Enhancing Anomaly Detection in Industrial Control Systems through Supervised Learning and Explainable Artificial Intelligence

Dhruv G. Bhatt[1], Parshad U. Kyada[1], Rajkumar Singh Rathore[2], M. K. Nallakaruppan[3,*], Faisal Mohammed alotaibi[4], Rutvij H. Jhaveri[1,*]

[1] Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar 382007, India

[2] Department of Computer Science, Cardiff School of Technologies, Cardiff Metropolitan University, Llandaff Campus, CF5 2YB Cardiff, U.K

[3] Balaji Institute of Modern Management, Sri Balaji University, Pune, Pincode-411033, India

[4] Department of Computer Science, Prince Sattam Bin Abdulaziz University, Al-Kharj, Riyadh 16278, Saudi Arabia

Emails: dhruv.bhatt.info@gmail.com; parshadkyada2003@gmail.com; rsrathore@cardiffmet.ac.uk; Nallakaruppan.K@bimmpune.edu.in; faisal.alotaibi@psau.edu.sa; rutvij.jhaveri@sot.pdpu.ac.in

Abstract

This paper addresses industrial control security (ICS) security, focusing on utilizing intrusion detection systems (IDS) to protect ICS networks. It suggests the use of a Measurement Intrusion Detection System (MIDS) over a Network Intrusion Detection System (NIDS), directly analyzing measurement data to detect unseen activities. Training MIDS requires a labeled dataset of various attacks, and a hardware-in-the-loop (HIL) system is used for safer attack simulations. The main aim is to assess MIDS performance through machine learning (ML) on this dataset. Explainable artificial intelligence (XAI) is integrated for transparency in decision-making. Various ML models, such as random forest, achieve high accuracy in detecting anomalies, notably stealthy attacks, with a receiver operating curve (ROC) of 0.9999 and an accuracy of 0.9795. This highlights the importance of machine learning in securing ICS, supported by XAI's explanatory power.

Keywords: Hardware in the Loop (HIL) System; Intrusion Detection; Machine Learning; Real-time Attack Detection; Stealthy Attacks

1 introduction

An Industrial Control System (ICS) is a computerized system used in industries to oversee industrial processes efficiently incorporating sensors, computers and software. ICS is used across industries such as manufacturing, power generation, transportation, and water treatment. Common types of ICS systems include Supervisory Control And Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and Programmable Logic Controllers (PLCs).[1] SCADA systems primarily monitor large-scale systems spread across locations, while DCS systems are more commonly used for smaller, more localized systems. PLCs control individual pieces of equipment or processes. An average ICS comprises numerous control loops along with Human-Machine Interfaces (HMIs), remote diagnostics, and maintenance tools.[2] These components are interconnected through network protocols, enabling communication among themselves and with the external world.

Network protocols are employed in ICS networks to ensure the security of network traffic, utilizing authentication, data encryption, and message integrity techniques. Conventional network security mechanisms are in

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

place but may not always be sufficient to safeguard ICS networks from all forms of suspicious activities and attacks.[3] This is because these mechanisms are typically designed to protect against general threats, such as unauthorized access and data theft. Consequently, ICS becomes a prime target for cyber-attacks. Therefore, the implementation of security measures specifically tailored to defend against these threats is crucial. These measures may include the use of Intrusion Detection or Prevention systems (IDS/IPS), often maintained by keeping the software up-to-date and by training employees involved in security monitoring.

In this paper, the proposed problem is addressed through an Intrusion Detection System (IDS), which monitors network traffic to identify symptoms of malicious activities.[4] When suspicious activity is detected by an IDS, an alert is generated. IDS utilizes various methods to detect intrusions, including Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS).[5] SIDS uses a database of known attack signatures to detect malicious activity. On the other hand, AIDS does not use a database with known attack signatures. Instead, it creates a baseline of normal network activity and then looks for deviations from that baseline to detect abnormalities.[6]

IDSs are security systems that monitor network traffic to identify potential threats. Network Intrusion Detection Systems (NIDS) are a type of IDS specifically focused on detecting malicious activity within network traffic. Machine learning methods have been employed by many researchers to train NIDS models responsible for detecting attacks in network traffic,[7],[8].[9] However, NIDS has limitations. Encrypted data packets, faked IP packets, and regular false positive alerts can render NIDS ineffective. If the above-said activity is encrypted, or if the attacker uses a forged IP address, it will not be detected by the NIDS.[10] Furthermore, false positive alerts are often produced by NIDS, leading security analysts to waste time investigating false alarms.[7] On the other hand, Measurement Intrusion Detection Systems (MIDS) are a type of IDS that monitors and measures data regarding patterns of malicious activity. MIDS does not monitor network traffic like NIDS. Instead, measurement data within the system is directly inspected by MIDS. This allows MIDS to detect malicious activities that may not be visible in network traffic, such as changing sensor setpoints or injecting fake data measurements into ICS network levels.[3]

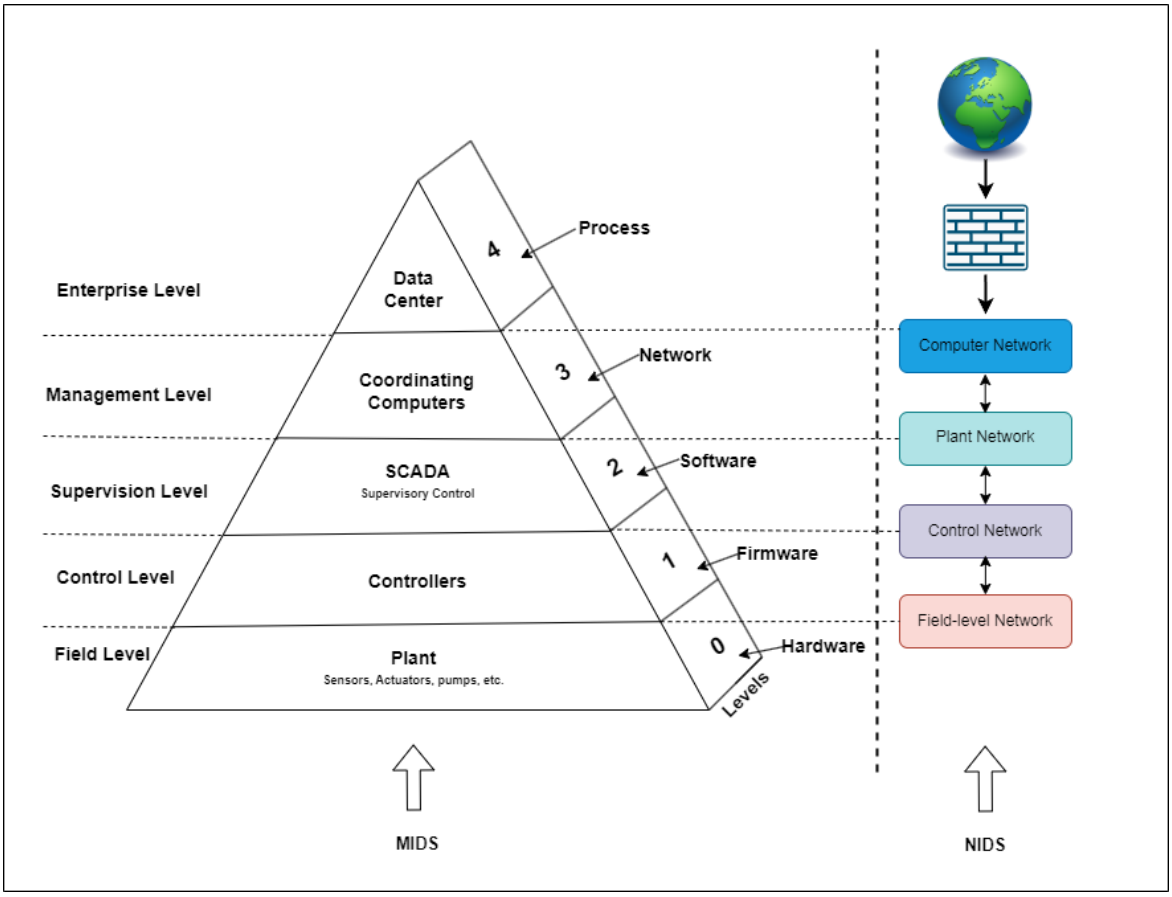


Figure 1: MIDS and NIDS in Industry Control System

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

In most studies, machine learning models are trained using normal activity datasets, limiting MIDS to comparing incoming data solely against normal behavior. However, this approach fails to detect stealthy attacks mimicking normal behavior. A solution to this, a labeled dataset that includes various attack types is necessary for training MIDS to detect malicious activities. However, this is still a challenging task, as it requires access to real-world attack data and this can be risky, as it could lead to a system failure or irreparable damages,[3].[7] To mitigate these risks, a hardware-in-the-loop (HIL) system is employed to simulate attacks without actually injecting them into the live system.[11] In recent years, HIL for power systems has been used to verify the stability, operation, and fault tolerance of large-scale electrical grids.

The main goal of this paper is to examine MIDS performance by training machine learning algorithms on a labeled dataset of both malicious and normal activities. Decisions made by machine learning algorithms will also be explained using Explainable Artificial Intelligence (XAI).[12] An experimental setup was developed, to evaluate the effectiveness of fault detection by monitoring measurement data in an ICS. The setup involved a power generation testbed, with sensor values recorded over several days. Various attack scenarios were injected into the system to generate a labeled dataset. Findings indicate that machine-learning algorithms achieved high accuracy in detecting malicious activities. Additionally, the study demonstrated that the decisions made by machine learning algorithms can be explained using XAI. Recent advancements further support this approach, as the integration of Transfer Learning and Explainable AI in IoT applications has been shown to enhance diagnostic accuracy and trustworthiness.[13] Overall, the study shows that machine learning can be a valuable tool for detecting malicious activities in ICS. However, the effectiveness of machine learning for ICS security can be improved with the use of XAI.

Overall, the following contributions have been made to the attack detection domain by this work:

- A modern approach has been introduced that can be incorporated as a second layer of defense mechanism with NIDS for intrusion detection using measurement data, thereby improving the security of the ICS system.
- The proposed approach is more reliable than previous approaches because it is trained on a dataset of real-world attacks. The HIL-based augmented security dataset has been applied to training a supervised machine learning model to detect intrusions and anomalies in ICS.
- Stealthy attacks can be detected by the proposed approach while posing no harm to the actual system because labeled data obtained from the HIL testbed is leveraged, allowing the model to be trained on a dataset of real-world attacks without having to put the actual system at risk.
- The results have been empirically evaluated, and the performance of the different machine learning algorithms for the detection of stealthy attacks in ICS has been assessed. The random forest algorithm was found to be the most effective, achieving the best accuracy and the lowest false positive rate.

2 Literature Review

In this section, relevant literature is reviewed to identify research gaps for our study. A recent implementation of threat detection for ICS is reviewed with an overview of the safety and security of ICS as proposed in the paper.[14] The line between safety and security for ICS is clearly defined, and the various approaches to industrial facility design and risk assessment are classified.[15] An overview of the state-of-the-art cyber security risk assessment of SCADA systems is also provided. However, both of these works lack a survey of machine learning techniques for ICS security. Machine learning is increasingly being used to improve the security of industrial systems. A study from 1997 highlights the potential of machine learning methods in enhancing security for ICS.[16] It is suggested that autonomous machine learning systems offer simpler and more systematic security solutions compared to traditional methods, providing fresh approaches to the challenge of building and controlling future industrial systems while preserving a decent level of security.

In,[22] machine learning algorithms are subjected to data analysis, addressing challenges such as volume, veracity, diversity, and validity. A classification framework categorizing methods for data analysis is presented,

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

Table 1: Previous work on Anomaly Detection in ICS.

Reference	Datasets Used	Models	Validation	Objectives	Keywords	Limitations
[17]	MNIST-C, MVTec	KDE, Autoencoder, Deep One-Class	Validation set with outliers	Propose an XAI framework for anomaly detection, identify Clever Hans effect	Anomaly detection, explainable AI, Clever Hans effect	Limited to unsupervised models
[18]	Manually Collected	One-Class SVM, Variational Autoencoder	Cross-validation	Detect anomalies in industrial control systems	Anomaly detection, Supervised learning, XAI	Limited to SVM, no comparison with other models
[19]	CICIDS2017	autoencoders artificial neural network architecture (ANN)	Accuracy, F1-score, AUC, G-means	Compare the performance of different models for anomaly detection	Anomaly detection, Supervised learning	Limited to the KDD Cup 1999 dataset
[20]	Manually Collected	Random Forest	Hold-out validation	Improve accuracy of anomaly detection in industrial control systems	Anomaly detection, Supervised learning, XAI	Limited to a specific type of anomaly, may not detect other types of anomalies
[21]	Manually Collected	Decision Tree	Leave-one-out validation	Anomaly detection in Industrial Control Systems using decision trees	Industrial control systems, anomaly detection, decision trees	Limited to a specific type of model, may not generalize to other models

based on factors like data type, algorithm, tasks, and assumptions. Furthermore, both the advantages and disadvantages of machine learning in data analysis are discussed, highlighting its potential in IoT application development. The author concludes with a case study using the Support Vector Machine (SVM) model in traffic data from Aarhus Smart City, demonstrating machine learning's potential in extracting relevant information from IoT sensor data.

In,[23] the application of machine learning algorithms for detecting attacks in ICS is explored. Six machine learning algorithms, namely Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), and others, are evaluated and shown to effectively identify attacks. Challenges associated with training machine learning models due to the infrequent nature of ICS attacks are addressed. Despite these obstacles, machine learning shows promise for enhancing the accuracy and dependability of attack detection systems for ICS. Various new technologies and methodologies, such as AI, neural networks, blockchain, and XAI, are being experimented with by researchers. A unique approach utilizing unsupervised deep neural networks, particularly convolutional neural networks (CNN), is proposed, showing effectiveness in detecting cyber attacks on ICS using the Secure Water Treatment testbed dataset.[24] Recent progress, in detecting intrusions in Wireless Sensor Networks (WSN) is also applicable to securing Industrial Control Systems (ICS). For example the WOGRU IDS system suggested in reference[25] presents a combined learning approach that fine tunes the parameters of a Gate Recurrent Unit (GRU) network using the Whale Optimization Algorithm. This method significantly enhances the precision and efficiency of identifying types of attacks such, as flooding, blackhole and grayhole attacks in WSN IoT settings. Additionally, the application of blockchain technology in ICS is discussed, known for its distributed ledger offering immutability, transparency, and distribution. However, challenges with blockchain technology in ICS, such as interoperability and scalability concerns, are highlighted.[26]

Explainable Artificial Intelligence (XAI) enhances the transparency and reliability of machine learning models. XAI techniques, such as LIME and SHAP, are utilized in the proposed study,[27],[28].[29] LIME explains model predictions by creating a model that approximates the decision boundaries of the original model, while SHAP assigns each feature an importance score, indicating its significance in influencing model predictions. Incorporating LIME and SHAP leverages both the transparency and reliability of machine learning models.

2.1 Research Gap

- The strength of the supervised learning for anomaly detection in ICS to find abnormal behaviors, with critical failures has been researched. However, there is a considerable gap in the area with the absence of thorough research on the integration of XAI models to improve the interpretability of the supervised anomaly detection models.
- In the context of critical industrial processes, where trust, transparency, and human-understandable decision-making are critical, the use of XAI models for the explanations of anomaly detection remains unexplored.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

- Analysis of the research gap is critical for developing a more robust anomaly detection framework that not only produces accurate predictions but also provides operators and engineers with actionable insights into the observed anomalies. This eventually leads to more reliable and secure ICS.

3 Methodology

In this section, a procedure for detecting attacks in ICS will be discussed. The discussion will begin by identifying the attacks that can happen in ICS. Then the approaches for detecting these attacks using machine learning techniques will be illustrated. This approach includes gathering data from the ICS, choosing the features from the dataset, and training machine learning models based on these selected features. Later the data can then be classified and potential attacks are identified.

3.1 Attack Description:

Anomaly detection is a crucial technique utilized to identify data points that deviate from established patterns, with applications ranging from malfunction detection to intrusion identification in ICS. Within ICS, anomalies may indicate malfunction in system components like sensors, as well as unauthorized access or modification of system data, including intentional cyber attacks such as denial of service attacks or data breaches. This paper explores anomaly detection using MIDS, which leverages sensor measurements, control signals, and system logs to detect anomalies. MIDS has proven effective in detecting both malfunctions and intrusions within ICS. For instance, deviations in sensor readings due to malfunctioning sensors are promptly identified, while intrusions are detected through abnormalities in access patterns and system data modifications. Advantages in anomaly detection within ICS are offered by SCADA systems, enabling the identification of activities potentially overlooked by (NIDS) at the network level. SCADA systems, commonly used for monitoring and managing operations, gather various sensor data, including parameters like temperature and pressure. Malfunctions or simple attacks manipulating measurement data within the system pose minimal challenges to MIDS. However, concerns arise regarding the effectiveness of MIDS against stealthy attacks, where attackers manipulate sensor measurements or control signals to mimic system behavior and avoid detection. This paper evaluates the performance of MIDS not only in detecting malfunctions but also in identifying and detecting these stealthy attacks.

3.2 Data Analyzing:

IDS is a security system designed to monitor network traffic and activities for detecting and responding to malicious behavior. The main objective of IDS modeling is to develop a machine-learning model, capable of distinguishing between normal and abnormal activities based on identifiable patterns found in system logs. One significant challenge arises in real-world scenarios, where instances representing normal conditions outnumber those indicating abnormal events, creating an imbalanced dataset. This imbalance may cause bias towards the majority class, leading to decreased accuracy in identifying instances from the minority class. To tackle this problem, techniques to minimize the impact of imbalanced datasets are often employed by researchers and professionals in the field of IDS modeling. These techniques include methods at the data level, the algorithmic level, and a combination of both. In our study, a data-level approach was utilized as a step to balance the dataset. Under-sampling is a data-level method aimed at reducing the number of instances from the majority class while maintaining a balanced class distribution.[30] Random Under Sampling (RUS) is a technique used to achieve this balance, involving the selection of instances from the majority class and removing them until the desired ratio between the classes is achieved. For example, if a dataset contains 100 instances with 400 negative instances, RUS would remove 300 instances to achieve an equal balance of positive and negative samples.[31]

In this research paper, the issue of imbalanced data is addressed by utilizing the RUS technique to normalize the dataset. Additionally, the new dataset ensures the distribution of normal and abnormal data during both the training and testing process.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

3.3 Feature Engineering:

In the field of ICS, data from numerous sensors is collected, providing valuable input for training machine learning models to anticipate and prevent cyber attacks. However, challenges arise in training and implementing these models due to the sheer volume of sensor data. Feature selection plays a crucial role in improving model performance by identifying significant features within the dataset, reducing noise, and enhancing accuracy. This is particularly important in ICS, where reducing the training and implementation time is crucial for real-time predictions to prevent cyber attacks. Feature selection approaches can be categorized into four groups: the filter approach, wrapper approach, embedded approach, and hybrid approach.[32] Among these, the filter method is preferred as it selects features based on performance measures without considering data modeling. Correlation analysis, particularly using Pearson correlation, is a useful criterion for feature selection, especially for non-categorical data,[33] which can be described as

$$\text{Corr}(i) = \frac{\text{cov}(m_i, n)}{\sqrt{\text{var}(m_i) \cdot \text{var}(n)}} \tag{1}$$

Where m_i is the ith feature, n is the target label, and cov() and var() represent the convenience and the variance functions, respectively. Corr(i) indicates the Pearson correlation technique, which shows the correlation between the ith feature and the corresponding target. Features with high correlation with the target variable should undergo feature selection with a maximum threshold, while the minimum correlation value for a feature must be considered for elimination to maintain simplicity in model evaluation and avoid redundancy.[34] Instead of dropping correlated features, they can be combined into a single feature to reduce multicollinearity, a process known as feature aggregation or feature engineering.

In our dataset, a challenge arises from features being measured using different units, which can hinder machine learning algorithms' ability to learn from the data. It is crucial to standardize the data before training the model to address this issue. Standardization involves transforming the values of each feature to a specific range. One such method is MinMaxScaler, which transforms feature values to a scale ranging from 0 to 1.[35] Equation (2) describes MinMaxScaler, where p_i is the ith feature, p_min and p_max are the minimum and the maximum values of the feature among the experiments, respectively. In addition, $p_i(scaled)$ indicates the scaled value for ith feature. Standardizing the data ensures that every feature contributes equally to the learning process and prevents any bias towards features, during the learning process.

$$p_{i(scaled)} = \frac{pi - min(p)}{max(p) - min(p)} \tag{2}$$

3.4 Machine Learning Models:

In supervised anomaly detection, ICS data on both normal and abnormal behavior is gathered. This data is then used to train a model that can determine whether new data falls into the abnormal category. The model becomes familiar with both regular behavior (the "normal class") and unusual behavior (the "anomaly class"). When new data is received, it is compared against the predictions of the model. If the new data significantly deviates from the model's expectations, it is considered to be an anomaly.

In this study, various machine learning algorithms were employed to train a model for detecting anomalies. The training process involved using a labeled dataset where each data point was categorized as either "normal" or "attack". This enabled the model to learn patterns and identify anomalies when data points don't fit these patterns.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

3.5 Model Evaluation metrics:

- **Confusion matrix:**

  The confusion matrix is a table that summarizes how well a machine-learning model performs on a given set of test data. It is commonly employed to evaluate the performance of classification models, which strive to predict labels for each input.

  The confusion matrix displays the count of True Positives (TR_P), True Negatives (TR_N), False Positives (FL_P), and False Negatives (FL_N) generated by the model when tested. These values can be used to compute metrics that determine the model's performance, including accuracy, precision, recall, and f1 score.

  - **Accuracy:** It refers to the percentage of instances that the model correctly classifies. It is calculated by dividing the number of positives and true negatives by the overall number of instances.

$$\textbf{Accuracy} = \frac{TR_P + TR_N}{TR_P + TR_N + FL_P + FL_N} \tag{3}$$

  - **Precision:** It represents the percentage of instances classified as positive that are truly positive. It is calculated by dividing positives by the sum of positives and false positives.

$$\textbf{Precision} = \frac{TR_P}{TR_P + FL_P} \tag{4}$$

  - **Recall:** It measures the percentage of instances that are correctly classified. It is calculated by dividing positives by the sum of positives and false negatives.

$$\textbf{Recall} = \frac{TR_P}{TR_P + FL_N} \tag{5}$$

  - **F1 Score:** It is a weighted average of precision and recall. It is computed using their harmonic mean.

$$\textbf{F1 Score} = \frac{2 \cdot TR_P}{2 \cdot TR_P + FL_P + FL_N} \tag{6}$$

- **Receiver Operator Characteristic (ROC):** The ROC curve shows the stability and performance of the classification model by comparing the rate of true positive (TPR) with false positive (FPR). TPR represents the rate of positive cases that are accurately classified while FPR indicates the rate of negative cases that are wrongly classified. The ROC curve helps assess the balance between TPR and FPR.[25]

- **Area Under the Curve (AUC):** The AUC is the area under the ROC curve. It is a measure of the overall performance of the model. A higher AUC indicates that the model is better at distinguishing between positive and negative instances.

4 Experimental Setup

4.1 ICS Testbed:

An ICS testbed is a physical or virtual environment that is used to simulate an ICS. As shown in 2, the test-bed system consists of four processes: a boiler process, a turbine process, a water treatment process, and a HIL simulator.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

- **Boiler Process:** The boiler process involves water-to-water heat transfer at low pressures and moderate temperatures. The boiler process used five controllers (a level controller, pressure controller, temperature controller, flow-rate controller, and cooling controller) to regulate the boiler pressure, temperature, and water level.
- **Turbine process:** The turbine process involves three controllers (speed control and over-speed and over-vibration trips) that rotate the turbine and prevent over-speed and over-vibration.
- **Water Treatment Process:** The water treatment process involves a level controller to manage the level control pump (LCP) and the level control valve (LCV). The level controller is responsible for pumping and releasing water between the upper and lower reservoirs.
- **HIL Simulator:** The HIL Simulator simulates two generators (i.e., a steam turbine power generator and a pumped-storage hydropower generator) and one power grid model for electrical load.

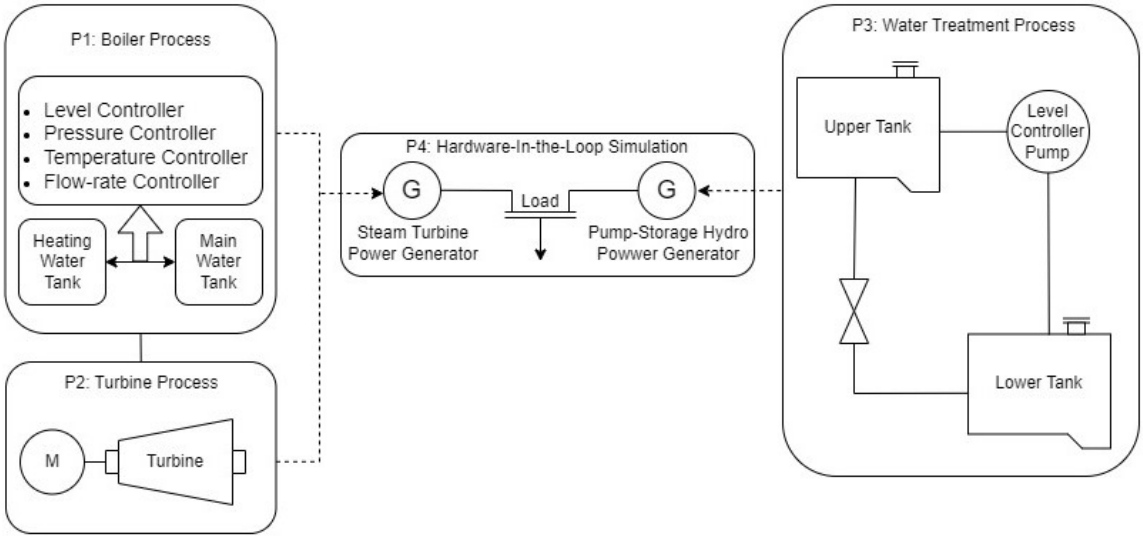


Figure 2: HIL-based augmented ICS.

4.2 Dataset:

The dataset used in this paper is from a HIL-based augmented ICS security (HAI) available at Kaggle. The testbed dataset is built by collecting measurements of 86 sensors every second for five days. 52 attacks were conducted, including 42 attack primitives and 10 combinations of attacks designed to perform two attack primitives simultaneously. All attack scenarios in the viewpoint of a feedback control scheme were configured based on four types of variables, namely the setpoints (SPs), process variables (PVs), control variables (CVs), and control parameters (CPs). The attacks are stealthy and cannot be detected easily by the conventional NIDS.

5 Result and Discussion

The MIDS method was evaluated using the HAI dataset for anomaly detection with a machine learning approach that provides explanations for its predictions (XAI). The Python programming language is applied to train and test the machine learning algorithms. The steps taken to generate the model are shown in Figure 3, with the best model chosen from the pool of models used for MIDS classification.

A feature selection process selects the most important features from the dataset. Correlation metrics are used to identify the features that were most strongly correlated with the target variable, which indicates the presence

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

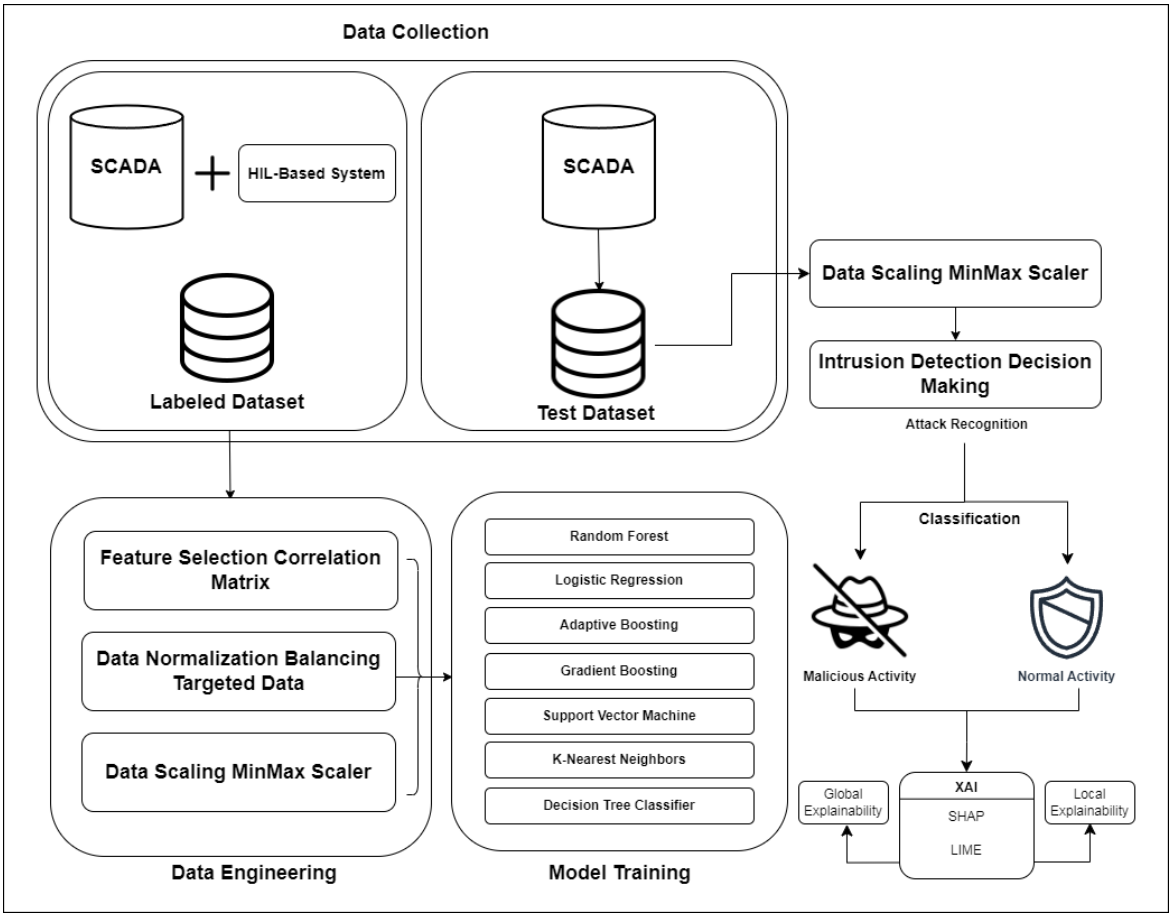
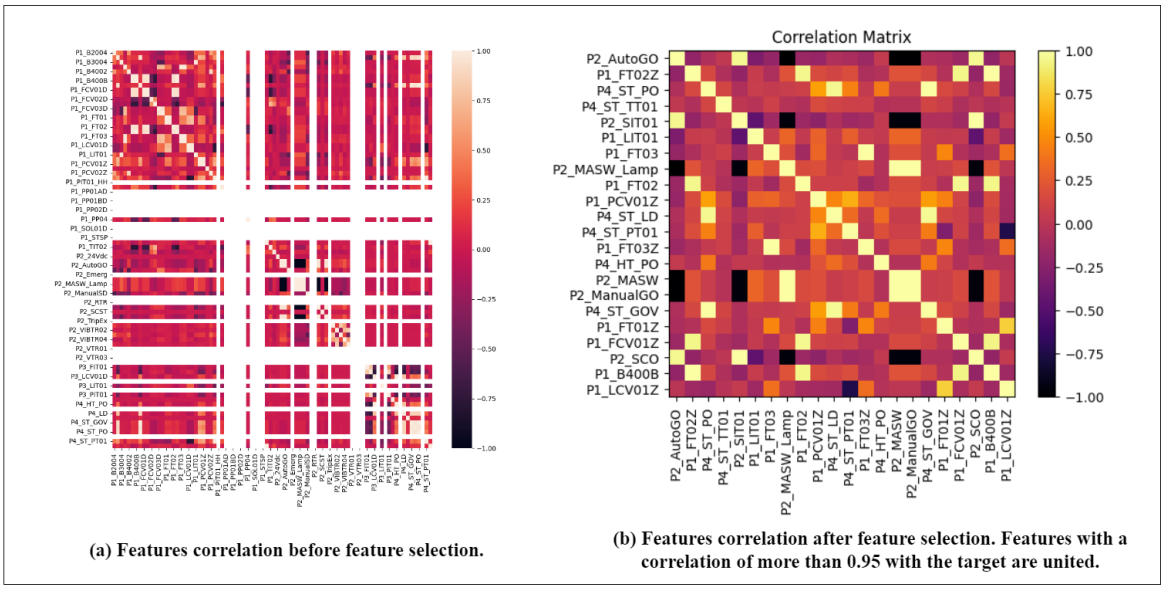


Figure 3: End-to-End Process Flow of MIDS in ICS with Enhanced Explainability for HAI Dataset

or absence of an anomaly. Out of 88 features, 65 are selected through this process. Figure 4a shows the correlation matrix before the feature selection, and Figure 4b shows the correlation matrix after the feature selection, respectively.



(a) Features correlation before feature selection.

(b) Features correlation after feature selection. Features with a correlation of more than 0.95 with the target are united.

Figure 4: Feature selection using a correlation metric.

DOI: https://doi.org/10.54216/JCIM.150125
Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

The imbalance of the target data in intrusion detection training datasets can pose a challenge for machine learning algorithms. Under-sampling is employed to achieve balance within the dataset by randomly eliminating instances from the majority class. This approach aids in mitigating the bias present in the dataset, thereby enhancing the efficiency of the trained models. The target distribution in the dataset before and after normalization is depicted in Figure 5.

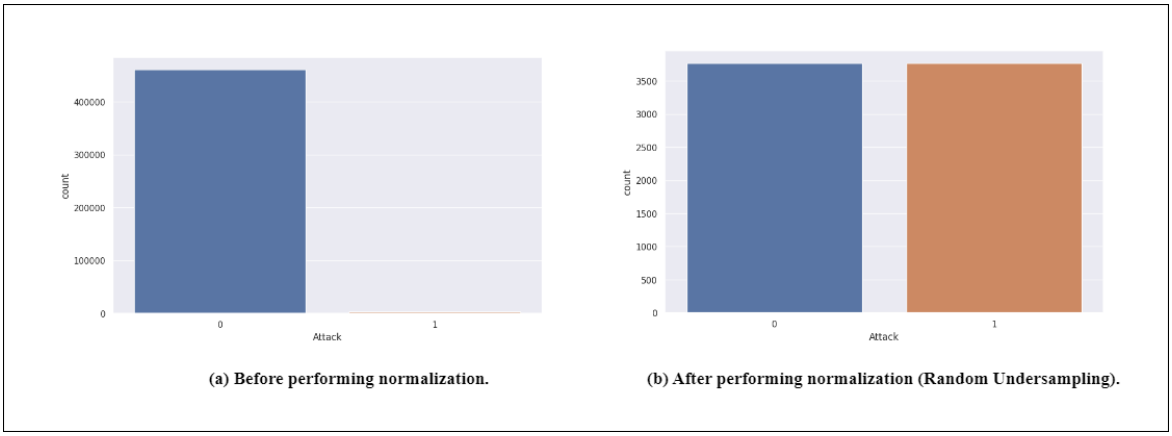


Figure 5: Normal and abnormal conditions' distribution.

Supervised classification models implemented include K-nearest neighbors (KNN), Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), Adaptive Boosting (AB), Support Vector Machine (SVM), and Decision Tree Classifier (DTC). Accuracy and time complexity are crucial factors when evaluating the performance of these models. The confusion matrix in Figure 6 is used to evaluate accuracy, precision, recall, f1-score, and other parameters.

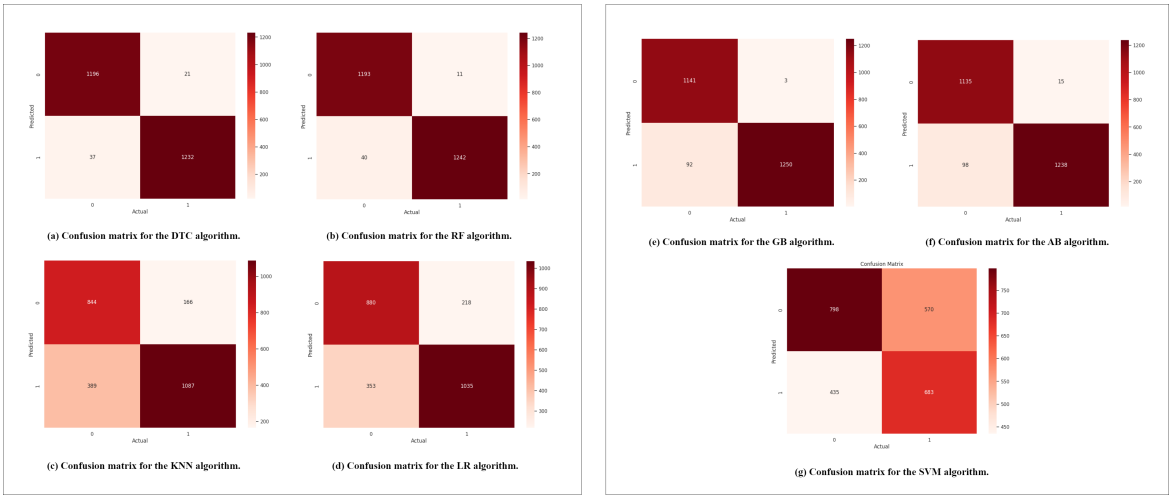


Figure 6: Confusion matrices.

The Random Forest algorithm outperformed all others in detecting anomalies in the dataset with an accuracy of 0.9795 and a ROC of 0.999. Conversely, the SVM algorithm had the lowest accuracy in predicting anomalies. The results indicate that MIDS can be considered a reliable solution to the anomaly detection problem. Using measurement data from SCADA to detect attacks can be seen as a new layer of protection in ICS, complementing NIDS for enhanced system reliability, especially against stealthy attacks. Table 2 shows the confusion matrices.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

Table 2: Model Evaluation Metrics

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree Classifier	0.9767	0.9708	0.9832	0.9770
Random Forest	0.9795	0.9688	0.9912	0.9799
K-Nearest Neighbors	0.7767	0.7364	0.8675	0.7966
Logistic Regression	0.7703	0.7457	0.8260	0.7838
Gradient Boosting	0.9618	0.9314	0.9976	0.9634
Adaptive Boosting	0.9545	0.9266	0.9880	0.9564
Support Vector Machine	0.5957	0.6109	0.5451	0.5761

In ICS, stealthy attacks at the foundational level often involve imitating normal behavior to evade detection and deceive protection systems. Building a dataset that includes stealthy attacks is a challenging task. A real-life dataset capturing stealthy attacks on sensor data measurements has been made available in the dataset used, obtained using hardware-in-the-loop (HIL) systems.

Explainable AI models are applied in the proposed work to study the quantum of feature weights to understand their impact in determining the target. Two such XAI models applied are Local Interpretable Model Agnostic Explainer (LIME) and SHAP Additive Explainer (SHAP).

5.1 LIME

LIME is a local surrogate model that explains a single instance in a dataset with the respective other features. The pictorial representation of the prediction of the lime is presented in Figure 7. The feature weights and their nature are depicted by this graph. This instance is towards the classification of an attack prediction and all the features correspond to the same nature. The features are positive towards the prediction with almost equal weights in the range of 0.1 to 0.3. The accuracy score is 0.96 for the prediction of the instance as per the local surrogate approximation. The nature and the importance of the features are presented in Figure 8. AutoSD and FCV022 defects are negative toward the prediction. The rest of the defects are positive towards the prediction. The LIME is a local surrogate model that performs the approximation of features based on the lasso regressor and determines the impact of the features on the prediction of a local instance.

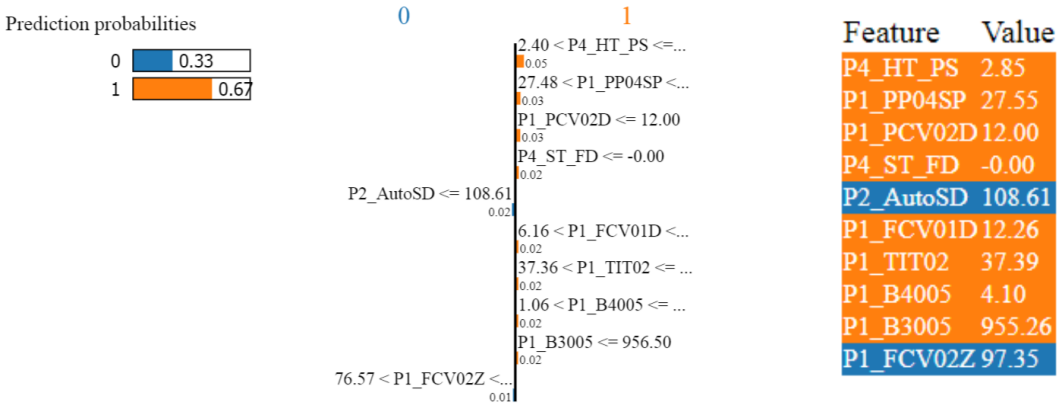


Figure 7: Lime Notebook for feature weight and prediction score approximation

DOI: https://doi.org/10.54216/JCIM.150125
Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

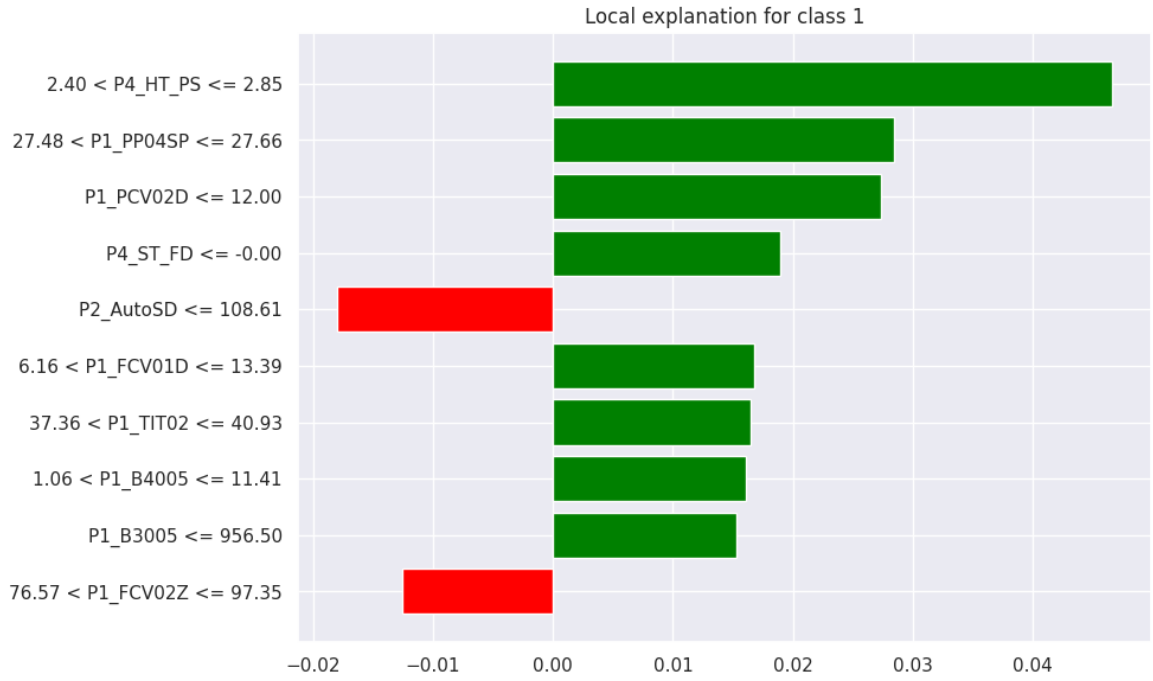


Figure 8: Feature importance and summary using LIME

5.2 SHAPELY

The SHAPELY additive explainer is a value-based explainer that determines the magnitude of the feature importance in determining the target with precision. This model has a substantial amount of plots both with local and global surrogates. The proposed work provides the most important plots, such as summary and decision plots. The Shapley value is the average expected marginal contribution of a feature across all possible combinations of features.

$$f(x') = \varphi_0 + \sum_{i=1}^{N} \varphi_i x'_i \tag{7}$$

Given g the explanation model. $z' \in \{0,1\}^M$ are the simplified features that describe the presence of the interested feature in the feature's combination with z' = 0 means the interested feature is absent in the combination and z' = 1 signifying the feature are present. M is the maximum coalition size and $\varphi_j \in R$ is the Shapley value for a feature j. The formula for the Shapley value is:

$$\varphi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1!)}{p!}(val(S \cup x_j) - val(S)) \tag{8}$$

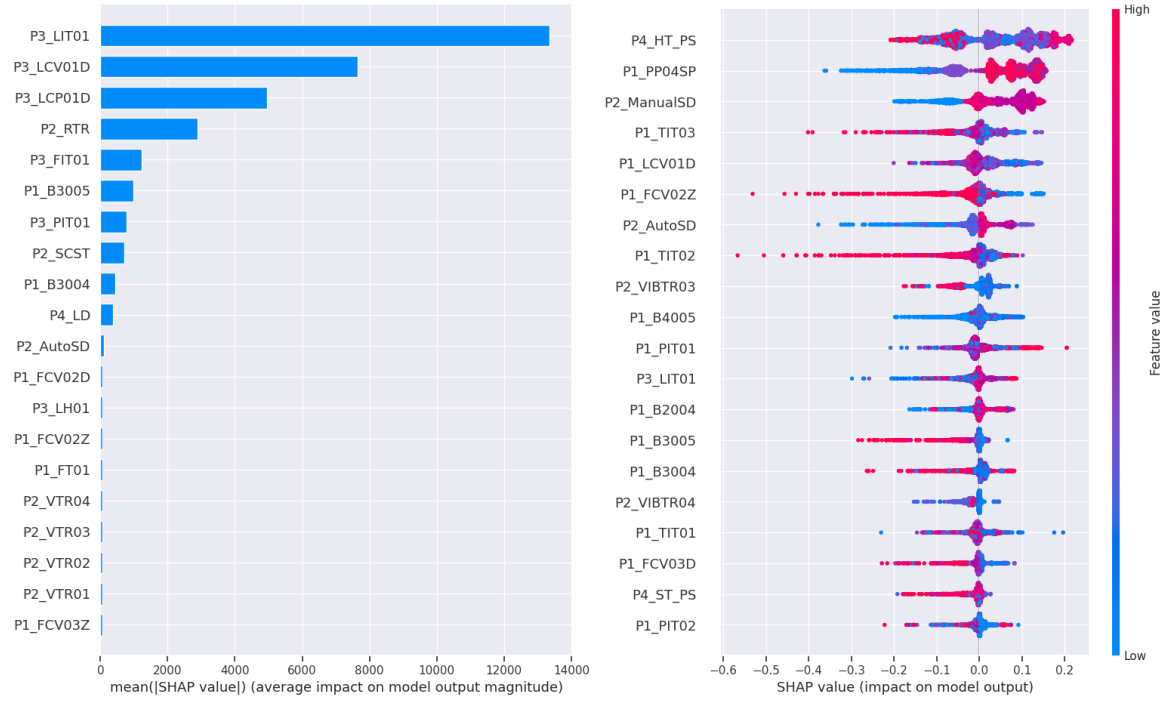
S is a subset of the features used in the model, x is the vector of feature values of the instance to be explained and p is the number of features. $val_x(S)$ is the prediction for feature values in set S that are marginalized over features that are not included in set S. $E_X(\hat{f}(x))$ is the average predicted value.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

$$val_x(S) = \int \hat{f}(x_1, ..., x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(x)) \tag{9}$$

The summary plot is depicted in Figure 9. P3_LIT01 is the most influencing attack type that has the highest weight in determining the magnitude of the target. P3_LCV01D, P3_LCP01D, and P2_RTR are the feature which has the successive order of importance followed by the LIT01 in determining the magnitude of the target. The next representation of the summary plot is a version that determines the importance of features based on the nature of the features towards the prediction of the target shown in Figure 9. P1_PP04SP is the feature that shows that a higher feature value predicts a defect at the target. P2_ManualSD also shows a similar nature. whereas P1_LCV01D and P2_FCV022 are the features that show that the lower value measured on this feature predicts a defect at the target. Thus these two summary plots evaluate the feature importance with the magnitude and nature of the features also in determining the target.



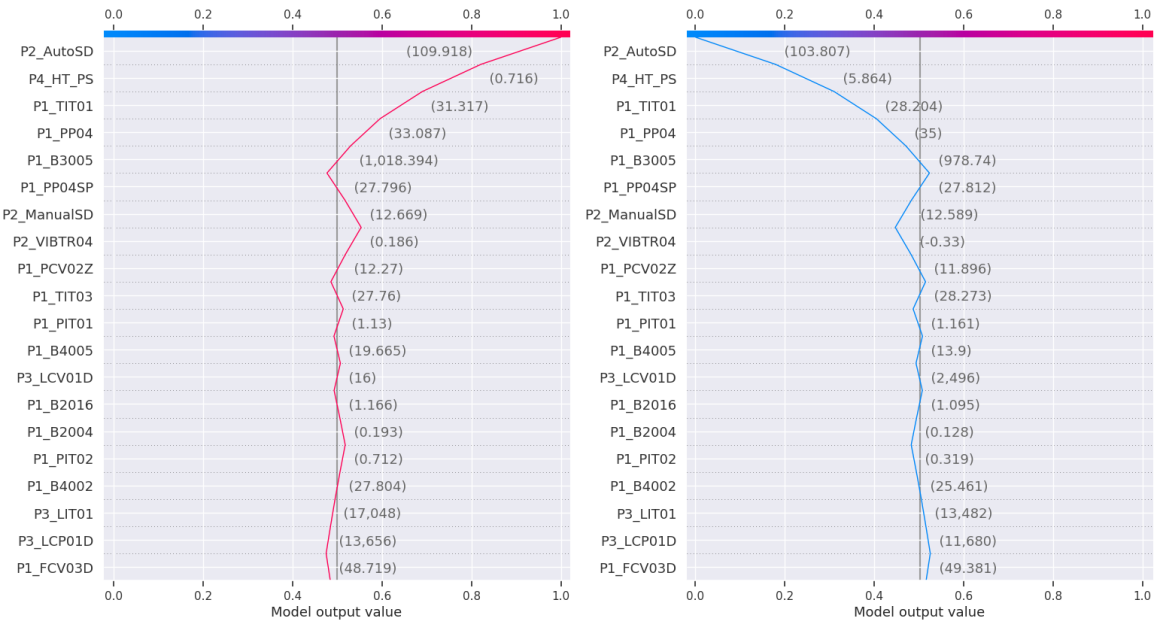
(a) SHAPELY summary plot feature importance (B) SHAPELY summary plot towards target magnitude

Figure 9: Comparison of two SHAPELY summary plots

The last plot of the discussion with XAI implementation is the decision plot. This plot describes the local surrogate values that either correspond to a defect on no-defect prediction (1 or 0) based on the value quantified for various features that correspond to the similar data instance. The decision plot for the prediction of a defect is shown in Figure 10. The decision plot for the prediction of a non-defect is shown in Figure 10. In both cases the features carry specific values that influence the decision plot to keep the prediction as 1 or 0, that is defect or non-defect. This local surrogate model thus describes how the quantification of every feature determines the classification of any instance in the database to predict a defective instance or a normal instance.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024



(a) Decision Plot for the prediction of Defect (b) Decision Plot for the prediction of Non-Defect

Figure 10: Comparison of two SHAPELY summary plots

6 Conclusions

In this research paper, a system called MIDS has been proposed that utilizes machine learning to identify activities in ICS. To enhance the reliability of MIDS and assist security analysts in understanding its decision-making process, XAI techniques have been incorporated. After evaluating the HAI dataset featuring data from a power generation system, it was observed that MIDS achieved an accuracy of 97.95% in detecting faults, using the Random Forest algorithm. Moreover, it was found that incorporating XAI significantly enhanced security analyst's trust and confidence in MIDS. Based on our results, it is believed that MIDS can serve as a tool for enhancing the effectiveness of IDS within ICS. The MIDS can effectively detect anomalous activities in ICS, including stealthy attacks that are difficult to detect using traditional network-based IDS. Furthermore, through XAI integration, insights into MIDS predictions can be gained by security analysts. It is firmly believed that the proposed research on XAI-enabled MIDS holds the potential to make contributions to the field of ICS security. The work aims to improve the effectiveness of IDS in detecting anomalies and attacks in ICS, with increasing the trust and confidence of security analysts in IDS technologies.

7 Author Contributions

All aspects of coding and model training including data collection and analysis, were handled by Dhruv G. Bhatt and Parshad U. Kyada. They also played a role in designing the structure of the paper, creating diagrams, and editing and reviewing the manuscript. The lead in developing the draft and overall structure of this research paper on anomaly detection, in industrial control systems based on measurement data was taken by M.K. Nallakaruppan. He was also responsible for writing and revising the manuscript. The expertise in the Explainable Artificial Intelligence (XAI) component of the research was contributed by Rajkumar Singh Rathore, Rutvij H. Jhaveri, and Faisal Mohammed alotaibi. They also provided guidance and valuable feedback throughout the research work.

DOI: https://doi.org/10.54216/JCIM.150125
Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

Algorithm 1 Anomaly Detection Algorithm

Input:

- $N \leftarrow$ number of defects
- $P_i \leftarrow$ Probability of the defect

Estimate Entropy:

- $e(x) = \sum_{i=1}^{n} -P_i \log P_i$

Estimate Gini Index:

- $Gi = 1 - \sum_{i=1}^{n} P_i$

Contributing Features:

- $F \leftarrow$ Contributing features

Complexity Function:

- $w \leftarrow$ Complexity function

Initialization:

- $t \leftarrow$ origin
- $u \leftarrow$ lasso estimator
- $s \leftarrow$ surrogacy function
- $l \leftarrow$ Loss function
- $r \leftarrow$ Value function
- $v \leftarrow$ value function of the players
- $RFR \leftarrow$ Random Forest Regressor

Decision Explained with Local Surrogates:

if RF is local **then**
$\theta(u) = L(F, t, \pi_u) + \omega(t)$ ▷ **Decision Explained with Local Surrogates**
end if

Decision Explained with Global Surrogates:

if RF is global **then**
$\theta(v) = \sum_{r \subset Rt} |r|!(R - |r| - 1)!/R! * (u(x \cup t - V(r))$ ▷ **Decision Explained with Global Surrogates**
end if

DOI: https://doi.org/10.54216/JCIM.150125
Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

Algorithm 2 Algorithm for Anomaly Classification

Input:

- $X = [\sum_{p=0}^{n} P_n]$
- $Y \leftarrow Y_{train}, Y_{test}$
- $X \leftarrow X_{train}, Y_{test}$
- $P \leftarrow$ samples of images
- $TR_P \leftarrow$ True Positive
- $TR_N \leftarrow$ True Negative
- $FL_P \leftarrow$ False Positive
- $FL_N \leftarrow$ False Negative

Metrics:

- **Accuracy:** $\frac{TR_P+TR_N}{TR_P+TR_N+FL_P+FL_N}$
- **Precision:** $\frac{TR_P}{TR_P+FL_P}$
- **Recall:** $\frac{TR_P}{TR_P+FL_N}$
- **F1 Score:** $\frac{2 \cdot TR_P}{2 \cdot TR_P+FL_P+FL_N}$

Activation: $\max$[Accuracy, Precision, Recall, F1 Score]

while Y $\neq 0$ **do**
  if X_{test} is Anomaly **then**
    X_{test} is Anomaly
    $ACCURACY \leftarrow \frac{TR_P+TR_N}{TR_P+TR_N+FL_P+FL_N}$
    $PRECISION \leftarrow \frac{TR_P}{TR_P+FL_P}$
    $RECALL \leftarrow \frac{TR_P}{TR_P+FL_N}$
    $F1 \leftarrow \frac{2 \cdot TR_P}{2 \cdot TR_P+FL_P+FL_N}$
  else
    X_{test} is Not Anomaly
    $ACCURACY \leftarrow \frac{TR_P+TR_N}{TR_P+TR_N+FL_P+FL_N}$
    $PRECISION \leftarrow \frac{TR_P}{TR_P+FL_P}$
    $RECALL \leftarrow \frac{TR_P}{TR_P+FL_N}$
    $F1 \leftarrow \frac{2 \cdot TR_P}{2 \cdot TR_P+FL_P+FL_N}$
  end =0

DOI: https://doi.org/10.54216/JCIM.150125
Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

References

[1] Bhamare, D., Zolanvari, M., Erbad, A., Jain, R., Khan, K., Meskin, N. (2020). Cybersecurity for industrial control systems: A survey. Computers & Security, 89, 101677.

[2] Stouffer, K., Falco, J., Scarfone, K., & Others. (2011). Guide to industrial control systems (ICS) security. NIST Special Publication, 800(82), 16–16.

[3] Mokhtari, S., Abbaspour, A., Yen, K. K., Sargolzaei, A. (2021). A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics, 10(4), 407.

[4] Bace, R. G., Mell, P., & Others. (2001). Intrusion detection systems.

[5] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1–22.

[6] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24.

[7] Zhang, J., Zulkernine, M., Haque, A. (2008). Random-forests-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(5), 649–659.

[8] Aloqaily, M., Otoum, S., Al Ridhawi, I., Jararweh, Y. (2019). An intrusion detection system for connected vehicles in smart cities. Ad Hoc Networks, 90, 101842.

[9] Javaid, A., Niyaz, Q., Sun, W., Alam, M. (2016). A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), 21–26.

[10] Kumar, S. (2007). Survey of current network intrusion detection techniques. Washington Univ. in St. Louis, 1–18.

[11] Isermann, R., Schaffnit, J., Sinsel, S. (1999). Hardware-in-the-loop simulation for the design and testing of engine-control systems. Control Engineering Practice, 7(5), 643–653.

[12] Bhandary, A., Dobariya, V., Yenduri, G., Jhaveri, R. H., Gochhait, S., Benedetto, F. (2024). Enhancing Household Energy Consumption Predictions Through Explainable AI Frameworks. IEEE Access, 12, 36764–36777.

[13] Murugan, R., Paliwal, M., Lakshmi Patibandla, R. S. M., Shah, P., Balaga, T. R., Gurrammagari, D. R., ... & Jhaveri, R. (2024). Amalgamation of Transfer Learning and Explainable AI for Internet of Medical Things. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 17(4), 40-53.

[14] Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., Halgand, Y. (2015). A survey of approaches combining safety and security for industrial control systems. Reliability Engineering & System Safety, 139, 156–178.

[15] Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. Computers & Security, 56, 1–27.

[16] Wang, C., Fang, L., Dai, Y. (2010). A simulation environment for SCADA security analysis and assessment. 2010 International Conference on Measuring Technology and Mechatronics Automation, 1, 342–347. IEEE.

[17] Kauffmann, J., Ruff, L., Montavon, G., Müller, K.-R. (2020). The clever Hans effect in anomaly detection. arXiv Preprint arXiv:2006. 10609.

[18] Pollastro, A., Testa, G., Bilotta, A., Prevete, R. (2023). Semi-supervised detection of structural damage using variational autoencoder and a one-class support vector machine. IEEE Access.

[19] Roshan, K., Zafar, A. (2021). Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation (SHAP). arXiv Preprint arXiv:2112. 08442.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024

[20] Li, Z., Zhu, Y., Van Leeuwen, M. (2023). A survey on explainable anomaly detection. ACM Transactions on Knowledge Discovery from Data, 18(1), 1–54.

[21] Song, Z., Skuric, A., Ji, K. (2020). A recursive watermark method for hard real-time industrial control system cyber-resilience enhancement. IEEE Transactions on Automation Science and Engineering, 17(2), 1030–1043.

[22] Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. Digital Communications and Networks, 4(3), 161–175.

[23] Arora, P., Kaur, B., Teixeira, M. A. (2021). Evaluation of machine learning algorithms used on attacks detection in industrial control systems. Journal of The Institution of Engineers (India): Series B, 102(3), 605–616.

[24] Kravchik, M., Shabtai, A. (2018). Detecting cyber attacks in industrial control systems using convolutional neural networks. Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, 72–83.

[25] Kadiyala, R., Revathi, A., Gayathri, A., Rutvij, H. J., Lakshmi, N. C., & Naveen, K. B. (2022). WOGRU-IDS—An intelligent intrusion detection system for IoT assisted Wireless Sensor Networks [J]. Computer Communications, 196.

[26] Mao, M., Xiao, H. (2018). Blockchain-based technology for industrial control system cypersecurity. 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018), 903–907. Atlantis Press.

[27] Roshan, K., Zafar, A. (2021). Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation (SHAP). arXiv Preprint arXiv:2112. 08442.

[28] Huong, T. T., Bac, T. P., Ha, K. N., Hoang, N. V., Hoang, N. X., Hung, N. T., Tran, K. P. (2022). Federated learning-based explainable anomaly detection for industrial control systems. IEEE Access, 10, 53854–53872.

[29] Hoang, N. X., Hoang, N. V., Du, N. H., Huong, T. T., Tran, K. P., & Others. (2022). Explainable anomaly detection for industrial control system cybersecurity. IFAC-PapersOnLine, 55(10), 1183–1188.

[30] Spelmen, V. S., Porkodi, R. (2018). A review on handling imbalanced data. 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 1–11. IEEE.

[31] Batista, G. E., Prati, R. C., Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1), 20–29.

[32] Hoque, N., Bhattacharyya, D. K., Kalita, J. K. (2014). MIFS-ND: A mutual information-based feature selection method. Expert Systems with Applications, 41(14), 6371–6385.

[33] Jović, A., Brkić, K., Bogunović, N. (2015). A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1200–1205. Ieee.

[34] Yu, L., Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. Proceedings of the 20th International Conference on Machine Learning (ICML-03), 856–863.

[35] Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., Padma, V. (2020). Study the influence of normalization/transformation process on the accuracy of supervised classification. 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 729–735. IEEE.

DOI: https://doi.org/10.54216/JCIM.150125

Received: April 18, 2024 Revised: June 15, 2024 Accepted: August 20, 2024