Schulich School of Engineering

December 7th, 2022

One-Class Support Vector Machine for Industrial Anomaly Detection

Results and Analysis Utilizing Publicly Available Simulated Chemical Engineering Processes Data

10102804

Monique Beaulieu

30068168

Aamna Amer

TABLE OF CONTENTS

Abstract

As cyber-physical systems (CPS) continue to evolve and become intrinsically interconnected, protocols against threats need to evolve as well, especially when it pertains to critical infrastructure. The recent emergence of machine learning as a means of protecting cyber-physical systems comes with a slew of new problems, including selecting an algorithm that best captures the underlying CPS. This report will analyze a defense system that targets anomaly detection for industrial control systems and will compare it to other existing models for the same publicly available dataset. This report will also attempt to discuss challenges present in anomaly detection which include: appropriate feature extraction, imbalanced distribution, and more. Without proper detection, predictive maintenance and condition monitoring wouldn't be accurate. We aim to solve the anomaly detection problem by proposing a machine learning framework with One-Class Support Vector Machine (OC-SVM) that detects anomalies in time-series data. OC-SVM is an unsupervised algorithm which learns a decision function for novelty detection, well suited for problems within the framework of one-class classification. This unsupervised algorithm includes a sliding window method that restructures the data on the window width of 90 seconds and helps detect anomalies based on the difference between the model output (predicted) and the actual value. In this paper the OC-SVM algorithm is trained and evaluated on the Industrial Control System (ICS) HIL-Based Augmented ICS (HAI) security dataset. The proposed OC-SVM algorithm improves the detection performance F1 score by 6.9% compared to the baseline model with no sliding window.

Introduction

Anomaly detection is the identification of occurrences that give indication of deviation from a dataset's expected behavior and can also be data points that differ in characteristic from most of the processed data. As cyber-physical systems are a complex ensemble of embedded systems, it can be challenging to model a robust cybersecurity protocol to help detect malware and other potential cybersecurity threats, such as intrusion detection, fraud detection, and defect detection.[1] As such, anomaly detection is often utilized in the industry due to its capacity to help detect novel attacks, insider attacks, as well as zero-day vulnerabilities that are unknown even to the supplying vendor.

Anomalies are often classified into three categories: global outliers, contextual outliers, and collective outliers, as shown in Figure 1: Types of Anomalies. Global outliers or point outliers are values that strongly deviate from the standard, whereas contextual outliers are values that are within a system's operating range, however are abnormal compared to the seasonal pattern. Collective outliers describe a collection of data points that are irregular in respect to the entire dataset. The data points may not differ in a global or contextual sense, but rather appear as a collection that deviates from the expected pattern. In general, large deviation in data points immediately signals a sign for further investigation, and as such, this principle is utilized in anomaly detection to promptly flag atypical behavior by combining typical behavioral and contextual attributes.

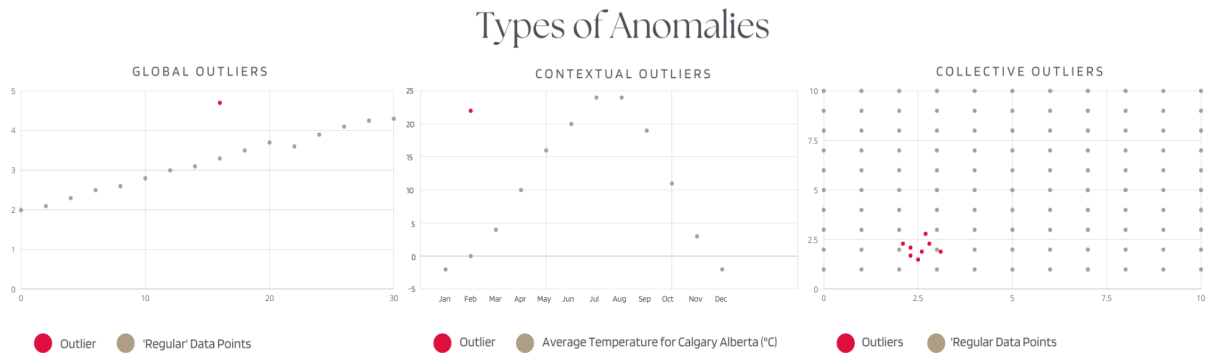

Figure 1: Types of Anomalies. *Depiction of three main types of anomaly classification, i.e. global outliers, contextual outliers, and collective outliers.*

Similarly to the type of anomalies, there are also three main anomaly detection classifiers: supervised, unsupervised and semi-supervised. Supervised learning requires data to be labeled with the expected model output and as such demands a dataset that is within the operating domain of the existing cyber physical system in order to function correctly. This approach lacks class imbalance in most cyber-physical systems, as data pertaining to attacks is often not made public.

Unsupervised learning is able to categorize data into clusters of similarity which makes it especially efficient at detecting outliers. Unsupervised machine learning is useful in anomaly

detection of ICS since these problems don't always have accurately labeled data, and often the model needs to identify an outlier without necessarily seeing an example of anomalous behavior before.[7] Therefore, it is necessary to construct a machine learning framework that accurately identifies an attack by learning how to distinguish abnormal data from normal states.

In this report, a publicly available HIL-Based Augmented ICS (HAI) security dataset was trained using unsupervised learning with One-Class Support Vector Machine (OC-SVM) and a sliding window method, for the purposes of identifying contextual outliers. OC-SVM is a special case of Support Vector Machine that measures the position of new data relative to the normal or inlier training data to determine if it is unusual.[7] The proposed method includes preprocessing data, adding a sliding window, fitting unsupervised OC-SVM and evaluating the model's test accuracy against previous iterations. The methods of assessing the model's effectiveness include accuracy, precision, recall, F1, and Area Under the Curve (AUC). The proposed algorithm improves the F1 score by 8.54% compared to the baseline model without a sliding window method. The results obtained from the proposed model are compared to literature papers that have utilized the same training and testing dataset. The results from the comparison are then used to make informed and logical decisions regarding future improvements.

Background

HAI Dataset 22.04[3]

The publicly available dataset was collected from an industrial control system (ICS) testbed augmentation with a Hardware-In-the-Loop (HIL) simulator that emulates realistic data regarding steam-turbine power generation and pumped-storage hydropower generation. The ICS testbed contains four processes:

1. Boiler Process: Includes a water-to-water heat exchanger operated at low pressure and moderate temperature.
2. Turbine Process: Consists of a rotor kit process that emulates an actual rotating machine.
3. Water Treatment Process: Involves pumping water to the upper reservoir and releasing it back into the lower reservoir.
4. Hardware-In-The-Loop: The boiler and turbine processes are synchronized to match the rotating speed of the virtual steam-turbine generation value. The pump and value in the water treatment process is controlled by the pumped-storage hydropower generation mode. Thus, creating an interconnected system between the other three processes.

One measurement is obtained every second with 86 different sensors, actuators, control devices etc. that represent the current status of the system. This dataset is pre-divided into two types of data, training and test. The training dataset was collected during normal operations and consists of 6 csv's with time continuity since they were collected in 6 different terms, summing to 269 hours. Similarly, the test dataset consists of 4 csv's with temporal continuity summing to 90 hours however, containing a total of 58 simulated cyber-attacks. A summary of the HAI 22.04 dataset can be seen in Table 1: Summary of Public HAI 22.04 Dataset.

Type	Number of Time Intervals	Seconds	Attacks
Training	6	968,400	0
Test	4	324,000	58

Table 1: Summary of Public HAI 22.04 Dataset

This simulated data closely resembles critical infrastructure utilized in chemical engineering processing plants, and thus the knowledge can be transferred over and applied in real-world industry. To further apply this, a model of the simulated ICS can be seen in Figure 2: Industrial Control System Model of the HAI Dataset to help us specify and understand the behavior of the HAI system. The ICS model relies on user input or tokens which can be showcased as black circles, controllers or transitions which are represented by the black rectangles,diamonds which showcase the action taken, and boxes which represent physical components of the HIL's simulation. The ICS model displays the relationships between the physical components and the digital inputs of the cyber-physical system.

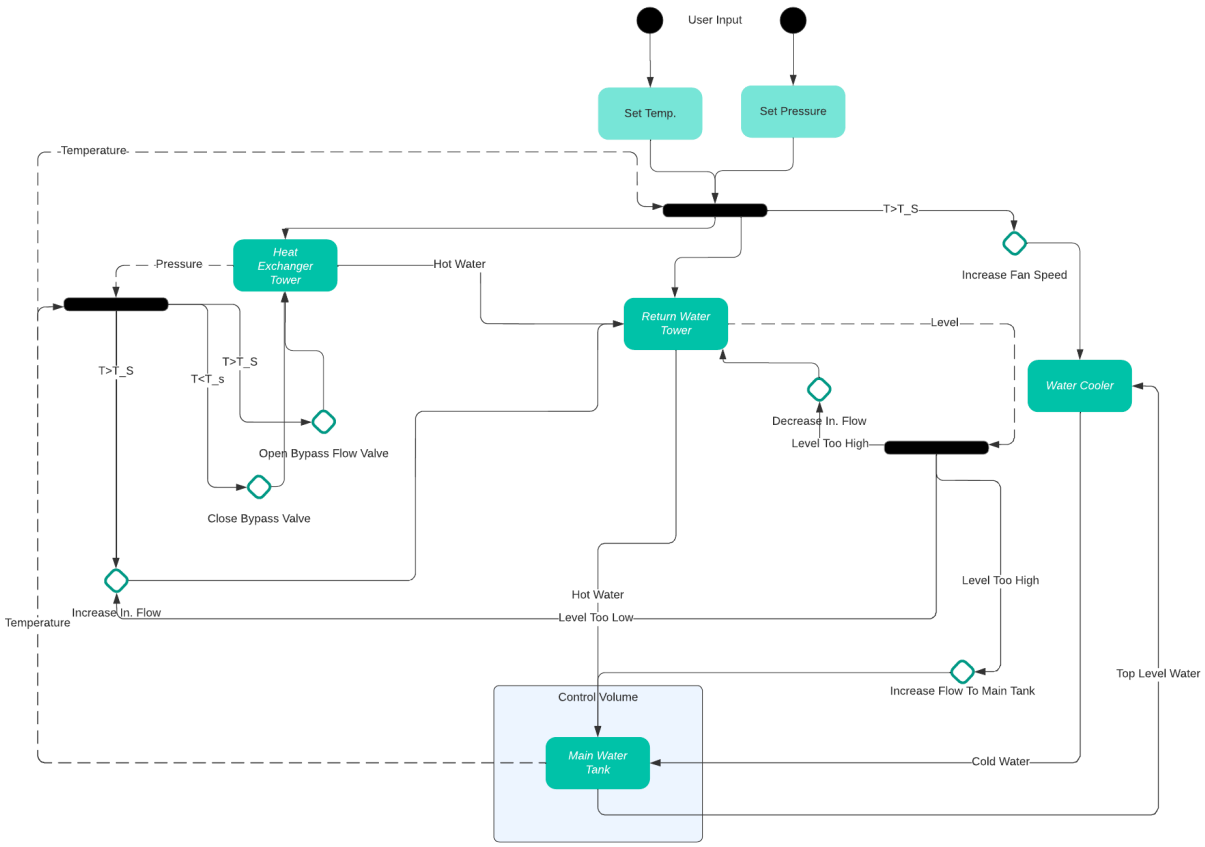

Figure 2: Industrial Control System Model of the HAI Dataset.

Methodology

System Structure

The overall structure of the OC-SVM framework for the HAI dataset can be seen in Figure 3. Overall framework of OC-SVM for Anomaly Detection. The framework includes data collection, a data engineering unit, an unsupervised model (OC-SVM) that uses a sliding time window, and the anomaly detection mechanism that classifies the output data.

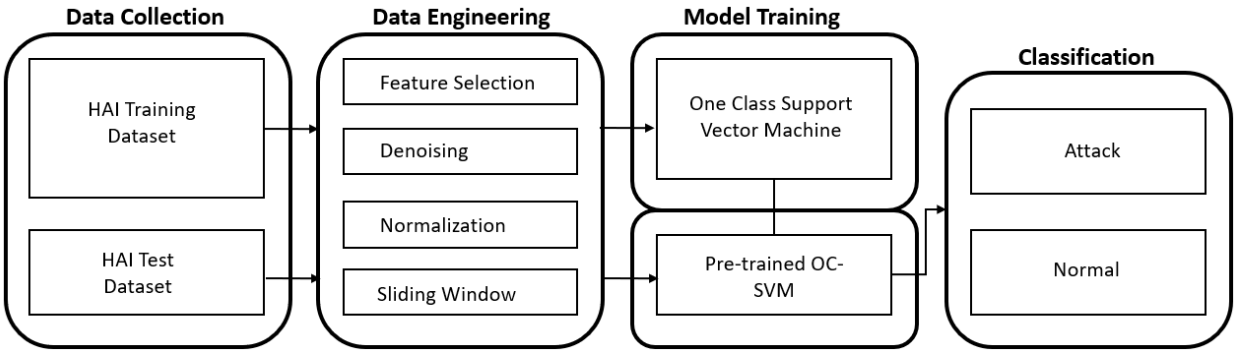

Figure 3. Overall framework of OC-SVM for Anomaly Detection

Data Analyzing and Preprocessing

The training and test data come from raw sensor measurements with 86 different features. When these features are visualized (as seen in Appendix A figures A.1 and A.2) the measurements range from bimodal, unimodal, and multimodal distributions. This can indicate possible categorical data, no variance data, and possible noisy data. Therefore, the data is normalized using min-max then the noise generated by the sensors is smoothed by using the exponential weighted function.

Feature Engineering

Due to the large quantity of sensors in this ICS, the measured data taken every second becomes too substantial for the model algorithm to predict within a reasonable time. The goal of finding principal features is to decrease the data down to the most effective features that lead to more accurate models and less computation time.[6] The first features dropped were categorical data identified by visualizing the features (Appendix figures A.1 and A.2). The next to be dropped were all features with a value of 0 when normalized and all features with no variance, these features did not contain any useful information for the model. In addition, a correlation criteria was employed to filter features with high multicollinearity. Multicollinearity is a statistical concept where independent variables in a model are correlated, this results in less reliable statistical inferences.[4] The multicollinearity of features were assessed through a correlation matrix. Only the features with low collinearity (less than 0.5) were chosen to proceed to the model. After applying these feature engineering techniques, the quantity of principal features became 14 rather than the original 86.

To further highlight the choice of the 14 principal features selected, Kernel Distribution Estimation (KDE) Plots of these features, when normal, were overlain with these same features

when they were under attack (Appendix A figure A.3). These plots visualize the probability density function of continuous data and show the contrasting behaviors of these features when under attack. It is obvious here that feature selection plays a very significant role in the performance of a model.

OC-SVM Model

One class classification (OCC) algorithms are effective for imbalanced classification datasets where there are few or no examples of the minority class (attacks).[2] OCC involves fitting a model on the normal class and once trained, can be used to classify new data as outliers or anomalies. In this framework, a OC-SVM model was used as the OCC algorithm since it captures the density of the majority class in an unsupervised manner and classifies examples on the extremes of the density function as outliers.

The input to the OC-SVM model is the measurement in the sliding window that restructures the 14 principal features selected and their time-series data. In the final framework the window was set to 89 seconds, the output of the model estimates the 90th second and the difference between the estimated and actual measurement. If the difference between the actual and predicted is large, the observation is likely an attack. Due to the unsupervised nature of this model, it was well-trained on normal situations where the difference between the actual and predicted value was normal. Therefore, a large difference would indicate an unfamiliar measurement not included in the training set likely labeling it as an outlier.

When training the OC-SVM, critical hyper parameters included:

- Nu: specifies the percentage of anomalies expected in our data, in this case it was set to 0.03445.
- Kernel: the non-linear function to project the hyperspace to a higher dimensionality to help the SVM model draw a decision boundary.[1] This was set to radial basis function (rbf) as the number of observations was much larger than the number of features.
- Gamma: the kernel coefficient that controls the influence of individual training samples, in our algorithm it was set to be 5e-05 to improve the smoothness and generalizability of the model.[7]

To find the best performing algorithm of OC-SVM, 6 different iterations were performed with varying feature selection and time window frames. The second iteration was selected as the baseline model with 15 key features with the most variance and no sliding time window implementation.

Result Analysis

Performance Metrics

The performance of machine learning algorithms can be measured via various metrics, including common ones such as root mean square error, average deviation, and many more depending on the AI domain selected. For the purpose of this report, the performance of the machine learning algorithm is measured in terms of accuracy, precision, recall, F1-score, and

Area Under the Curve (AUC). Accuracy will help measure the degree to which the results conform to the standard or correct value, whereas precision will help measure the refinement or reliability of the scores produced and can be calculated with the following equation:

$$Precision = TP/(TP + FP)$$

The values obtained for TP, or true positive, and FP, or false positive, are summarized in the figure below.

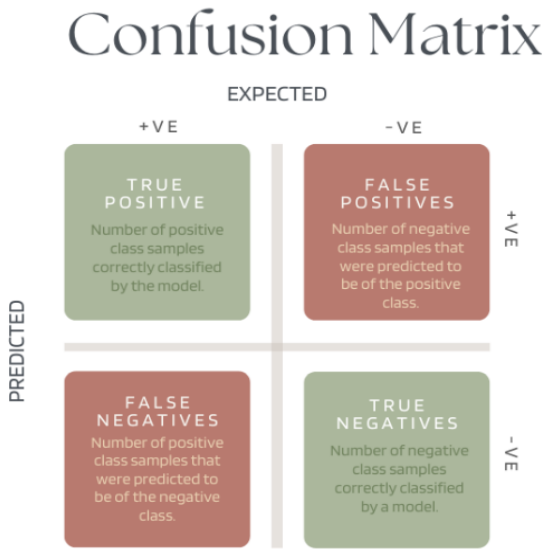

Figure 4: Confusion Matrix of a Binary-Class Dataset.

Recall, on the other hand, refers to the ratio of true positives to the combined total of true positives and false negatives outputted by the system, and is often a value between 0 (no recall) to 1.0 (full recall).

$$Recall = TP/(TP + FN)$$

The resulting value measures an algorithm's ability to detect positive samples. The F-measure or F1-score is a statistical measure of accuracy of a machine learning model and takes into account both precision and recall via a harmonic mean mathematical formula:

$$F - measure = \frac{2*Recall*Precision}{Recall + Precision}$$

The preferred value of the F-measure is a score of 1 which indicates perfect accuracy and recall of the model. As for the area under the curve metric, AUC is a popular classification metric used to evaluate a machine learning algorithm’s ability to distinguish between classes. The higher the AUC, the better the performance of the model at distinguishing positive and negative classes, or in the case of this report, normal and attack classes. With the exception of accuracy, all the metrics proposed are robust and often used to determine the performance of classification algorithms in their attempt to classify occurrences or data points in the cyber-physical layer into the normal or attack class.

Results

The machine learning model underwent six iterations, a summary of the changing variables can be found in Table 2: Summary of Iterations Conducted and Variables Altered. A baseline was established for the sake of making meaningful comparisons and for providing insights regarding the variables altered. Since the first iteration had significant errors and produced insignificant results, the second iteration was established as the baseline.

Iteration	Feature Type	Time Window	Kernel	Gamma, γ	Nu, ν
1	15 Random Features	N/A	RBF	0.00005	N/A
2	15 Key Features with Most Variance	N/A	RBF	0.00005	0.03445
3	9 Key Features with Most Variance and No Collinearity	40	RBF	0.00005	0.03445
4	14 Key Features with Most Variance and No Collinearity	40	RBF	0.00005	0.03445
5	19 Key Features with Most Variance and No Collinearity	40	RBF	0.00005	0.03445
6	14 Key Features with Most Variance and No Collinearity	90	RBF	0.00005	0.03445

Table 2: Summary of Iterations Conducted and Variables Altered. *The final iteration as well as the baseline model are highlighted in the table to showcase their significance.*

The final results of the proposed machine learning model , i.e. iteration six, which incorporated vector quantization for anomaly detection, as well as the baseline model can be seen in Table 3: One Class Support Vector Machine Results for Anomaly Detection in Industrial Processes. The last column in the table showcases an improvement score obtained by calculating the percent difference in metric values from iteration one and six and then dividing over the values obtained from run six.

A significant performance increase can be noticed as between the 2nd and 6th iteration when collinearity is eliminated. This is expected to happen as collinear features can cause the model to bias towards the wrong independent variable that is governing the system. Thus, the elimination of collinearity results in the model's ability to identify the relationship between independent variables and the output of the system. Additionally, the correlation between performance and an increased time window selection can be explained by the ability of the model to identify patterns that persist over a longer period of time as opposed to noise that is often short lived and would have a higher contribution to the results in shorter time windows.

A comparison of all the iterations is depicted in Figure 5: Graph of Performance Versus Iteration Against Multiple Metrics.

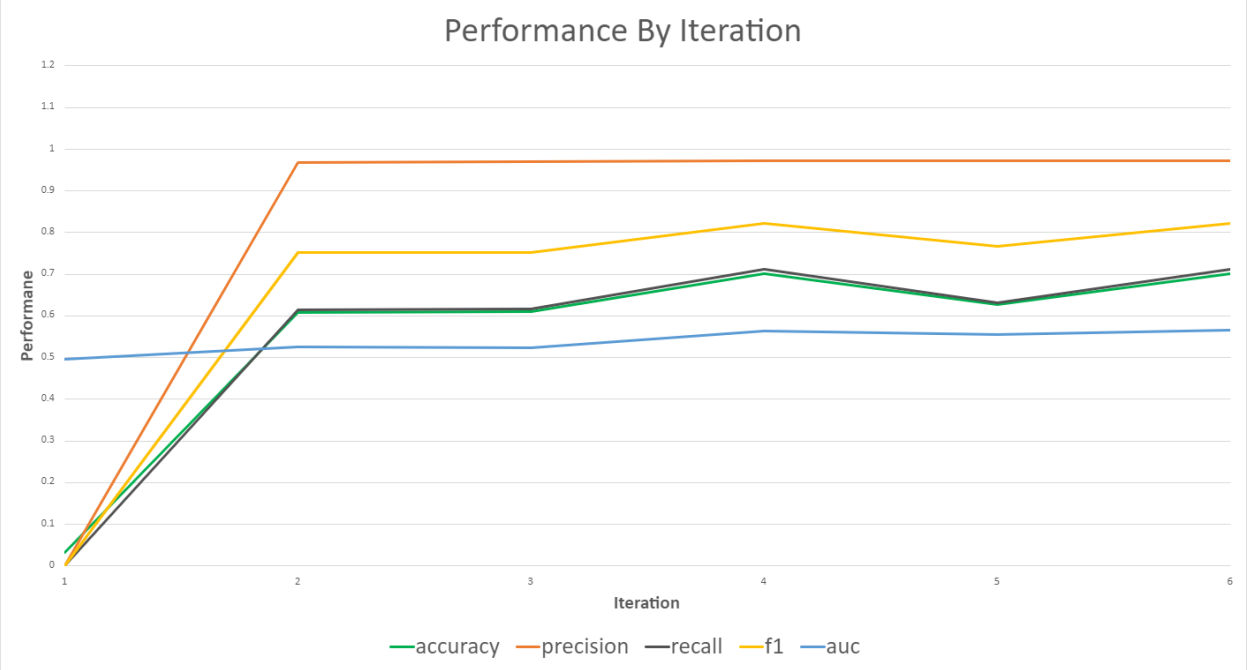

Figure 5: Graph of Performance Versus Iteration Against Multiple Metrics

From the above graph we notice that an increase in key features results in an increase in most performance metrics. Importantly, iteration 5 and 6 help isolate the impact of the time sliding window parameter over the number of key features. We notice that an increased time window achieves the highest performance out of all the iterations. Similarly, between iterations 4 and 5, we notice a drop in recall and accuracy as the number of key features is increased to 19. Thus signifying that the optimal value may be 14 key features.

Metric	**Values**		
	6th Iteration	**2nd Iteration (Baseline)**	**Performance Difference (%)**
Accuracy	0.70	0.61	12.86%
Precision	0.97	0.97	0.00%
Recall	0.71	0.62	12.77%
F-Measure	0.82	0.75	8.54%
Area Under the Curve (AUC)	0.56	0.52	7.14%

Table 3: One Class Support Vector Machine Results for Anomaly Detection in Industrial Processes

From the values obtained for the sixth iteration, it is evident that the accuracy of the model is not within the acceptable margin of error of +/- 10%, thereby signifying that more work must be done to finetune the model, which could include, but is not limited to, adjusting the hyperparameters and integration of an additional machine learning algorithm. The precision of the algorithm is considerably better, indicating that the model is able to correctly identify and classify data into the normal class. Much like accuracy, the recall value exceeds the acceptable range of error, however, similarly to precision, the higher the value, the better the results. In the case of the recall value, a low recall demonstrates a model's inability to correctly identify the normal class as such. The combination of a high precision score and low recall value indicates that the proposed solution returns few results, but that most of its predicted values are correct. The proportionality of recall and precision is also captured in the F-measure, which although still not acceptable, is still much better than the accuracy and recall values.

As for the AUC metric, it is evident that the area under the curve metric performed the worst overall. Although there is no threshold for AUC values, any value near 0.5 is considered poor and is equivalent to a model that randomly guesses. In general, a value of 0.5 showcases no determination, 0.5-0.7 indicates poor discrimination, 0.7-0.8 is considered acceptable discrimination, 0.8-0.9 implies excellent discrimination, and finally, anything beyond 0.9 signifies outstanding discrimination. The results from AUC metric demonstrates that there is little to no discrimination between the attack and normal class, which further implies that the model explored within this paper cannot accurately and reliably determine anomalies.

Discussion

Comparison

The latest dataset (version: HAI 22.04) contains 86 data points per second with six training files and 4 test files containing a total of 58 sophisticated cyber-attacks.[3] The dataset utilized in this project report had a 98% to 2% normal to attack ratio, with the attack types being categorized into three main categories: injection, replay and modifications. The sample dataset was 23.5 MB in size with approximately 43,201 data points. As this publicly available dataset is robust and continuously updated year over year since its inception in 2017, it is a popular choice for testing machine learning algorithms in the academic realm. There have been several research papers that have utilized this dataset, with the most recent addition being "Benchmarking Machine Learning based Detection of Cyber Attacks for Critical Infrastructure," published by IEEE and authored by Ajit Kumar and Bong Jun Choi.[5] The performance of the algorithm proposed in this paper was compared to existing models addressed in the IEEE paper, the results of which can be found in Table 4: Comparison of Proposed Model Against Models from Literature Sources. Algorithms that performed the best in any given category were highlighted in green. Similarly, any algorithms with the lowest score in the five defined metrics were highlighted in green.

Metric	Accuracy	Precision	Recall	F-Measure	Area Under the Curve (AUC)

Proposed Model (OC-SVM)	0.70	0.97	0.71	0.82	0.56
Gaussian Naive Bayes (GNB)	0.72	0.52	0.86	0.46	0.98
Linear Support Vector Classifier (LSVC)	0.98	0.49	0.50	0.50	0.52
K-NN (K-Nearest Neighbor)	0.99	0.96	0.95	0.96	0.98
DT (Decision Tree)	0.99	0.99	0.99	0.99	0.94
RF (Random Forest)	0.99	1.00	0.99	0.99	0.99
ABoost (AdaBoost)	0.99	1.00	0.98	0.99	0.99
BSVC	0.99	0.49	0.50	0.50	0.50
LR (Logistic Regression)	0.98	0.49	0.50	0.50	0.84
Gboost (Gradient Boost)	0.99	0.99	0.97	0.98	0.99

Table 4: Comparison of Proposed Model Against Models from Literature Sources.[5] *Best scores for the five metrics are highlighted in green and the worst scores are highlighted in red.*

Based on Table 3: Comparison of Proposed Model Against Models from Literature Sources, it is clear that the random forest classifier outperformed all the other algorithms, with the decision tree classifier following close behind. The worst performing algorithms included the likes of BSVC, logistic regression, and the linear support vector classifier. Although the model proposed in this research paper did not perform the worst overall, it was found to be lacking in the accuracy metric given by the fact that it only managed to acquire a 70% for accuracy while the rest of the models averaged around the 99th percentile. As for the recall, F-measure, and AUC metrics, the proposed model performed on the lower end of average and performed better than average for the precision metric.

Future Recommendations

Due to a time constraint, a simplified machine learning algorithm was utilized for training and testing**.** As the anomalies present in the dataset are intentionally designed to be difficult to detect, consequently making it a better standard to aim for in industries with critical infrastructure, a generic, out of the box solution is not enough.

- In order to create a robust and adaptive machine learning algorithm, an ensemble of machine learning algorithms such One-Class Support Vector Machine with the addition of Vector Quantization could have been explored to detect the anomalies present, while

being mindful of overfitting the model. The addition of the Vector Quantization (VQ) would have assisted in managing the large stream of data being inputted into the industrial control system. Vector Quantization is a technique that relies on lossy data compression to compress information to 40-60% of its original size while retaining all necessary components.[5]

- Other reduction methods could have also been explored, such as principal component analysis (PCA), which much like vector quantization, helps reduce data quantity without a significant or apparent decrease in accuracy. Principal component analysis is a popular reduction method for datasets with a high number of dimensions or features per observation. Since HAI 22.04 contained approximately 86 observable features, it would have been worthwhile exploring the effect of PCA and VQ on the dataset's accuracy.
- It is also important to note that based on literature results showcased in Table 3: Comparison of Proposed Model Against Models from Literature Sources, it would be imperative to explore creating an ensemble with the random forest classifier as the base model instead of OC-SVM since it outperformed nine different machine learning models.

Conclusion

In this paper, we propose an anomaly detection framework called One Class Support Vector Machine (OC-SVM) that detects outliers in time-series data from the publicly available dataset. HAI 22.04 is composed of nearly 43,000 data points and has a 98:2 normal to attack ratio, with the attack types being categorized into three main categories: injection, replay and modifications. The proposed model attempts to ingest the large volume of data and correctly identify the datapoint into the attack or normal class. This is done by utilizing an unsupervised framework with a sliding time window to help detect anomalies based on the patterns that the model has never seen in the training dataset before. The explored OC-SVM algorithm, which takes into account 14 key features with the most variance and no collinearity, demonstrates improvements by upwards of 8.54% (F1 score) in comparison to the baseline models which detects 15 key features with the most variance only. Although this paper showcases improvements in comparison to the baseline model, the test metrics do not meet industry standards in detecting cyber attacks in real-time and signify that further adjustments need to be made in order to create a long withstanding and dynamic detection algorithm.

Appendix A

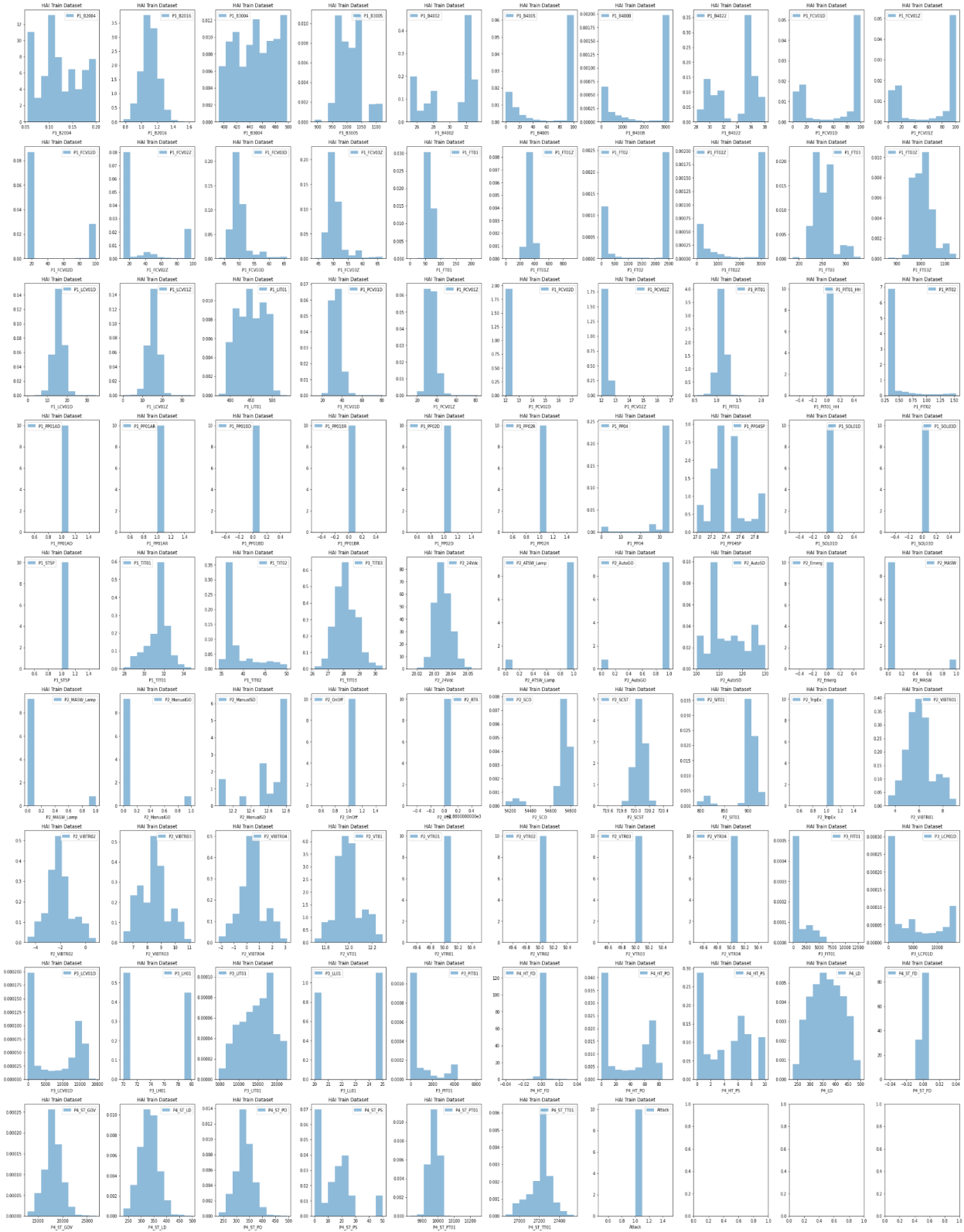

Figure A.1: Training Data Feature Visualization

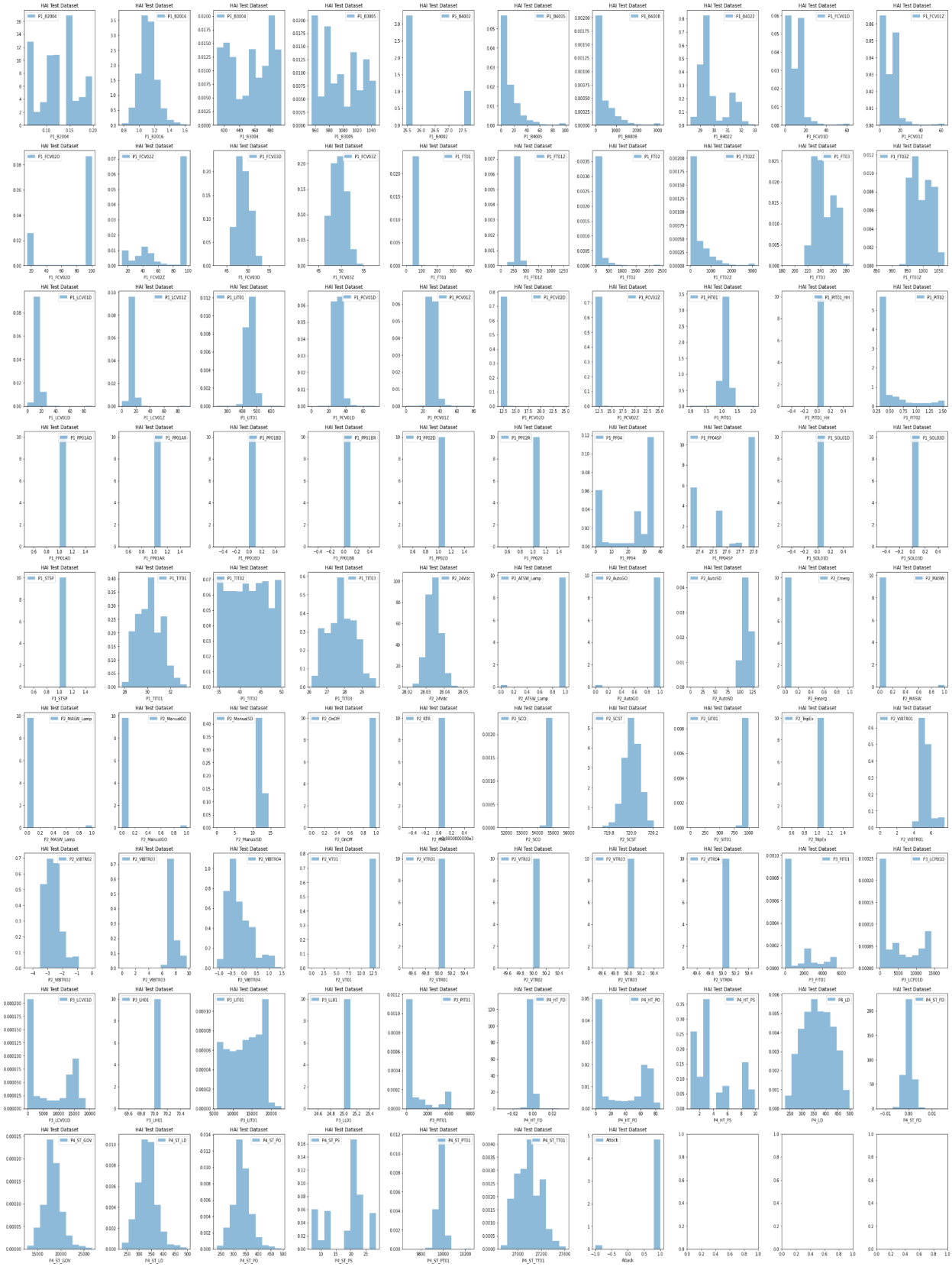

Figure A.2: Testing Data Feature Visualization

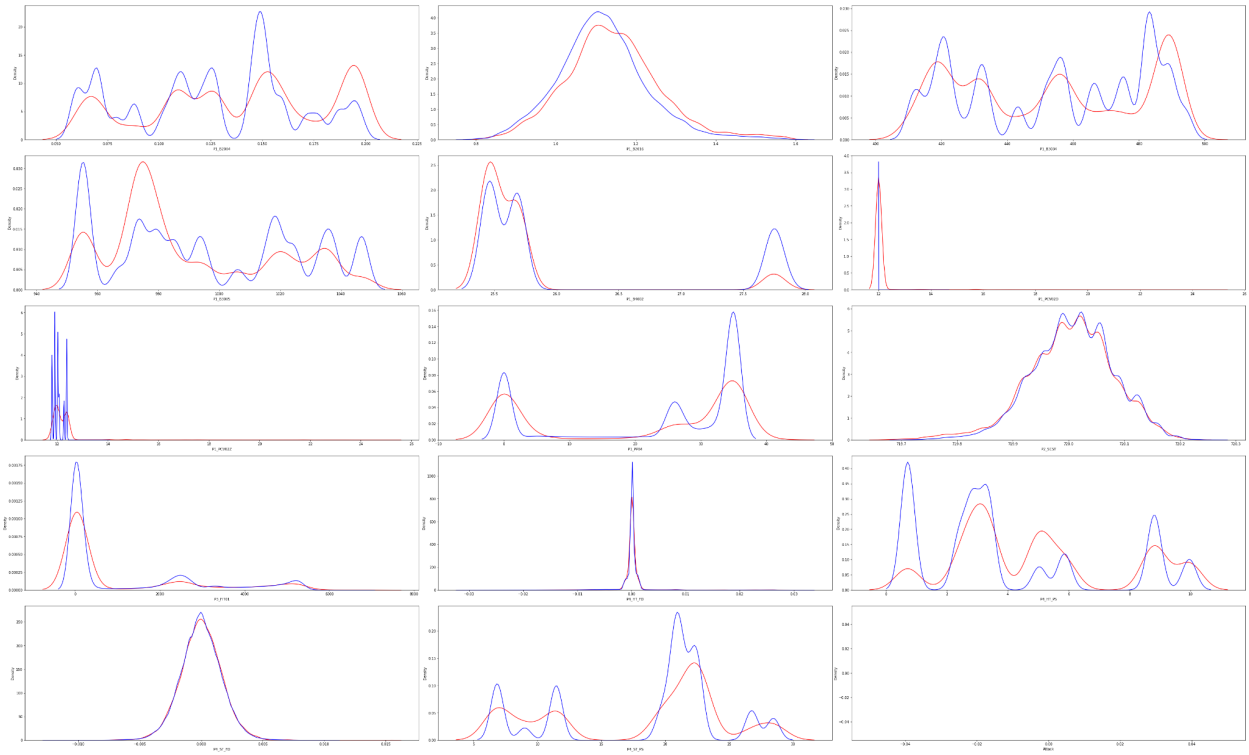

Figure A.3: Normal vs Attack Principal Feature KDE Plots

References

1. GrabNGoInfo, A. (2022, November 15). One-class SVM for anomaly detection. Medium. Retrieved December 5, 2022, from https://medium.com/grabngoinfo/one-class-svm-for-anomaly-detection-6c97fdd6d8af
2. Brownlee, J. (2020, August 20). One-class classification algorithms for imbalanced datasets. MachineLearningMastery.com. Retrieved December 5, 2022, from https://machinelearningmastery.com/one-class-classification-algorithms/
3. Hai Security dataset. Kaggle. (n.d.). Retrieved December 5, 2022, from https://www.kaggle.com/datasets/icsdataset/hai-security-dataset
4. Hayes, A. (2022, September 27). Multicollinearity. Investopedia. Retrieved December 5, 2022, from https://www.investopedia.com/terms/m/multicollinearity.asp#:~:text=Multicollinearity%20is%20a%20statistical%20concept,in%20less%20reliable%20statistical%20inferences.
5. Kumar, A., & amp; Choi, B. J. (n.d.). Benchmarking machine learning based detection of cyber attacks for ... Benchmarking Machine Learning based Detection of Cyber Attacks for Critical Infrastructure. Retrieved December 5, 2022, from https://ieeexplore.ieee.org/document/9687293/
6. Mokhtari, S., Abbaspour, A., Yen, K. K., & Sargolzaei, A. (2021, February 8). A machine learning approach for anomaly detection in industrial control systems based on measurement data. MDPI. Retrieved December 5, 2022, from https://www.mdpi.com/2079-9292/10/4/407/htm
7. Stradling, J. (2016, October 25). Unsupervised machine learning with one-class support vector machines. Medium. Retrieved December 5, 2022, from https://medium.com/@jamesstradling/unsupervised-machine-learning-with-one-class-support-vector-machines-129579a49d1d