) and an upgraded Carlini and Wagner (C&W) method. The authors claim that both proposed algorithms required less time to generate adversarial examples [11].

B. Model-based Tools for SWaT Risk Assessment

Besides the computer vision domain, various techniques have been used for generating adversarial samples in the cybersecurity domain. In [12], the authors generated adversarial samples using popular adversarial deep learning attack methods, such as JSMA, FGSM, and Carlini Wagner (CW), with modern IDS datasets (UNSW-NB15 and Bot-IoT). The study reported in [13] generated adversarial samples using FGSM against condition-based maintenance (CBM) capabilities, evaluating the performance of a CBM system under attack.

When attacking an autoencoder IDS, the authors in [4] generated adversarial ICS attacks by substituting original data with readings within the normal sensor range. Although each sensor reading could be replaced from an arbitrary initial value to a value within the normal range, the perturbation applied in this method could be significantly large. Using the SWaT dataset, the authors in [14] introduced stealthy poisoning during the training phase to avoid detection in the test phase. They developed attacks for a residual signal threshold-based detector using seven attacks from the dataset.

A new adversarial attack method, called the Selective and Iterative Gradient Sign Method, was proposed in [15], which required less time compared to the basic iterative method (BIM). The authors in [16], [2], and [17] evaluated long short-term memory (LSTM) networks for detecting cyber-physical attacks in the SWaT infrastructure, achieving an optimal LSTM with an F1 score of 0.80.

III. SWaT Vulnerabilities and Attack Data

A. SWaT Testbed

The SWaT plant is an operational testbed available at the Singapore University of Technology and Design (SUTD)

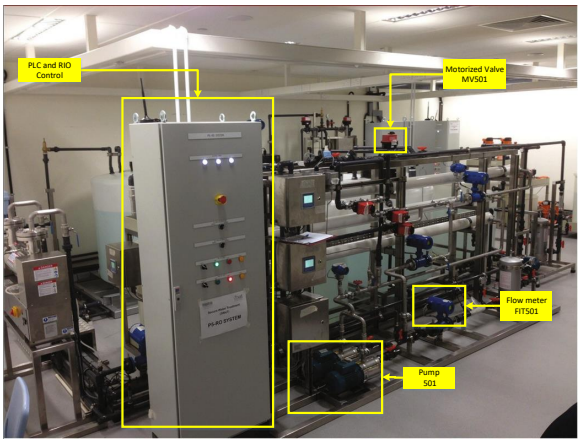

Fig. 1: An overview of real-time SWaT system.

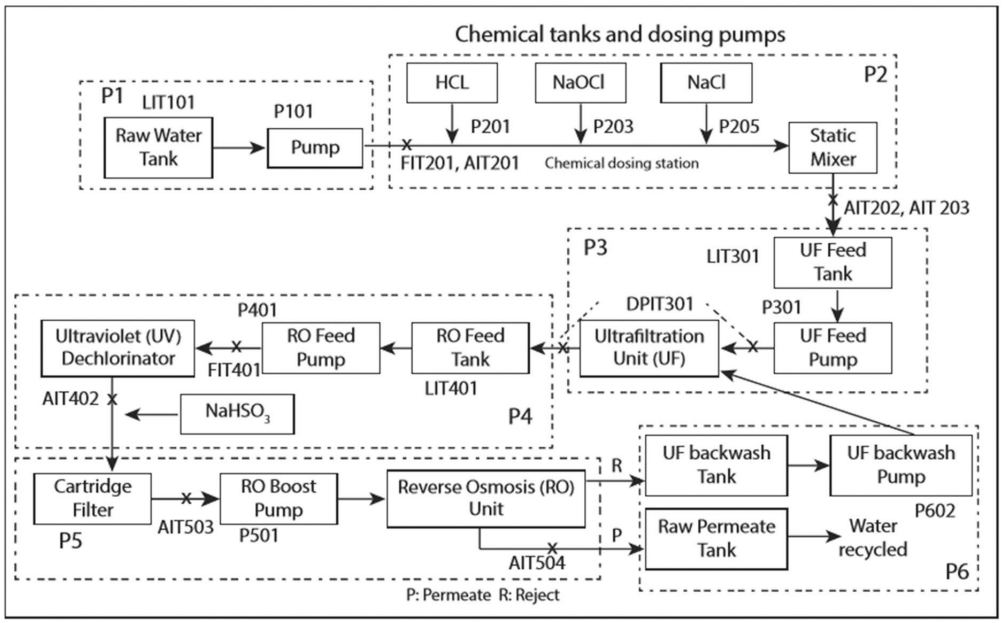

Fig. 2: Illustration of SWaT testbed [18].

[19]. A pictorial view of SWaT is shown in Figure 1. It can produce five gallons of treated water per minute. SWaT is a distributed control system (DCS) that consists of six stages, as shown in Figure 2. Each stage comprises a group of sensors and actuators, totaling 51 sensors and actuators. The sensors measure physical properties such as the water flow rate in pipes, the water level in tanks, and pressure. Moreover, chemical monitoring sensors calculate a range of properties, including water conductivity, oxidation-reduction potential, and pH level. The actuators regulate the flow rate of water and chemical dosing. The six stages of SWaT, from P1 to P6, are summarized in Figure 2.

- **P1**: ensures that the raw water tank has adequate water to supply the other processes.
- **P2**: responsible for guaranteeing the quality standards of water.
- The water is then sent to **P3**, once it has reached the required purity. In this stage, an ultra-filtration (UF) system with a fine filtration membrane removes any leftover unwanted items in the water.
- All leftover chlorine is removed through dechlorination using ultraviolet (UV) rays in **P4**. The next step is to minimize the number of inorganic contaminants in the water.
- The water in **P4** is then pumped into **P5** for reverse osmosis (RO).
- The treated water is then distributed in **P6**.

B. SWaT Attack Data

A large number of researchers have used the SWaT testbed to examine cyber-attacks and their defense mechanisms for ICS [9], [20]–[24]. The SWaT dataset [7] was generated by running the plant continuously for eleven days. During the first seven days, the plant was operated in a normal state. In the remaining four days, a total of 36 different attacks were performed on the SWaT testbed. The duration and goals of these attacks varied, with several attempting to cause underflow/overflow conditions in water treatment tanks, while others aimed to break pipes and stop filtration processes. The attacked points, according to the type of attack and stage, are presented in [25]. One such attack targeted the level sensor LIT-101 of Stage P1, where the goal was to overflow the tank by manipulating the values of the LIT-101 sensor and turning off the pump P-101 [26].

For instance, in Fig. 3a, the water level readings from sensor LIT301 clearly show the water consumption pattern from the tank. It illustrates the daily water consumption, where normal pumping events occur to fill the tank. The threshold must be maintained within the specified range. However, due to the cyber attack, the water level fell below the lower limit, posing a critical risk that requires immediate mitigation, as illustrated in Fig. 3b.

By analyzing the water level data, if an attack lowers the readings, it could falsely trigger (1) the pump to operate and (2) increase the risk of overflow. This inevitably results in more pumping events, leading to increased energy consumption. It is essential to monitor the float level and identify any false positives or false negatives. Similarly, in the case of the flow meter in Fig. 4, the attack data reveals significant variations in flow, which can lead to potential malfunctions. In the following section, we propose an ML framework that uses attack data while incorporating adversarial sampling.

C. Threat Model for Water Treatment Plant

In this section, we formalize the types of attacks launched on our secure water treatment testbed (SWaT) as explained

above. Essentially, the attacker's model encompasses the attacker's intentions and capabilities. The attacker may choose its goals from a set of intentions [27], including performance degradation, disturbing a physical property of the system, or damaging a component. These goals include under-flowing and over-flowing the water tank, bursting pipes, intentionally wasting water by passing it to the drain, and unnecessarily reducing the water in the tank.

It is assumed that the attacker knows the system dynamics and the control inputs and outputs. We consider a strong adversary who is able to launch both cyber and physical attacks. In an ICS, sensors, actuators, and PLCs communicate with each other via communication networks. An attacker can compromise these communication links in a classic *Man-in-The-Middle (MiTM)* attack [28]–[30], for example, by breaking into the link between sensors and PLCs. Besides false data injection in sensor readings via the cyber domain, an adversary can also physically tamper with a sensor to drive an ICS into an unstable state. Sensors can be connected to remote *input/output* units via wired and wireless connections. A cyber attacker can remotely spoof sensor readings without needing physical access.

Data Injection Attacks: For data injection attacks, it is considered that an attacker injects or modifies the real sensor measurements. The attacker's goal is to deceive the control system by sending incorrect sensor measurements. In this scenario, the level sensor measurements are increased while the actual tank level remains unchanged. This makes the controller think that the attacked values are true sensor readings, causing the water pump to keep working until the tank is empty, which can lead to the pump burning out. The attack vector can be defined as,

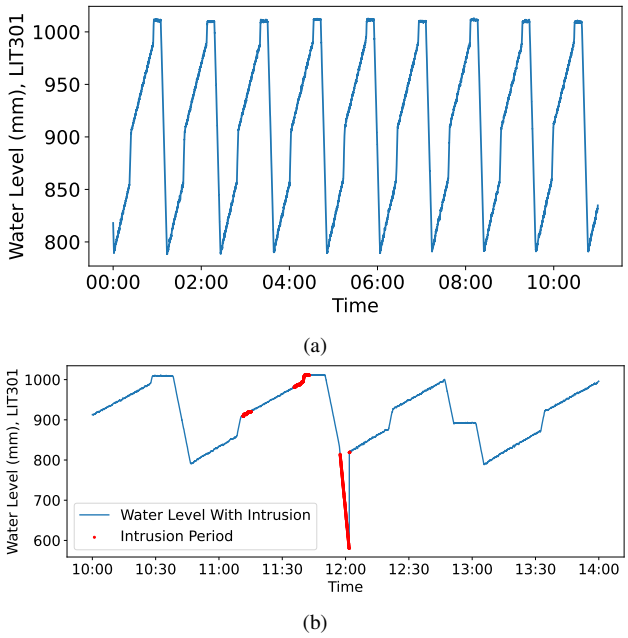

Fig. 3: Example of normal (a) and attack (b) data, tank water level readings from sensor LIT301.

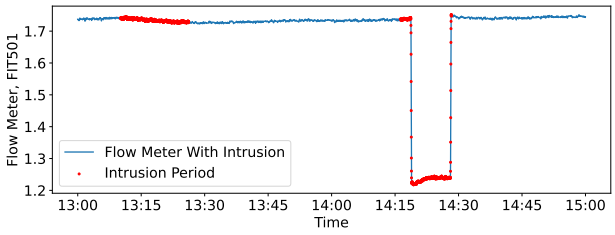

Fig. 4: An example of attack data, showing flow meter readings from sensor FIT501.

$$\bar{y}_k = y_k + \delta_k. \tag{1}$$

Where y_k is the sensor measurement, $\bar{y}_k$ is the sensor measurement with the attacked value, and δ_k is the bias injected by the attacker. We can obtain a similar expression for an actuator attack as well.

IV. Application of Adversarial Machine Learning using JSMA

Machine learning (ML) models are subject to adversarial attacks, where the attacker modifies input data to cause misclassification. An adversarial sample is carefully designed to disrupt the performance of a machine learning classifier. The attacker creates malicious inputs to fool the machine learning algorithms during the test phase [31]. This is one of the techniques in Adversarial Machine Learning (AML). Designing and developing robust ML-based algorithms to resist cyber-attacks is also part of this technique [32].

Specific characteristics of the attack model, the adversary, and the defenses are described in relevant research on AML. According to [33], such an attack has three primary characteristics. The attacker's capability is referred to as influence, which might be causative or exploratory, i.e., changing the input training data and learning classifier decisions after sending instances to the classifier. Security violation is the second property, which includes integrity, availability, and privacy. The third property is the attack's specificity: indiscriminate (the goal is to fail the classifier across a wide range of classes) and targeted (the goal is to fail the classifier for a specific instance). There are two types of potential attacks described by the threat models: black-box attacks, where the attacker is unaware of the model, and white-box attacks, where the attacker has knowledge of the model.

Adversarial samples can be generated using various approaches. The complexity, speed of generation, and performance of these methods vary. Manual perturbation of input data points is a naive method of creating such samples. However, manual perturbations are slower to develop and analyze than automated techniques. The Jacobian-based Saliency Map Attack (JSMA) was introduced by Papernot et al. in [31]. The authors used JSMA for image recognition tasks. In the current study, we have used it to generate adversarial samples for time series data composed of different sensors and actuators of the SWaT.

The JSMA approach uses saliency maps to generate perturbations. The saliency map identifies the important features of input data for classification; if these features are changed, the target values will most likely be classified differently. A percentage of (θ) is perturbed using (γ), i.e., the noise. The model then determines whether the introduced noise has led to misclassification by the targeted model. If the model's performance is unaffected by the noise, a new collection of features is chosen, and a new cycle begins until a saliency map that can generate the adversarial samples emerges. The technique acquires the Jacobian matrix as described in equation 2, where i is the input component and j is the class derivative for input sample x.

$$J_F(x) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial F_j(x)}{\partial x_i}\right]_{i \times j} \tag{2}$$

$$S(x,t)_i = \begin{cases} 0, & \text{if } \frac{\partial F_t(x)}{\partial x_i} < 0 \\ & \text{or} \sum_{k \neq t} \frac{\partial F_k(x)}{\partial x_i} > 0 \\ \frac{\partial F_t(x)}{\partial x_i} \cdot \left| \sum_{k \neq t} \frac{\partial F_k(x)}{\partial x_i} \right|, & \text{otherwise} \end{cases} \tag{3}$$

$$x_i^{\text{new}} = x_i + \epsilon \tag{4}$$

Here, x_i represents the i-th feature of the input sample x. In eq 3 the saliency map is calculated, the input is iteratively modified by selecting the feature x_i with the highest saliency score as described in eq 4. Here, ϵ is the perturbation step size that is chosen to be small enough to maintain a gradual change in the perturbation.

In Equations (2) and (3), **F** represents the output of the penultimate layer (related to the output of the softmax layer) [34]. The perturbation is selected, and the method is iterated until the target is misclassified or the maximum number of perturbed features is reached. If this fails, the algorithm proceeds to the next feature, which is then added to the perturbed sample. For instance, the authors in [34] achieved a 97% adversarial success rate while modifying only 4.02% of the input features per sample. This procedure requires complete knowledge of the design and parameters of the target model [31].

V. EXPERIMENTAL SETUP

We have used CleverHans V.3.0.0 library to generate the adversarial samples more specifically to implement JSMA [35]. The Tensorflow V.1.14.0 [36], and Keras V.2.0.0 [37] were used for the pre-processing, experimental evaluations, and analysis. Our study is based on a binary classification problem as the SWaT dataset comprises two classes i.e., attack or normal. Based on this we trained multiple models using various algorithms. Here we have summarized the experimental evaluation for the top three algorithms i.e., Classification And Regression Trees (CART), Random Forest (RF), and Gradient Boosting Classifier (GBC).

A. Data Pre-processing

It is crucial to structure the dataset in the pre-processing step, especially for supervised machine learning. To transform nominal values into numerical values, we used label encoding. For instance, in the SWaT dataset [7], the target label has two nominal values, namely attack and normal, which need to be mapped into their respective numerical values, with normal and attack mapped to 0 and 1, respectively. Since the dataset contains features with different distributions, min-max normalization was applied to all features after label encoding. For min-max normalization, 0 and 1 were chosen as the minimum and maximum range, respectively.

B. Adversarial Sample Generation

We used two publicly available SWaT datasets. There are 51 attributes in the SWaT datasets. Among these attributes, 25 are related to sensor readings and the remaining 26 are related to actuator readings. The first SWaT dataset was collected during the normal operation of the plant, which we refer to as the normal dataset. This dataset contains 410,400 transactions and was collected at a frequency of one transaction per second. The second dataset was collected by performing 36 attacks at different time instances. We refer to this dataset as the attack dataset. This dataset contains 449,919 transactions and was also collected at a frequency of one transaction per second. The attack dataset contains 53,900 anomalous and 396,019 normal transactions. We used the attack dataset to generate the adversarial samples using JSMA. The Cleverhans library was used for the implementation of JSMA. A Multi-Layer Perceptron (MLP) was chosen as the pre-trained underlying model for the generation of adversarial samples. We generated 112,480 adversarial samples using the proposed approach.

C. Supervised Model Training using Adversarial Samples

The 112,480 adversarial samples generated earlier were merged with the SWaT normal dataset, which contains 410,400 transactions. Therefore, the merged dataset contains 522,880 transactions. This merged dataset was used to train the supervised models. In particular, we used CART, RF, and GBC for this purpose. Among these algorithms, CART is a simple decision tree algorithm, while RF and GBC are ensemble algorithms that use decision trees. The ensemble algorithms build a collection of classifiers and take a vote from each classifier's predictions to classify new data points [38]. The main purpose of training these models is to test the effectiveness of generative adversarial samples on attack detection. Therefore, we evaluated the performance of the trained models on the SWaT attack dataset. The complete process of adversarial sample generation, model training, and evaluation is described in Figure 5.

D. Results

The performance of the previously trained models was tested using the SWaT attack dataset. The results are shown

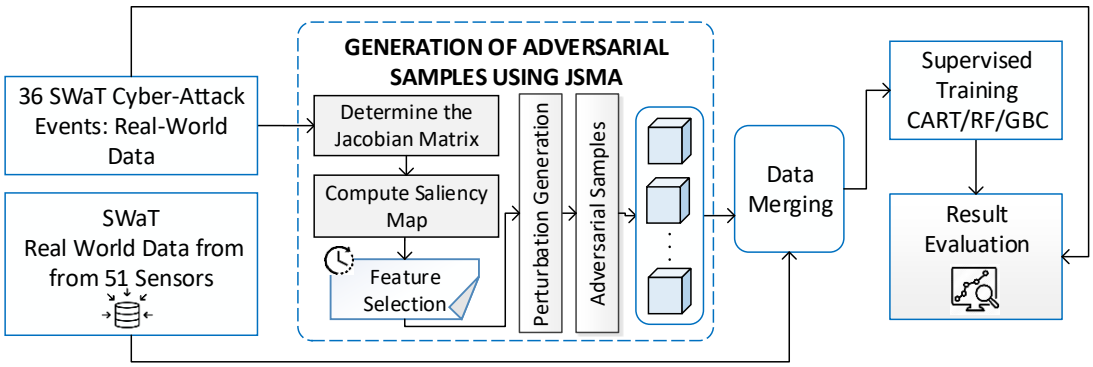

Fig. 5: Proposed workflow for evaluating 36 SWaT cyber-attack events using real-world data from 51 sensors. The process includes supervised training with CART/RF/GBC and adversarial sample generation via JSMA.

TABLE I: Results from various modern ML classifiers.

Classifier	Accuracy			Precision			Recall		
	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best
CART	0.88	0.90	0.95	0.88	0.90	0.94	0.99	0.99	0.99
RF	0.88	0.88	0.88	0.88	0.88	0.88	1.0	1.0	1.0
GBC	0.95	0.95	0.95	0.95	0.95	0.95	1.0	1.0	1.0

TABLE II: FPR and F1-Score for various modern ML classifiers.

Classifier	FPR			F1-Score		
	Worst	Average	Best	Worst	Average	Best
CART	0.99	0.80	0.41	0.94	0.95	0.97
RF	1.0	1.0	1.0	0.94	0.94	0.94
GBC	0.37	0.37	0.37	0.97	0.97	0.97

TABLE III: Confusion Matrix of CART, RF, GBC

	Worst Score			Average Score			Best Score		
	Predicted Class → True Class ↓	Normal	Attack	Predicted Class → True Class ↓	Normal	Attack	Predicted Class → True Class ↓	Normal	Attack
CART	**Normal**	395935	84	**Normal**	395935	84	**Normal**	395954	65
	Attack	53854	46	**Attack**	53854	46	**Attack**	21927	31973
RF	**Normal**	396019	0	**Normal**	396019	0	**Normal**	396019	0
	Attack	53900	0	**Attack**	53900	0	**Attack**	53900	0
GBC	**Normal**	396019	0	**Normal**	396019	0	**Normal**	396019	0
	Attack	20236	33664	**Attack**	20178	33722	**Attack**	19840	34060

in Tables I, II, and III. The SWaT attack dataset is highly imbalanced, with normal-class samples far outnumbering attack-class samples. Therefore, accuracy alone can be misleading. Instead, we evaluated the proposed technique using additional metrics such as precision, recall, false positive rate (FPR), and F1-Score. All these metrics are defined in the following.

Where True Positive (TP) represents the attack instances that are correctly classified as attack. The True Negative (TN) represents the normal instances that are correctly classified as normal. The False Positive (FP) represents the normal instances that are incorrectly classified as attack. The False Negative (FN) represents the attack instances that are incorrectly classified as normal. The TP, FP, TN, and FN form the confusion matrix of the ML classifier. Note that in calculating the F1 score and accuracy, we determine the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) at a trace level:

- **True Positive:** A faulty trace that is flagged instances that are correctly classified as normal.
- **False Positive:** Either a normal trace that is flagged as faulty or a faulty trace that is flagged as faulty before the time of occurrence of the fault.
- **True Negative:** represents the attacked instances that are correctly classified as attacked.
- **False Negative:** A faulty trace that is not flagged as faulty.

We have presented three scenarios: worst, average, and best case for accuracy, precision, recall, FPR, and F1-Score for each classifier, as shown in Tables I and II. Accuracy represents the overall performance of the classifier. CART and GBC achieved a maximum accuracy of 95%. However, the average scores of both classifiers differ, with CART at 90% and GBC at 95%. As mentioned earlier, the current problem is class imbalance; therefore, accuracy alone is not sufficient to assess the performance of classifiers. We also calculated the precision and recall of all the classifiers. Precision here represents the

performance of classifiers in identifying the normal instances in the dataset, while recall represents the identification of normal instances with respect to the total normal instances in the dataset. There is a trade-off between precision and recall. For this purpose, we use another metric, the F1-score, which is the harmonic mean of precision and recall. Improving the F1-score helps maintain the balance between precision and recall. Additionally, we calculated the false positive rate for each classifier, as a high rate of false positives makes the IDS impractical for real-world applications. For an in-depth evaluation of the classifiers' performance, the confusion matrices of each classifier are given in Table III.

From the confusion matrix of RF in Tables I and II, it is evident that the classifier was unable to differentiate between attack and normal instances. Consequently, it classified all attack instances as normal, even though its accuracy is 88%. The confusion matrix of CART in Table III shows that its performance was better than RF in detecting attack instances. The confusion matrix of GBC in Table III shows that it performed better than CART not only in detecting attacks but also in classifying normal instances. These results highlight that the examples generated by JSMA proved useful in improving the accuracy of detectors without needing to be trained on attack data.

VI. Conclusions

In this paper, we assessed the quality of the malicious data created by the JSMA attack method, using the SWaT dataset as a testbed. Although JSMA was originally designed to create perturbations for image data, it was successfully exploited for time series data. Machine learning classifiers often lack sufficient data to defend against attacks. Our results show that the proposed approach improves the performance of these classifiers against previously unseen attacks. Future work will focus on enhancing the robustness of IDS against a wider range of adversarial attacks. This includes exploring other adversarial attack methods and developing more sophisticated defense mechanisms. Additionally, we plan to extend our evaluation to other ICS datasets to further validate the effectiveness of our approach. Investigating the integration of ML-based IDS with other security measures in ICS will also be a key area of future research.

Acknowledgement

This research is supported in part by the National Research Foundation, Singapore, under its National Satellite of Excellence Programme "Design Science and Technology for Secure Critical Infrastructure: Phase II" (Award No: NRF-NCR25-NSOE05-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, and Félix J García Clemente. Madics: A methodology for anomaly detection in industrial control systems. *Symmetry*, 12(10):1583, 2020.

[2] Moshe Kravchik and Asaf 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*, pages 72–83, 2018.

[3] Yosef Ashibani and Qusay H Mahmoud. Cyber physical systems security: Analysis, challenges and solutions. *Computers & Security*, 68:81–97, 2017.

[4] Alessandro Erba, Riccardo Taormina, Stefano Galelli, Marcello Pogliani, Michele Carminati, Stefano Zanero, and Nils Ole Tippenhauer. Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems. *arXiv preprint arXiv:1907.07487*, 2019.

[5] Muhammad Azmi Umer, Khurum Nazir Junejo, Muhammad Taha Jilani, and Aditya P Mathur. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, 38:100516, 2022.

[6] 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)*, pages 31–36, 4 2016.

[7] iTrust. Dataset and models. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/, 2021.

[8] Ihai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Adversarial machine learning attacks and defense methods in the cyber security domain. *arXiv preprint arXiv:2007.02407*, 2020.

[9] Muhammad Azmi Umer, Chuadhry Mujeeb Ahmed, Muhammad Taha Jilani, and Aditya P Mathur. Attack rules: an adversarial approach to generate attacks for industrial control systems using machine learning. In *Proceedings of the 2th Workshop on CPS&IoT Security and Privacy*, pages 35–40, 2021.

[10] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[11] Shize Huang, Xiaowen Liu, Xiaolu Yang, Zhaoxin Zhang, and Lingyu Yang. Two improved methods of generating adversarial examples against faster r-cnns for tram environment perception systems. *Complexity*, 2020, 2020.

[12] Yulexis Pacheco and Weiqing Sun. Adversarial machine learning: A comparative study on contemporary intrusion detection datasets. In *ICISSP*, pages 160–171, 2021.

[13] Hamidreza Habibollahi Najaf Abadi. Adversarial machine learning attacks on condition-based maintenance capabilities. *arXiv preprint arXiv:2101.12097*, 2021.

[14] Moshe Kravchik, Battista Biggio, and Asaf Shabtai. Poisoning attacks on cyber attack detectors for industrial control systems. *arXiv preprint arXiv:2012.15740*, 2020.

[15] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, Félix J García Clemente, and Frances Cleary. Crafting adversarial samples for anomaly detectors in industrial control systems. *Procedia Computer Science*, 184:573–580, 2021.

[16] Jun Inoue, Yoriyuki Yamagata, Yuqi Chen, Christopher M Poskitt, and Jun Sun. Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1058–1065. IEEE, 2017.

[17] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 140–145. IEEE, 2017.

[18] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *International conference on critical information infrastructures security*, pages 88–99. Springer, 2016.

[19] Aditya P Mathur and Nils Ole 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)*, pages 31–36. IEEE, 2016.

[20] Khurum Nazir Junejo and Jonathan Goh. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, CPSS '16, page 34–43, New York, NY, USA, 2016. Association for Computing Machinery.

[21] Muhammad Azmi Umer, Aditya Mathur, Khurum Nazir Junejo, and Sridhar Adepu. Generating invariants using design and data-centric approaches for distributed attack detection. *IJCIP*, 28:100341, 2020.

[22] Dušan M Nedeljković, Živana B Jakovljević, Zoran DJ Miljković, and Miroslav Pajić. Detection of cyber-attacks in systems with distributed control based on support vector regression. *Telfor Journal*, 12(2):104–109, 2020.

[23] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[24] Chuadhry Mujeeb Ahmed, Muhammad Azmi Umer, Beebi Siti Salimah Binte Liyakkathali, Muhammad Taha Jilani, and Jianying Zhou. Machine learning for cps security: Applications, challenges and recommendations. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, pages 397–421. Springer, 2021.

[25] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*, 2018.

[26] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *CRITIS*, pages 88–99, Cham, 2017. Springer.

[27] S. Adepu and A. Mathur. Generalized attacker and attack models for cyber physical systems. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 283–292, 6 2016.

[28] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105. ACM, 2016.

[29] Sridhar Adepu and Aditya Mathur. Distributed detection of single-stage multipoint cyber attacks in a water treatment plant. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 449–460, New York, NY, USA, 2016. ACM.

[30] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems x2014;part i: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, 9 2013.

[31] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[32] A. Cardenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry. Attacks against process control systems: Risk assessment, detection, and response. In *6th ACM Symposium on Information, Computer and Communications Security*, pages 355–366, 2011.

[33] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, 2006.

[34] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[35] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, Patrick McDaniel, et al. cleverhans v2. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 10, 2016.

[36] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015.

[37] François Chollet. keras. https://github.com/fchollet/keras, 2015.

[38] Thomas G. Dietterich. Ensemble methods in machine learning. In *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857*, pages 1–15. Springer, 2000.