Securing Industrial Control System(ICSs) With Digital Twin

Abubakar Abubakar Yusif,
Normaziah Binti Abdul Aziz (Assoc. Prof. Dr.).
Department Of Computer Science, Kulliyyah of ICT, International Islamic University Malaysia
abuyusif01@gmail.com

Abstract

In the era of Industry 4.0, ensuring security in critical infrastructure like Industrial Control Systems (ICSs) is paramount. Digital twins for industrial control systems have taken place, the increased capabilities of digital twins in the fields of simulation, optimization, and predictive maintenance have attracted a lot of interest. Throughout recent research, the use of digital twins for intrusion detection and vulnerability detection in industrial control systems has been greatly highlighted. This project aims to study digital twins and apply it for monitoring cyber attacks on an ICS. As part of the study, a prototype of digital twin is developed with a real-time Application Programming Interface (API), capable of detecting, reporting and possibly mitigating intrusions and abnormalities on a network, with the help of a Supervised Machine Learning (ML)trained Model.

Keywords— Industrial Control System (ICSs), Machine Learning (ML), Digital Twins (DTs), Intrusion, Vulnerability.

1. Introduction

The evolution of Industry 4.0 Changed our perspectives on Virtualization and Simulation. By giving us almost real-world like copies of a system using 3D.

A digital twin replica takes this to the next level by giving us the ability to virtually copy a system's functionality just like how it is in real word, as well as analyzing its performance in the realtime (Synchroni-zed) with an unpredictable input.

As ICSs become more connected to communication networks, they become more vulnerable to cyberattacks. Due to the sheer importance of the industrial activities handled by these systems, ensuring their security against cyberattacks is critical and should be examined thoroughly. Considering the uniqueness characteristics of ICSs, performing penetra- tion testing on certain sorts of systems might be difficult or in some cases impossible, as it might result in disruption of services or sometimes permanently damaging the system, bearing in mind those systems are required to be always online, Digital Twin provides more options in terms of computing resources while having no negative influence on the efficiency of running systems. Using DT as a component that allows a system to simulate a physical system in the digital realm in real-time. As a result, security analysis and vulnerability identifica- tion using digital twins seem to be an effective way to protect ICSs against cyberattacks.

2. Related Works

Prior to the advent of the "digital twin" (DT) concept, the utilization of real-time data for simulations was not a focal point for many industries[16]. Traditional simulations primarily relied on static inputs and historical data, often conducted in isolated environments with limited integration of real-time data feeds. The concept of the Digital Twin addresses the challenges posed by simulating systems in real-time and accounting for the unpredictable nature of user input. It enables the design of a system by closely mirroring its physical counterpart. Consequently, all the works that have been reviewed share a common thread with Digital Twin and virtualization, aligning with these concepts to varying degrees.

A framework has been proposed by Eckhart et. al [1] called CPS Twinning; This framework can automatically generate the digital twin of an ICS using Mininet-WiFi from the specification of the ICS. Two operation modes of the digital twin are supported in this framework, the first one is simulation mode where there is no need for coexistence of the physical system, and the second one is replication mode which supports synchronization with the physical system.
Gehrmann et al [2] proposed a digital twin- based security architecture for IACS, and it mainly focuses on detailing the security requirements for different

components of the proposed architecture. It also introduces the concept of an active state replication approach using clock synchronization at regular intervals to achieve synchronization between the physical twin and the digital twin.

M Dietz et al [3] demonstrated the feasibility of integrating digital twin security simulations into the Security Operations Center; the framework proposed in [3] is as a microservice architecture using Docker containers. A digital twin implementation using Mininet and MiniCPS is used to achieve security simulation, and security analytics is performed with a Security Information and Event Management (SIEM) module that uses a rule-based attack detection from system logs.

Seba Anna [4] proposed a Digital Twin-base Intrusion detection system for Industrial Control Systems. It uses supervised Machine Learning to detect a cyber attack on a Network. The project uses docker containers, LogStash with mininet to simulate the chosen Digital twin. In addition, the proposed solution Utilized SIEM to show system Logs in real-time as well as abnormalities if any

In contrast to other approaches, Qinghu et al. [5] introduced a novel method that employs generators for identifying abnormalities in Industrial Control Systems (ICSs) using a digital twin. The proposed solution, named Anomaly detection with digital twin (ATTAIN) involves the continuous and automated creation of a digital twin through real-time data from a Cyber-Physical System (CPS) to detect anomalies. ATTAIN constructs a digital representation of the CPS in the form of a Timed Automaton Machine (TAM) and employs a Generative Adversarial Network (GAN) to identify anomalies

Philip et al. [6] introduced the SOAR4IoT Framework, which is designed to enhance the management and security of Internet of Things (IoT) through the integration of Digital Twin Technology. The authors emphasize the challenges associated with safeguarding and sustaining Industrial Control Systems, attributing these difficulties to financial constraints, human fallibility, and the inherent complexity of system usage. This research amalgamates IoT security with Digital Twin principles to establish a robust security framework capable of identifying improper usage of IoT devices. Additionally, the study showcases the efficacy of this approach in promptly identifying and halting the operation of a potentially harmful botnet, highlighting the utility of digital twin system logs in achieving near real-time threat mitigation.

Akbarian et al. [7] proposed an alternative approach in contrast to the methodology introduced by Gehrmann et al. [2]. Their focus was on developing an intrusion detection framework with the ability to identify and categorize the severity of attacks. Employing supervised machine learning techniques, their solution aimed to establish an environment for patching Industrial Control Systems (ICSs) without jeopardizing the integrity of actual systems. The attack classification algorithm utilized in this research effectively segregates data into discrete classes, with Support Vector Machine being the chosen machine learning algorithm for model training. Furthermore, for attack detection, they incorporated an existing method known as the Kalman filter. This filter serves the purpose of accurately estimating the system's correct output, thereby enabling optimal mitigation of the destructive impact of attacks. By effectively suppressing the adverse effects caused by attacks, the filter facilitates the identification of network intrusions in compromised systems.

3. Methodology

Our methodology presents a prototype-driven approach for enhancing the security of Industrial Control Systems (ICSs) through the integration of Digital Twin Technology. The prototype comprises a dynamic digital twin model that mirrors the behavior of the physical ICS components. Leveraging supervised machine learning techniques, specifically Support Vector Machine, the Model is trained to discern normal and anomalous patterns within the system.

To achieve real-time monitoring and analysis, an Application Programming Interface (API) is developed to enable seamless communication between the digital twin and the actual ICS components. This API facilitates continuous data exchange, enabling the prototype to effectively detect anomalies as they occur. In tandem, an interactive dashboard prototype is implemented to visually represent attack and defense scenarios. The prototype's anomaly detection capabilities trigger real-time notifications within the dashboard, providing administrators with timely insights into potential threats.

Moreover, our approach extends beyond the confines of the dashboard, encompassing a comprehensive email notification system. This enhancement enhances the prototype's efficacy by stratifying detected anomalies according to their severity levels. Upon anomaly

detection, the system dynamically assesses the gravity of the anomaly and, if deemed necessary, initiates an email alert to apprise administrators. By seamlessly integrating this dual-layered notification mechanism—consisting of real-time dashboard updates and severity-driven email alerts—the prototype ensures that administrators are expeditiously informed and equipped to make well-informed decisions, thereby fortifying the security stance of the ICS.

The prototype's effectiveness is evaluated through a series of simulated attack scenarios, measuring key metrics such as attack detection accuracy and dashboard responsiveness. These evaluations demonstrate the practical viability of our prototype-driven approach in bolstering ICS security.

3.1 Design

The model architecture adopted in this project aligns with the proposal presented by M. Eckhart and A. Ekelha [1]. The rationale for selecting this specific architectural framework over alternative options is multifaceted: it offers a fully open-source solution, operates as an independent application, and faithfully emulates real-world use cases of Industrial Control Systems (ICSs). For a comprehensive overview of the comparative analysis among various reviewed solutions, refer to Table 1.0.

NO	Proposed Solution	Summary	Open Source?	Standalone Solution?	Real-word ICS Use case?
1	[18]	Generates digital twin automatically from system specification using Mininet-WiFi	Yes	Yes	Yes
2	[16]	Digital twins run as Virtual Machine (VM)s on isolated environments	No	No	No
3	[17]	Digital twin simulation using Mininet and MiniCPS; physical process as a simple simulation in Python.	Yes	Yes	Yes
4	[19]	Uses a benchmark industrial dataset as a cronjob for evaluation purposes.	No	Yes	No

Table 1: DT Proposed Design comparison

While the solution proposed by M. Eckhart and A. Ekelha [1] serves as the foundation, certain modifications are essential to elevate its capabilities. These enhancements encompass the development of an API, creation of an Administrative Dashboard, and establishment of a real-time reporting system. To achieve these goals, we retain the core network design while effecting changes to other components. As depicted in *Figure 3 [1].* presents the initial proposed solution, whereas *Figure 4* illustrates the evolved system that will be implemented in this project, reflecting the integration of these refinements.

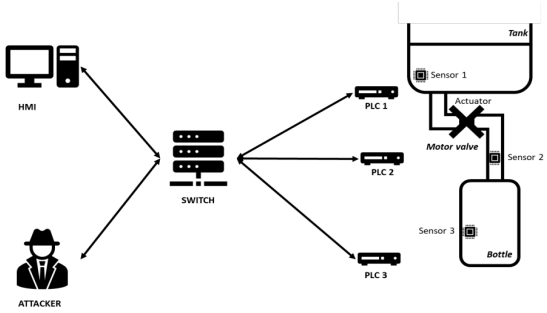

Figure 1: Digital Twin Design [1]

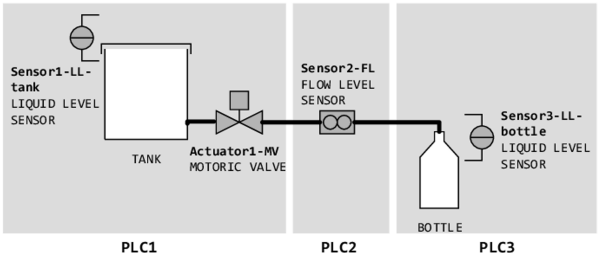

Figure 2: Filling Plant DT prototype [1]

Regarding the simulation aspect, our methodology involves the utilization of a digital twin replica representing an Industrial filling plant. Within this framework, each Programmable Logic Controller (PLC) acquires data from sensors and transmits it to a central Application Programming Interface (API) for thorough analysis and detection of anomalies. Simultaneously, individual PLCs preserve their respective logs within dedicated files, accessible solely to authorized administrators. The underlying network infrastructure is established using Mininet [8], affording a segregated environment that insulates the digital twin from external influences.

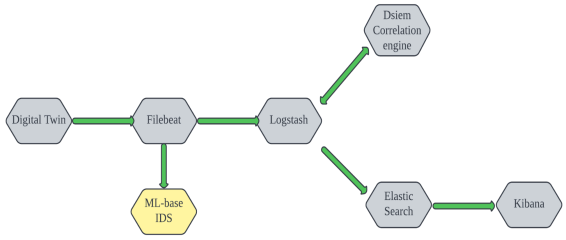

Figure 3: Initial system design by [1]

In the work presented by M. Eckhart and A. Ekelha [1], the network simulation involved the utilization of Docker and Kibana, with each node being represented by an individual container. *Figure 10*. In contrast, our project takes a distinct approach, employing a

standalone host environment without the incorporation of Docker containers. Within this context, all components operate on a single, autonomous host. Notably, an integral enhancement introduced in our project is the implementation of an Application Programming Interface (API). This API facilitates seamless communication between the frontend and backend services, streamlining and optimizing the exchange of information between these vital elements of the system architecture.

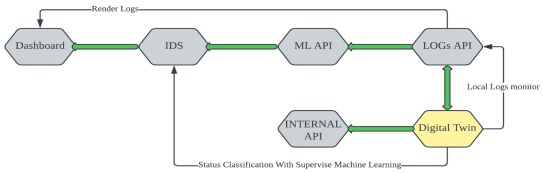

Figure 4: Enhanced System design [11]

The design of the API is rooted in the Python web framework Flask [9] (Grinberg, 2018), supplemented with flasgger [10] (flasgger, 2021) for documentation and local testing.

The API structure is divided into three categories: Model, Internal Logs, and Internal Communication. The Model section includes two endpoints for real-time predictive model outputs ("get_status") and specific file access ("plc_log"). In the Internal Logs segment, three routes ("get_data," "card_info," and "gen_table") dynamically facilitate data presentation and update mechanisms for the admin dashboard.

The Internal Communication API functions as a central server for PLCs, supporting real-time sensor data updates through its "get_value" and "set_value" endpoints. The project's user interface is anchored by the admin dashboard, featuring sections like real-time logs, terminal commands, network updates, intrusion alerts, system information, and user management.

For comprehensive details and practical demonstrations, kindly refer to the project's official GitHub repository: https://github.com/abuyusif01/dtss

3.2 Model Implementation

This section delves into the technical intricacies of our framework's predictive model implementations. Our ensemble comprises Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), Naïve Bayes (NB), and a Stacking ensemble technique, all orchestrated using the sklearn machine learning library [13].

Random Forest (RF) employs decision tree ensembles, harnessing bootstrapped data subsets to ensure robust predictions. In contrast, Gradient Boosting (GB) sequentially constructs weak learners, iteratively minimizing prediction errors. Logistic Regression (LR) delves into probabilistic modeling for binary outcomes. Naïve Bayes (NB) leverages Bayes' theorem to predict class probabilities. Our framework optimally integrates these algorithms through Stacking, synergizing their predictive capabilities.

At the core of our prototype lies Random Forest (RF), selected for its adaptability and competence in addressing diverse data distributions. The Stacking technique further heightens predictive accuracy and generalization. Facilitating seamless adoption, our models are harmoniously incorporated within Flask API framework [9], ensuring compatibility across platforms and self-hosted environments,

The user interface underscores cross-platform adaptability, employing HTML, CSS, and JavaScript for a unified codebase accessible via web browsers. JavaScript's intrinsic capabilities are harnessed to proficiently implement API helpers, thereby enhancing the overall efficacy of the prototype.

General Application workflow

- *Figure 5* shows how we periodically check the logs of PLC1, PLC2, and PLC3 Update all relevant databases with the latest log information
- Parse the logs through an intrusion detection model
- Send the results of the model back to the dashboard server
- PLC1 is responsible for deciding to close the actuator of PLC2 and PLC3 If an intrusion is detected.
- The system will alert the relevant parties and take appropriate action to mitigate the threat in our case its the admin – *Figure 9*

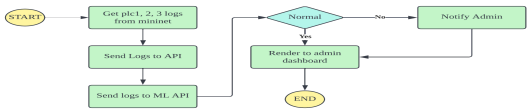

Figure 5: Application workflow [11]

3.3 Dataset Collection

The dataset was collected by manually monitoring the system logs and labeling the resulting data points. A variety of different types of attacks were then launched on the network and the resulting system logs were added to the dataset.

The resulting dataset was of sufficient size and quality for model training, with a total of 5000 data points. The dataset was preprocessed and cleaned to ensure that it was ready for use in training the models. The dataset was then split into training and test sets, with 80% of the data used for training and the remaining 20% used for testing.

The dataset includes a variety of different types of attacks and system logs, ensuring that the models trained on this dataset will be robust and capable of handling a wide range of scenarios.

3.4 Attack Implementation

Assuming the adversary knows the specific process of the ICSs, there's total of 3 different attack categories modeled.

1. Denial of Service: is an attempt by an adversary to flood a server with traffic to overwhelm the service. According to M. Zolanvari el al [14]. Dos is to be considered the biggest threat of ICSs. This attack is carried on by flooding the internal server with unnecessary traffic, hence leading to service disruption.
2. Command Line Injection: An attack involved in granting an attacker system command execution. In [15] Yao-bin stat- ed that when hackers or unknown threats are attacking ICSs, the first thing they tend to utilize is the command line injection, due to updated or unauthen- ticated protocols used in ICSs.
3. Man In the Middle: ICSs are interconnec- ted systems, Hence they share system status, setting values and updating values with peers in the same network. An adversary with the right access to the network can easily sniff the traffic and potentially analyze or in worse case be able to modify the traffic.

3.6 Model Training

A supervised machine learning approach was used to create a model using linear regression algorithms. The model was trained on Digital Ocean servers, which provided the necessary extra GPU for better performance. A total of 5 different algorithms were used: random forest, gradient boost, stacking, naive bayes, and logistic regression.

The training process was conducted using a dataset of size 50000, all models achieved good performance. The user has the option to use any of the trained models by specifying the model name as a parameter in the API request. The results of the model evaluation will be examined in section 3.7

3.7 Model Evaluation

Five models were developed and assessed in this project: random forest, gradient boost, stacking, naive bayes, and logistic regression. The dataset, sourced from log API, underwent preprocessing to prepare for model training. Employing a supervised machine learning methodology, the training procedure utilized a blend of linear regression algorithms.

The performance of the models is evaluated using three metrics: accuracy, precision, and recall. Accuracy measures – the proportion of correct predictions made by the model. Precision measures – the proportion of true positive predictions among all positive predictions made by the model. Recall measures – the proportion of true positive predictions made by the model among all actual positive cases.

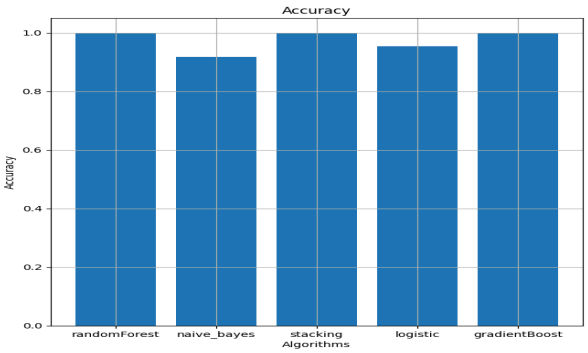

Figure 6: Model Accuracy Comparison [11]

Overall, all models achieved high accuracy, good precision and recall scores. However, Random forest, Gradient boost, and stacking model performed noticeably better than the other models. *Figure 6* shows an accuracy comparison between all trained models.

The default model for this project is the random forest model because it is faster than the other models.

However, if the user requires the maximum accuracy possible, the stacking model may be the best option.

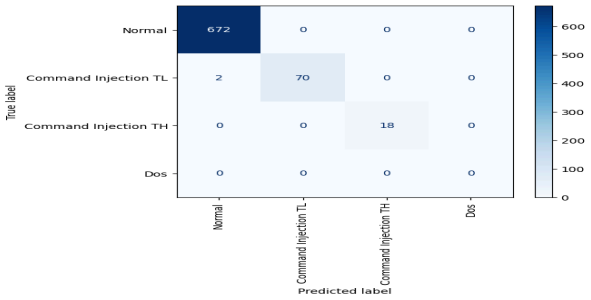

Figure 7: Confusion matrix Random Forest [11]

4. Dashboard Development

This section outlines the development of our dashboard, a crucial component of the prototype simulation. The dashboard serves as an administrative portal, facilitating real-time data visualization and interaction. Constructed using HTML, CSS, jQuery, and JavaScript, it seamlessly integrates with the backend infrastructure running on Express.js, while Flask handles API integration.

The dashboard offers dynamic data presentation, ensuring cross-platform compatibility with HTML and CSS. jQuery and JavaScript enhance user interaction and responsiveness. The backend, powered by Express.js, manages data flow between the dashboard and the digital twin simulation. Flask-constructed APIs enable efficient data exchange, bolstering real-time capabilities.

Integral to the dashboard is its connection to a MySQL database, facilitating data storage and retrieval. Real-time updates ensure the visualization aligns with the simulation's progress. Notably, the dashboard includes a notification system for administrators, enhancing situational awareness.

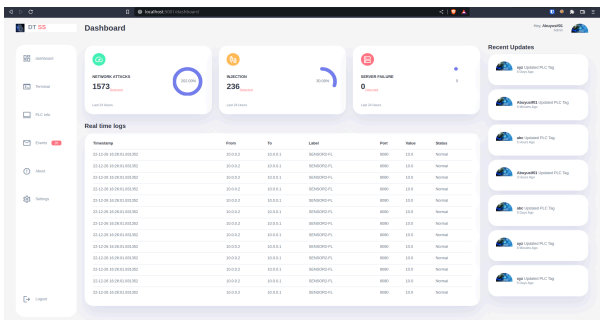

Figure 8: Admin Dashboard [11]

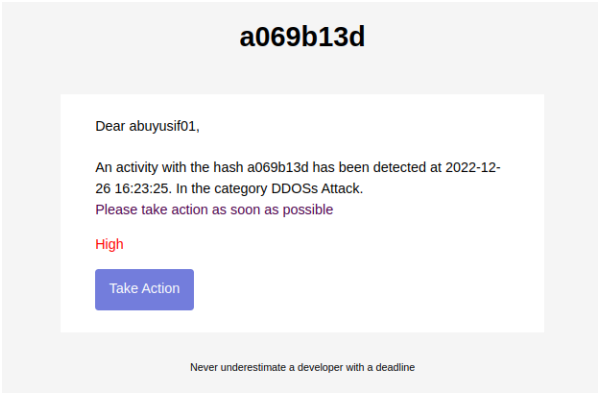

Figure 9: Notify email [11]

5. Conclusion and Future works

This paper presents a comprehensive exploration of a prototype simulation leveraging Digital Twin technology for enhanced security in Industrial Control Systems (ICS). The integration of predictive models within the simulation demonstrates its potential to proactively detect anomalies, offering an invaluable layer of defense. Through meticulous model implementation, encompassing Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), Naïve Bayes (NB), and Stacking techniques, the simulation showcases adaptability to various data distributions.

The seamless interaction between the digital twin and the actual ICS components, facilitated by a custom-built Application Programming Interface (API), establishes real-time monitoring and analysis capabilities. The developed prototype dashboard, constructed using HTML, CSS, jQuery, and JavaScript, serves as both a visual data representation tool and a multifunctional administrative portal. The integration of Flask and Express.js ensures efficient API interaction and seamless backend communication.

Incorporating a MySQL database further enriches the prototype's capabilities by enabling real-time data updates and storage. The dashboard's notification system empowers administrators with timely alerts, enhancing situational awareness. Our comprehensive approach aligns with the ever-evolving demands of ICS security, offering a versatile solution that bridges predictive modeling, digital twin technology, and real-time monitoring.

This work contributes to the broader discourse on securing Industrial Control Systems, presenting a tangible prototype that bridges the gap between

simulation and real-world applicability. As the field continues to advance, the insights gained from this research hold promise for future advancements in ICS security, underpinned by the innovative fusion of digital twin technology and predictive analytics.

6. References

[1] M. Eckhart and A. Ekelhart, "A specification based state replication approach for digital twins," ser. CPS-SPC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 36–47. [Online]. Available: https://doi.org/10.1145/3264888.3264892

[2] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 669–680, 2020. DOI: 10.1109/TII.2019.2938885 https://lup.lub.lu.se/search/files/81503765/IEEETransIndInfGehrmannGunnarsson.pdf

[3] M. Dietz, M. Vielberth, and G. Pernul, "Integrating digital twin security simulations in the security operations center," in Proceedings of the 15th International Conference on Availability, Reliability, and Security, ser. ARES '20. New York, NY, USA: Association for Computing Machinery, 2020. DOI: 10.1145/3407023.3407039. ISBN 9781450388337. [Online]. Available: https://doi.org/10.1145/3407023.3407039

[4] Varghese, S. A., Ghadim, A. D., Balador, A., Alimadadi, Z., & Papadimitratos, P. (2022, March). Digital Twin-based Intrusion Detection for Industrial Control Systems. In 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (pp. 611-617). IEEE. http://kth.diva-portal.org/smash/get/diva2:1637561/FULLTEXT01.pd

[5] Xu, Qinghua, Shaukat Ali, and Tao Yue. "Digital twin-based anomaly detection in cyber-physical systems." 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST). IEEE, 2021. https://www.simulamet.no/sites/default/files/publications/files/digital_twin_icst_2_1.pdf

[6] Empl, P., Schlette, D., Zupfer, D., & Pernul, G. (2022, August). SOAR4IoT: Securing IoT Assets with Digital Twins. In Proceedings of the 17th International Conference on Availability, Reliability and Security (pp. 1-10). https://dl.acm.org/doi/abs/10.1145/3538969.3538975

[7] F. Akbarian, E. Fitzgerald and M. Kihl, "Intrusion Detection in Digital Twins for Industrial Control Systems," 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2020, pp. 1-6, doi: 10.23919/SoftCOM50211.2020.9238162. https://ieeexplore.ieee.org/document/9238162

[8] Mininet creates a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native), in seconds, with a single command http://mininet.org/

[9] Grinberg, M. (2018). Flask Web Development. O'reilly Media, Incorporated. https://flask.palletsprojects.com/en/2.2.x/

[10] flasgger. (2022, August 23). flasgger/flasgger. GitHub. https://github.com/flasgger/flasgger

[11] DTSS (2022, December, 1) dtss/dtss Github https://github.com/abuyusif01/dtss

[12] Digital Ocean: Cloud hosting provider https://www.digitalocean.com/

[13] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp https://scikit-learn.org/stable/

[14] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6822–6834, 2019 https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8693904

[15] Liu, K., Wang, J. Y., Wei, Q., Zhang, Z. Y., Sun, J., Ma, R. K., & Deng, R. L. (2021). HRPDF: A Software-Based Heterogeneous Redundant Proactive Defense Framework for Programmable Logic Controller. *Journal of Computer Science and Technology*, *36*(6), 1307-1324. https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=7927&context=sis_research

[16] International Business Machines Corporation (IBM). What is a digital twin? www.ibm.com/my-en/topics/what-is-a-digital-twin