ANOMALY DETECTION USING MACHINE LEARNING IN CYBER PHYSICAL SYSTEMS

A research project report submitted
in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Chemical Engineering

by

Divyang Deep Tiwari [16CH10029]
Sourjya Naskar [16CH10006]
Arasavelli Venkata Siva Sai [16CH10010]

Under the guidance of

Dr. Venkata Reddy Palleti
Dr. Veerabhadra Rao Chandakanna

The Department of Chemical Engineering

Indian Institute of Petroleum and Energy

Visakhapatnam

Indian Institute of Petroleum and Energy, Visakhapatnam

CERTIFICATE

It is certified that the research project entitled "**ANOMALY DETECTION USING MACHINE LEARNING IN CYBER PHYSICAL SYSTEMS**" submitted to Indian Institute of Petroleum and Energy, Visakhapatnam, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Chemical Engineering by **Divyang Deep Tiwari, Sourjya Naskar, Arasavelli Venkata Siva Sai** is a record of bonafide work carried out by them under the supervision and guidance of **Dr. Venkata Reddy Palleti and Dr. Veerabhadra Rao Chandakanna** during the academic session 2019 - 2020.

Dr. Venkata Reddy Palleti
Assistant Professor
Chemical Engineering Department
Indian Institute of Petroleum and Energy
Visakhapatnam

Dr. Veerabhadra Rao Chandakanna
Assistant Professor
Computer Science and Engineering
Indian Institute of Petroleum and Energy
Visakhapatnam

Abstract

Cyber Physical Systems (CPS) are the computer systems where a process is controlled or monitored by computer-based algorithms. In CPS, physical and software components are deeply intertwined and interact with each other in ways that change with context. In this report, we tried to train different Machine Learning models for anomaly detection and compared their performance based on the confusion matrix of predicted outcomes. These anomaly detection methods are: One-class SVM, Isolation Forest and Elliptic Envelope. The data from normal operation was extracted, cleaned and transformed. Thereafter, models were trained, tested on new data and accordingly their parameters were tuned for optimum performance. The objective of this analysis is to improve the security of continuous processes against deviation as well as intrusion detection capability.

Keywords— *Cyber Physical Systems (CPS), Anomaly Detection, Machine Learning, One-Class SVM, Isolation Forest, Elliptic Envelope, Principal Component Analysis (PCA), Dimensionality Reduction, Confusion Matrix, Imbalanced Datasets, Supervised and Unsupervised Learning.*

TABLE OF CONTENTS

LIST OF TABLES

Tables

LIST OF FIGURES

Figures

I. Introduction

The automation in industries is increasing proportionally with the advancement of technology. Although, security infrastructure of Industrial Control Systems (ICS) is continuously undergoing development, there is a possibility that Machine Learning and Data Science applications could merge to help us identify deviation in processes more effectively. This work is about exploring this possibility. Moreover, Machine Learning allows us to take into account all inputs at once and return the status of the process. Therefore, it is possible that these trained models can be installed in process industries and could potentially act as another layer of security and monitoring.

The dataset used in this research is of the Water Distribution (WADI) Plant provided by itrust Centre at Singapore Institute of Technology and Design (SUTD). Entire process is divided into 3 sub-processes (Process 1, 2 & 3) and the complete data has been divided into 2 parts:

- Initial 12 days operation data (standard operation)
- Final 2 days operation data (all the attacks were deployed in this duration)

A total of 15 different types of attack has been planned and conducted on the plant which cumulatively approximate to 10,000 data points. The aim is to maximize the attack detection with minimum false alarms raised i.e increasing both precision and recall simultaneously.

II. Literature Review

This section discusses the relevant work carried out by other researchers in the field of intrusion and anomaly detection. A number of papers discuss supervised learning algorithms, such as Decision Trees, Naive Bayes, Neural Network, Support Vector Machines (SVM), Multi Layer Perceptron (MLP), Random Forest, Random Tree, k-Nearest neighbour classifiers.

Observations made by Gharibian and Ghorbani [1] demonstrated that Decision Trees are very sensitive to the training data and don't learn quite well from imbalanced data. Furthermore, they found that Decision Trees and Random Forests (ensemble of Decision Trees) are very sensitive to the training data and their performance can vary significantly based on it. Naive Bayes Classifier implements Bayes theorem for classification problems. In comparison to MLP, Naive Bayes classifiers can be trained within a short period of time [2]. According to Domingos and Pazzani [3] & Langley and Sage [4] Naive Bayes can perform quite well when there exists a reasonable dependency in the data. It has been also observed that the performance of Naive Bayes classifier improves when redundant features are removed.

The observations from Huy Anh and Deokjai [5], where they applied a wide variety of algorithms: Bayesian approach, Decision Trees, Rule based models, etc states that no single algorithm is capable of

detecting all kinds of attack with high detection and low false alarm rate. Muda Z, et al. [6] performed a hybrid learning approach by combining Naive Bayes and K-means Clustering. The training dataset was divided into k-clusters based on an initial value known as the seed points into each cluster center. The results showed that this hybrid approach performed better as compared to only Naive Bayes Classifier. Wang H. et al. [7] made an attempt to improvise SVM by combining Principal Component Analysis (PCA) and Particle Swarm Optimization (PSO). PCA is quite effective in reducing dimensions of data. Thereafter, PSO was used to optimize the kernel parameters. The experimental results showed that the performance of PCA and PSO combined SVM was higher than those of PSO-SVM and standard SVM. Another work is conducted by Cheng Feng, Venkata Reddy et. al [8], proposed an approach which combined several machine learning and data mining techniques to generate a significant number of invariant rules (defined as a physical condition that must be satisfied for any given state of an ICS [9] [10]). They observed that generated invariant rules can achieve high anomaly detection performance by demonstrating on two real world ICS case studies. Their results outperform commonly used residual error-based anomaly detection models. Another advantage of this methodology is that it can be applied to diverse ICS frameworks as it is dependent only on general control dynamics of ICS.

The approach followed in this work is slight different and includes the outlier detection methods for unsupervised learning as the WADI dataset doesn't have a labelled target variable. The merits of these methods are as follows: 1) Can be trained in a reasonable amount of time 2) Random samples can be used instead of complete dataset as the methods compute a hypothetical surface boundary for the training data 3) Sensitive to training set (i.e customizable for the definition of inliers).

III. Objective

The objective of this work is to explore the possibilities in which Machine Learning models could be deployed on the field to detect abnormalities in the process. These abnormalities are the deviations from standard operation that may arise when an attacker attempts to harm the process.

Outlier and Novelty Detection classification algorithms, such as One-Class SVM, Isolation Forest and Elliptic Envelope will be implemented on cleaned and transformed data. Then the trained models will be used to predict the remaining 2 days data which consists of all the implemented attack's instances as well as instances which are under normal conditions. Thereafter, the performance of individual models will be evaluated with the help of confusion matrix. This approach will be applied twice, each for Process 1 and Process 2 respectively.

IV. Experimental Work and Methodologies

The dataset used in the analysis does not have any labelled target variable, i.e this is an unsupervised learning problem. In addition to that, validation data is imbalanced. The stated attack detection problem is essentially an anomaly detection problem. Thus, the analysis can be narrowed down to an outlier/novelty detection problem where the model will be trained with normal data and will classify validation dataset either as inlier (normal) or outlier (abnormal)[1].

There are some unsupervised outlier detection methods in Scikit Learn (Machine Learning Library in Python) such as One-Class SVM, Isolation Forest and Elliptic Envelope [11]. These methods consist of some parameters that need to be altered and tuned for the most optimum performance. Training an entire dataset at once which has about 130 dimensions is time consuming and less accurate. This issue could be addressed by dividing the dataset into its subsequent processes and further analyzing them individually. Hence, Process 1 and Process 2 have been sliced from complete normal data. Process 3 is not taken into account as there is no attack implemented on it, plus it has very few process variables. Even after slicing, the number of dimensions in both process 1 and 2 were still significant. For that, Principal Component Analysis (PCA) technique is used for dimensionality reduction of data. The analysis is divided into 2 parts as below:

A. Process 1 (Primary Supply and Analysis)

Process 1 (P1) comprises 19 variables which includes different sort of equipments and meters such as motorized valve, level indicator, transfer pumps, physical properties indicators (eg. turbidity, conductivity, pH, ORP, TRC). With the help of an in-built module of PCA in Scikit Learn [8], a PCA function is defined with 95 percent explained variance. It means the number of principal components obtained after fitting, will altogether explain at least 95% variance of complete Process 1 variables. After fitting and transforming PCA function on P1's normal data, it was found that only 3 PCs explained around 99.8% variance.

```
In [12]: pca.explained_variance_ratio_
Out[12]: array([0.60433615, 0.27824022, 0.11554166])
```

Similarly, attack data is also transformed using the same PCA function and both normal and attack data of Process 1, after principal component transformation, were stored in data frames. The first two principal

[1] **NOTE:** All the outlier detection methods used in this report will classify inlier (normal) as 1 and outlier (attack) as -1

components of both the data frames were plotted to visualize the similarity and difference in the data points. The plot is shown below:

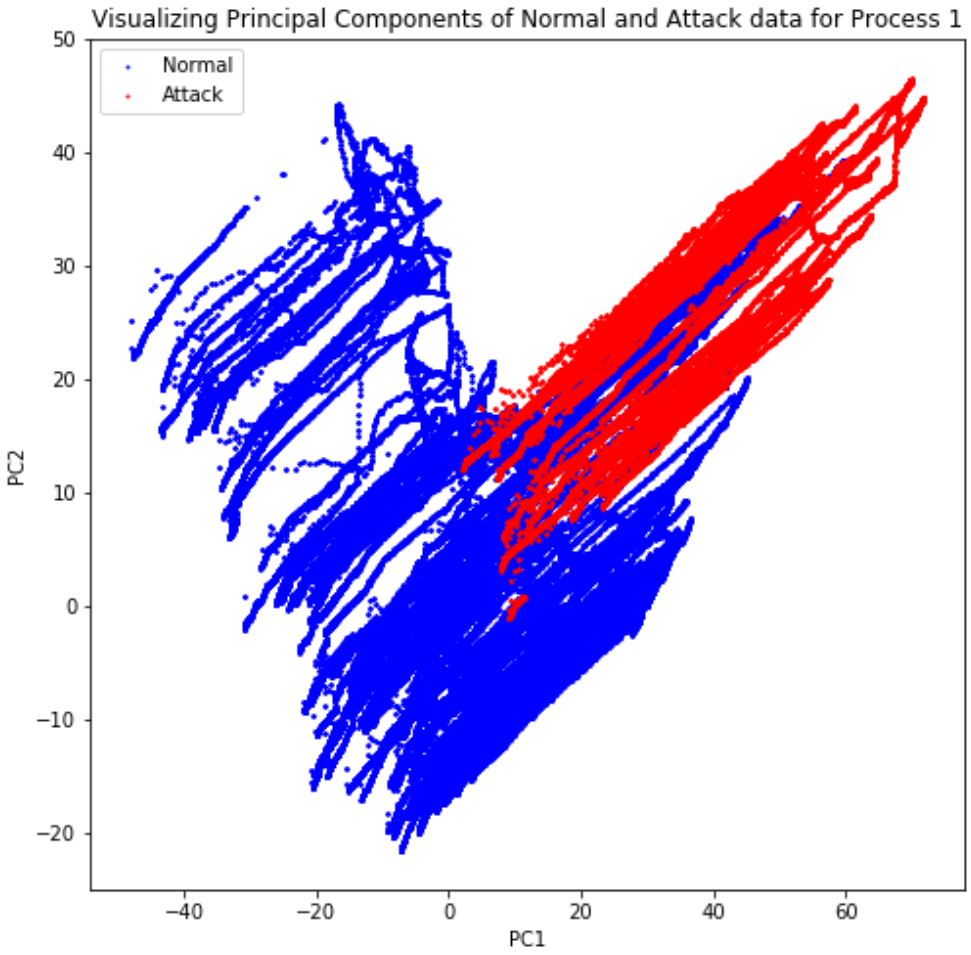

Fig 1 (Visualizing Principal Components for Process 1)

The above figure [Fig 1] was plotted after manual removal of some extreme outliers, for better scalability and visuals of plot. It is observed that a large proportion of attack data log is behaving as normal, which indeed, is as expected. The dataset is now prepared to train the models.

1) <u>One Class SVM (OC-SVM):</u>

The Scikit Learn module of OC-SVM has some parameters associated with it. One of them is *'nu'* which represents the fraction of training errors. Different experiments were conducted for different values of *nu*. In general, the trend found is, with decreasing the fraction of training errors, more data points in attack

data log has been identified as normal. Most appropriate case (*nu* = 0.0001) is chosen for further evaluation which identified 1,53,976 instances out of 1,72,795 as normal.

```
# Let's predict how many data points has been identified as normal in attack data.
print(np.count_nonzero(OC_SVM.predict(attack_PC)==1),'data points has been identified as normal')
```

```
153976 data points has been identified as normal
```

Furthermore, the points that have been detected as attacks, along with their 'date time' stamps, were plotted to understand the predictions of the model. That plot for different days is shown below:

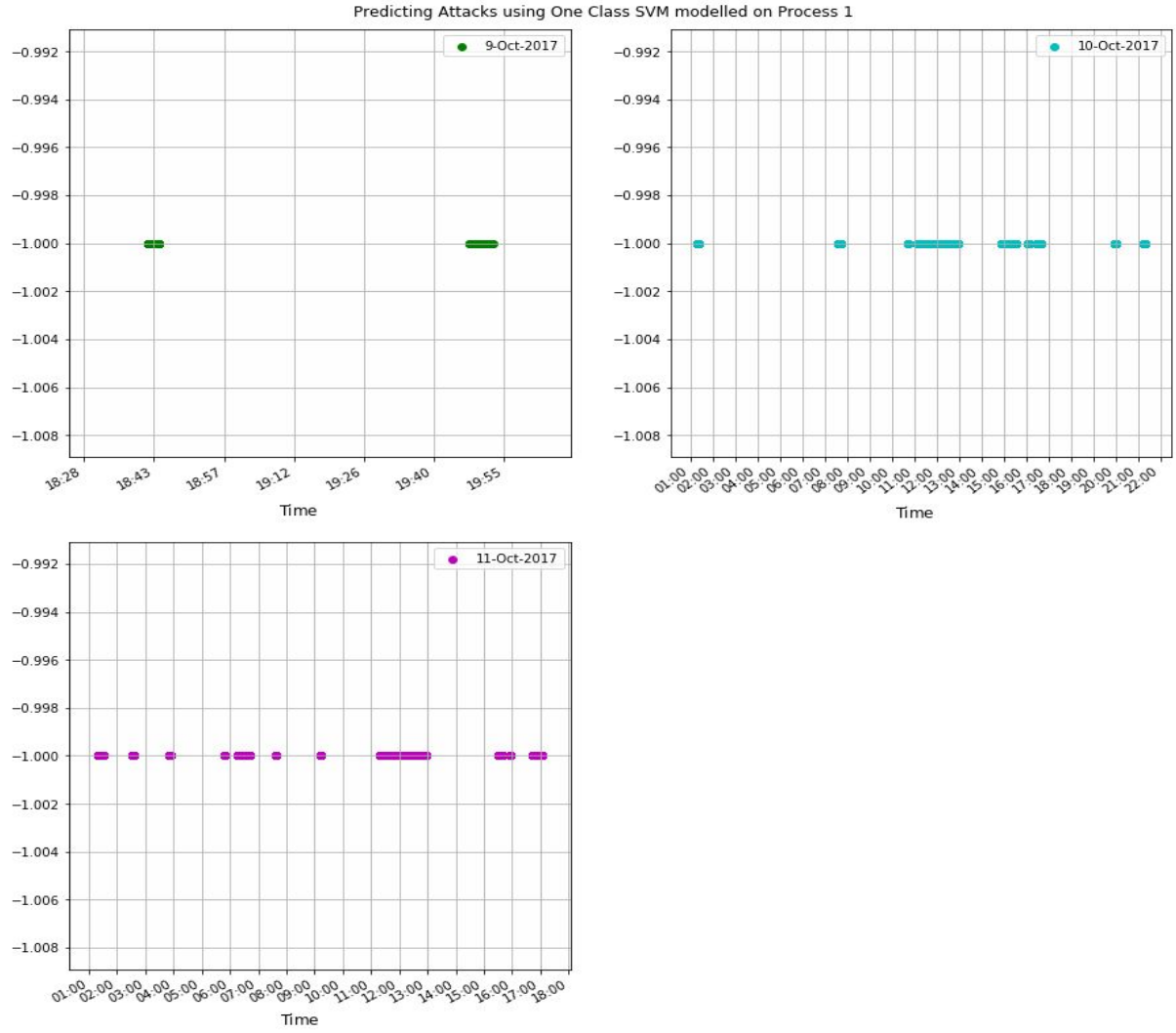

Fig 2 (OC-SVM predictions for Process 1)

A pattern of inconsistency in attack detection, false alarms, delayed detection can be observed in above plotted predictions of OC-SVM model [Fig 2].

There are some characteristics of OC-SVM model that are highlighted below:

i) Very specific to the range of training data variables:

Normal data for Process 1 had a discrepancy for a variable '1_LT_001_PV' which is the level indicator of Primary Tank 1 (Raw Water Tank 1). The readings of this indicator fluctuated very frequently, and even at many instances, found close to 100. Thus, to avoid redundancy in the model, training data were selected in such a way that the '1_LT_001_PV' variable had the range of 40-70.

ii) High sensitivity:

This argument can be supported by the predictions of the first attack. According to the Table 1, the details of first attack is as follows:

Attack Identifier	Starting Time	Ending Time	Duration (minutes)	Attack description
1	9/10/17 19:25:00	9/10/17 19:50:16	25.16	Motorized valve 1 MV 001 is maliciously turned on, this causes an overflow on primary tank **should reflect on 1LT001 and 1FIT001**

Table 1 (Description of First Attack)

Above description states that the attack started at 19:25:00 but the OC-SVM model started predicting the readings as anomaly close to 19:48:00, which is a lag time of more than 20 minutes. This delay occurred because the '1_LT_001_PV' indicator's reading, in attack log, crossed the value of 70 at 19:47:26 as shown in Table 2:

	Date	Time	1_LT_001_PV
6440	10/9/2017	7:47:20.000 PM	70.000
6441	10/9/2017	7:47:21.000 PM	70.000
6442	10/9/2017	7:47:22.000 PM	70.000
6443	10/9/2017	7:47:23.000 PM	70.000
6444	10/9/2017	7:47:24.000 PM	70.000
6445	10/9/2017	7:47:25.000 PM	70.000
6446	10/9/2017	7:47:26.000 PM	70.238
6447	10/9/2017	7:47:27.000 PM	70.238
6448	10/9/2017	7:47:28.000 PM	70.238
6449	10/9/2017	7:47:29.000 PM	70.238

Table 2 (Locating the time when level indicator value crossed 70)

Hence, it justifies the sensitivity of OC-SVM to the training set variable's range and its impacts on detection of abnormalities in the process.

2) Isolation Forest:

The IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature [12]. This method has two main parameters that need to be defined before training:

'contamination' - The proportion of outliers in the data set.

'n_estimators' - The number of base estimators in the ensemble.

In general, it is found that decreasing contamination increases the number of instances identified as normal (inlier). Though, increasing base estimators increases the number of inlier predictions, it also makes the algorithm computationally heavy which ultimately raises the training time. Therefore, based on the problem, an optimum balance between the parameters should be established. After conducting several experiments, an appropriate combination of *n_estimators* & *contamination* is found at 200 & 0.0001 respectively. This model identified 1,64,306 entries as normal out of 1,72,795.

The corresponding predicted attacks versus 'date-time' plot is shown below:

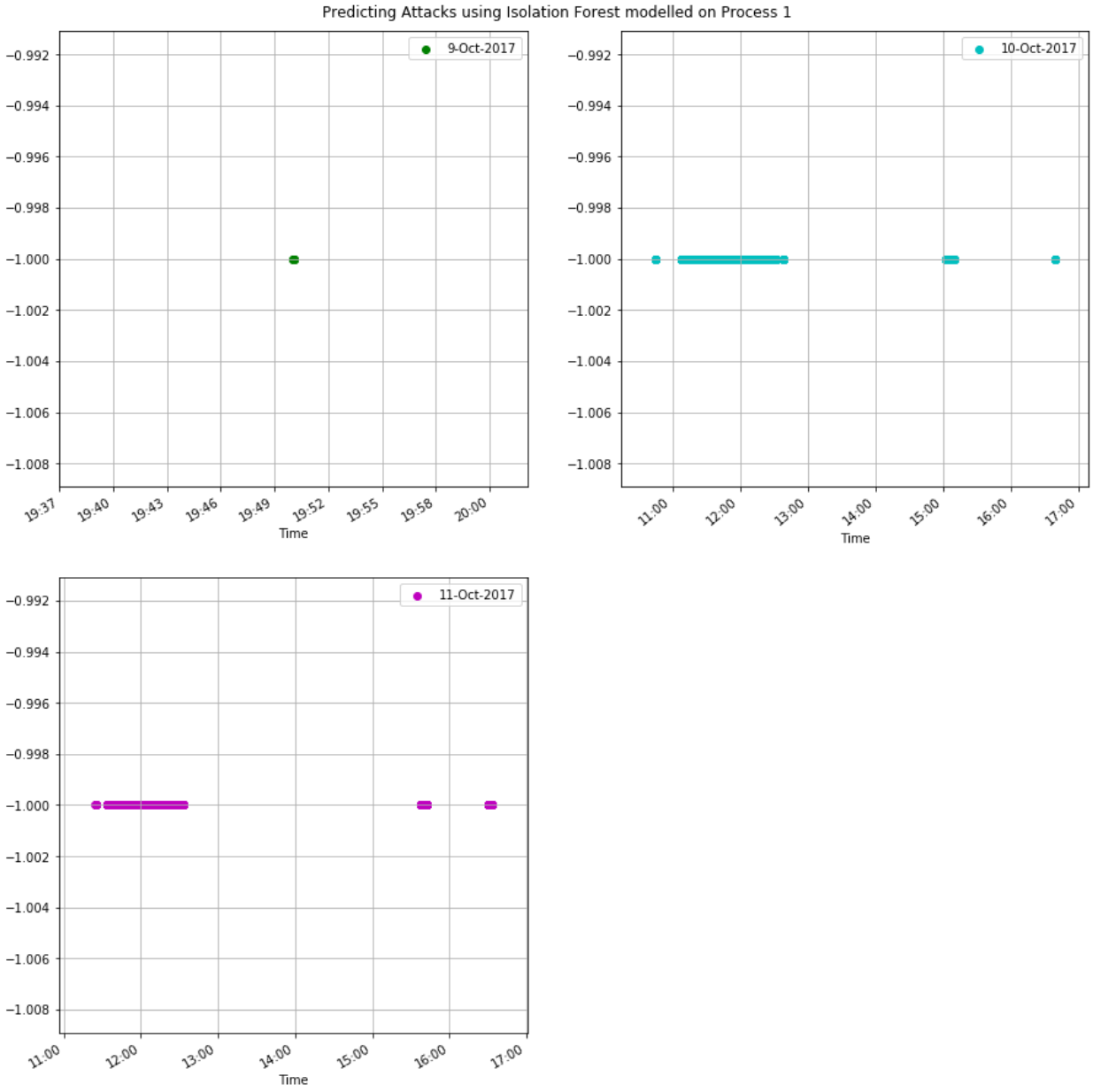

Fig 3 (Isolation Forest predictions for Process 1)

Some observed facts from above plot [Fig 3] are:

A. Isolation Forest predicted more instances as normal in comparison to the OC-SVM model (i.e, less number of false alarms).

B. Number of attacks predicted out of total attacks occured (Recall) is more in case of OC-SVM.

C. It appears that Isolation Forest is sensitive to only major deviations in the process variable's readings.

3) Elliptic Envelope:

This method is for detecting outliers in a Gaussian distributed dataset [13]. This technique is quite harsh and less effective for the current problem. It is recommended that this method might not be effective on large dimensional datasets [13]. Similar to Isolation Forest, this module too, has a contamination parameter that needs to be defined before fitting the model. The most suitable model was trained which predicted 1,57,496 data points as normal out of 172795.

The corresponding predicted attacks versus 'date-time' plot is shown below:

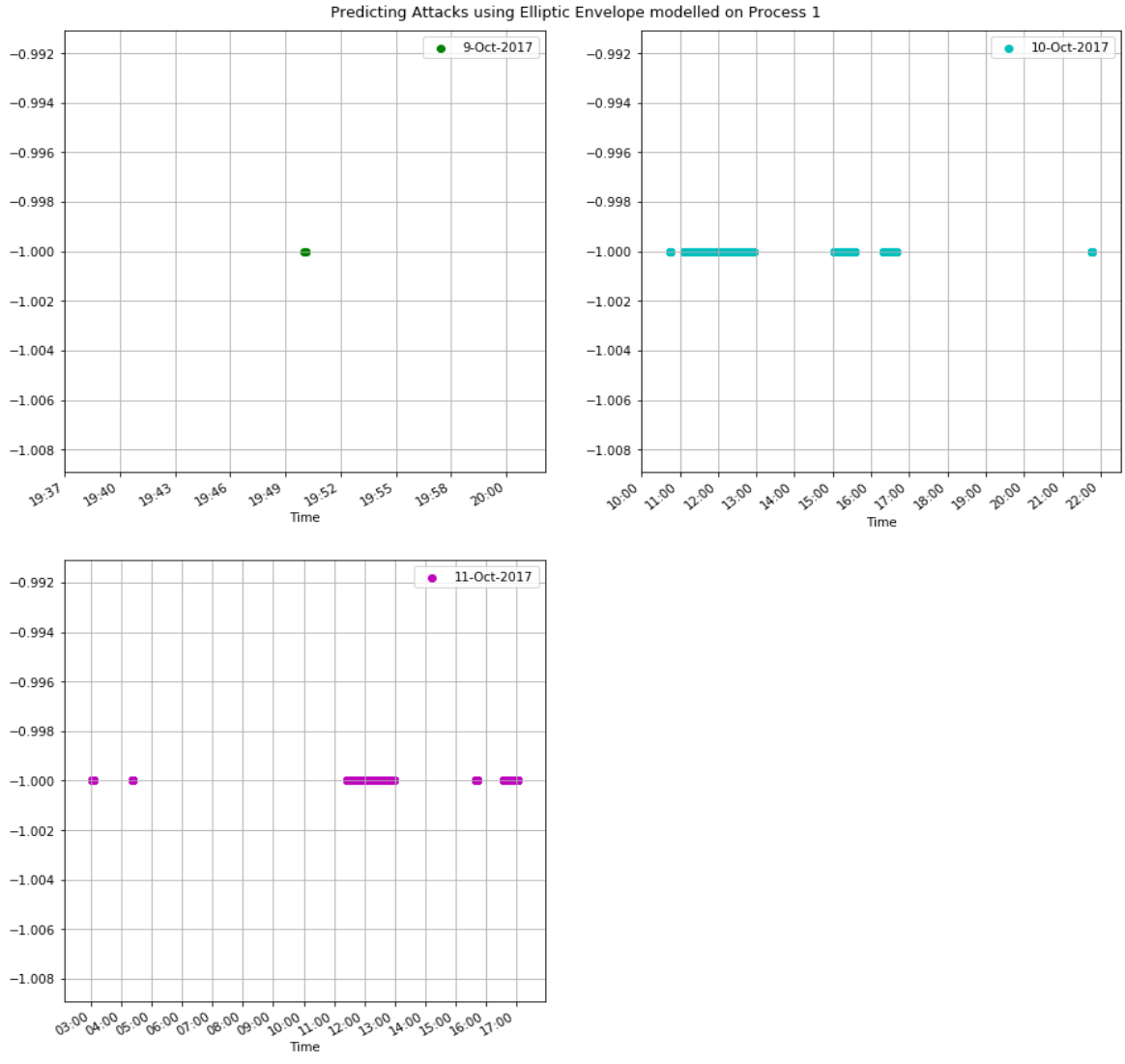

Fig 4 (Elliptic Envelope predictions for Process 1)

Insights, that can be obtained from above plots [Fig 4] are:

a. Elliptic Envelope and Isolation Forest performed almost equal in detecting the data points of 9-Oct-2017.
b. Elliptic Envelope performance is least efficient, both in terms of precision and recall, as compared to the other two processes.

B. Process 2 (Domestic Grid with Booster Pump)

The overall approach of analyzing Process 2 will be very much similar to what we did in Process 1. This process is the largest among these three and initially consists of about 86 variables in the raw format. However, after discarding some less effective variables, we are left with a total of 75 variables that need to be passed through the PCA function. Here again, PCA is fitted for explaining 95% of the variance which resulted in 7 Principal Components. Unlike Process 1, the range of the first two PCs in Process 2 is comparatively larger. Infact, PCA reduced normal and attack dataset, at first, doesn't represent much difference as shown in plot below [Fig 5]:

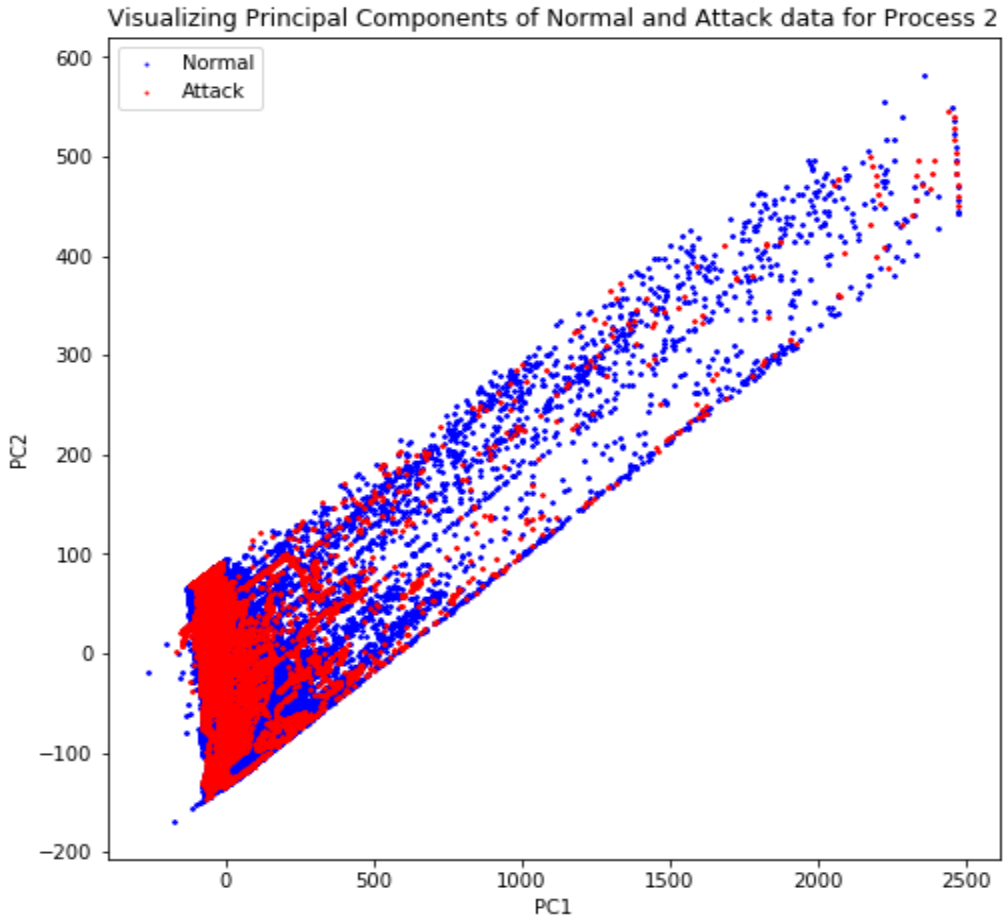

Fig 5 (Visualizing Principal Components for Process 2)

Training model on the above dataset is ineffective against attack detection. Hence, outliers were removed from the normal dataset while retaining 90-95 % of the information.

After filtering outliers from PCA reduced normal data, the plot obtained is as follows [Fig 6]:

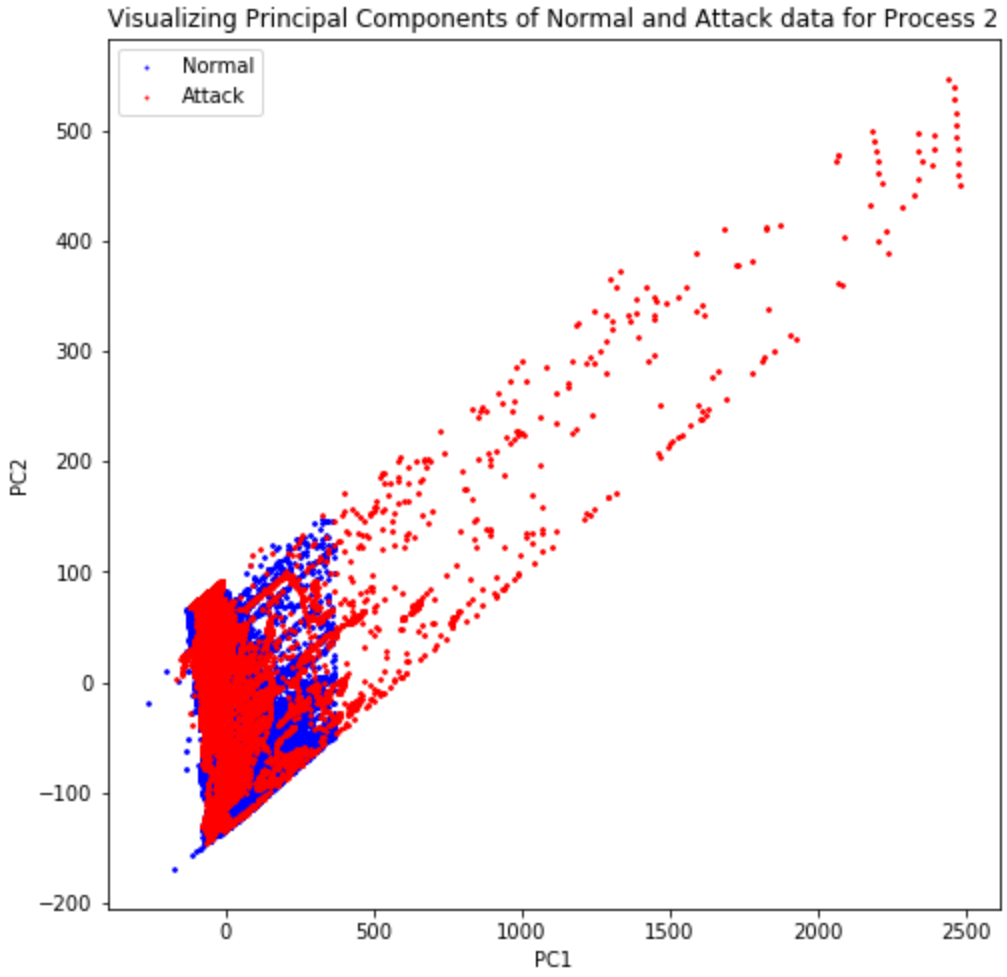

Fig 6 (Visualizing Outlier Removed Principal Components for Process 2)

Now, the normal data is eligible to be trained in different models. Beyond this point, the training and evaluation of the algorithms is conducted in the manner similar to Process 1.

1) <u>One Class SVM (OC-SVM):</u>

After several experiments, most optimum value for *'nu'* is obtained as 0.0001. The model predicted 1,58,073 instances as normal out of 172801.

```
# Let's predict how many data points has been identified as normal in attack data.
print(np.count_nonzero(OC_SVM.predict(P2_attack_PC)==1),'data points has been identified as normal')
```

```
158073 data points has been identified as normal
```

2) Isolation Forest:

Tuning the *'contamination'* and *'n_estimators'* parameters through repeated trials, optimum combination is found to be 0.0001 and 150 respectively (Infact *n_estimators* didn't have a significant impact once it crossed the value of 100). This model identified 1,65,087 instances as normal out of 172801.

```
# Let's predict how many data points has been identified as normal in attack data.
print(np.count_nonzero(Isolation_Forest.predict(P2_attack_PC)==1),'data points has been identified as normal')
```

```
165087 data points has been identified as normal
```

3) Elliptic Envelope:

The performance of this method on Process 2 variables is found to be least effective. This would be due to the Non-Gaussian distribution nature of the data. Nevertheless, the model was trained with *'contamination'* of 0.1 and it predicted 1,52,797 instances as normal out of 172801.

```
# Let's predict how many data points has been identified as normal in attack data.
print(np.count_nonzero(Elliptic_Envelope.predict(P2_attack_PC)==1),'data points has been identified as normal')
```

```
152797 data points has been identified as normal
```

V. Results and Discussions

After modelling of all three methods on both processes, their performance evaluation is carried out with the help of confusion matrix. ROC curve evaluation will be less effective in this case because the test data is imbalanced [15].

A confusion matrix is a summary of prediction results on a classification problem. Some important terms of confusion matrix are described below:

I. **Positive:** Prediction of instance as +1 (normal).
II. **Negative:** Prediction of instance as -1 (attack).
III. **True Positive(TP):** Observation is normal and predicted as normal.
IV. **False Positive(FP):** Observation is attacked but predicted as normal.
V. **True Negative(TN):** Observation is attacked and predicted as attack.
VI. **False Negative(FN):** Observation is normal but predicted as attack.

Main objective is to minimize the values of False Negative and False Positive while on the other hand to maximize True Positive and True Negative. Precision and Recall are the metrics for evaluation of classification algorithms's performance.

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP} \text{ (for positive cases)} = \frac{TN}{TN + FN} \text{ (for negative cases)}$$

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN} \text{ (for positive cases)} = \frac{TN}{TN + FP} \text{ (for negative cases)}$$

Precision is the computation of how precisely the model is correctly classifying observations and Recall is the calculation of the fraction of true classified observations to the total number of actual observations of that class (i.e how many attacked observations are detected out of total). Another important parameter for performance evaluation is the F-measure (or F1 score) which is the harmonic mean of precision and recall.

$$\textbf{\textit{f1-score}} = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

The summarized report of all 3 models for both processes is shown below (Table 3 & 4):

REPORT OF TRAINED MODELS FOR PROCESS 1			
One Class SVM			
	Precision	Recall	F-1 score
Attack (-1)	0.25	0.48	0.33
Normal (1)	0.97	0.91	0.94
Isolation Forest			
	Precision	Recall	F-1 score
Attack (-1)	0.39	0.33	0.36
Normal (1)	0.96	0.97	0.96
Elliptic Envelope			
	Precision	Recall	F-1 score
Attack (-1)	0.21	0.32	0.26
Normal (1)	0.96	0.93	0.94

Table 3 (Model's Summary for Process 1)

REPORT OF TRAINED MODELS FOR PROCESS 2			
One Class SVM			
	Precision	Recall	F-1 score
Attack (-1)	0.32	0.48	0.38
Normal (1)	0.97	0.94	0.95
Isolation Forest			
	Precision	Recall	F-1 score
Attack (-1)	0.34	0.26	0.3
Normal (1)	0.96	0.97	0.96
Elliptic Envelope			
	Precision	Recall	F-1 score
Attack (-1)	0.07	0.13	0.09
Normal (1)	0.94	0.89	0.91

Table 4 (Model's Summary for Process 2)

Observations from above tables (Table 3 & 4) are listed below:

- Elliptic Envelope isn't effective for any process and even poor for process 2.
- It is found that deciding the better method between OC-SVM and Isolation Forest for process 1 is grinding. Where Isolation Forest gave high precision, on the other hand, OC-SVM have shown high recall. The decision in such a situation is user, as well as, problem dependent. Based on the desired constraints, priority will be given to an algorithm.
- Somewhat similar ambiguity is also found for Process 2. However, OC-SVM had outrun Isolation Forest with some differences in this case.

NOTE: The evaluation of above methods has been conducted in strict manners. Delay in equipment readings has not been taken into consideration. Whereas, in real time scenario the effect of attack will be projected in equipment's readings after some response interval. If this argument is taken into consideration, then the precision, as well as recall for attack class will increase.

VI. Conclusions

Based on the above conducted analysis, it can be concluded that Machine Learning models could be installed as an additional security layer in order to detect abnormalities in the process. However, there are some points to be considered before and during training such models:

- ➢ These algorithms are sensitive to input data (training data). Thus, the user has to be very alert and conscious of their definition of normal operation, and accordingly, the range of variables should be assigned/selected in training data.
- ➢ The in-function parameters of these methods have a very critical impact on the predicted results. Thus, after iterative experimentation of training the models, a suitable combination of parameters has to be evaluated.
- ➢ Elliptic Envelope technique is suggested to be trained on low dimensional data which has the possibility of following Gaussian distribution.
- ➢ One Class SVM algorithm has a high sensitivity to outliers. Therefore, this algorithm is best, to be trained on an outlier removed dataset. That's a reason why this technique is best suggested for novelty detection problems.

VII. Future Prospects and Limitations

The study that is explored in above sections can be extended to different process industries by training the Machine Learning models on standard operation data of a system boundary. There is a potential in these models to be installed as a live monitoring setup which will take readings from sensors, pass through transformation functions and classify the state of process as normal or anomaly. In addition to the above discussed three strategies, Local Outlier Factor is another method available in Scikit Learn [14] which is yet to be tested for the purpose.

The limitations of the methodologies discussed in this report are: 1) As these are data-driven models the changes due to attack has to be forecasted in testing data. Therefore, attack detection would often face delayed response. 2) Being sensitive to training sets could sometimes behave as a limitation for the model and such models can possibly fail to identify a few attacks. 3) There is a likelihood that sometimes, fitting might turn out as overfit or underfit. Again, it will depend majorly on the selection of the training set.

Due to time and resource constraints, detailed study of individual attacks have not been possible in this work. We hope to explore more opportunities of anomaly detection in the coming future.

VIII. Acknowledgements

We are sincerely grateful to Indian Institute of Petroleum and Energy for providing us this opportunity in the form of a semester project. We would like to extend our thanks to the mentors: Dr Venkata Reddy Palleti and Dr Veerabhadra Rao Chandakanna for their continuous support and suggestions throughout the semester without which conducting this research wouldn't have been possible. In addition, we are hugely thankful to itrust Centre at Singapore Institute of Technology and Design (SUTD) for providing the WADI dataset for conducting the research.

References

1. Gharibian F, Ghorbani A.A , Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection, Proc. of the Fifth Annual Conference on Communication Networks and Services Research, 2007, pp. 350–358.

2. C. Fleizach and S. Fukushima, "A naive bayes classifier on 1998 kdd cup," 1998.

3. Domingos P. and Pazzani M., Beyond Independence: Conditions for the optimality of the simple Bayesian Classifier, In proceedings of the 13 th Intnl. Conference on Machine Learning, 1996, pp.105-110.

4. Langley P, Sage S, Induction of selective Bayesian classifiers, Proc. of the Tenth Conference on Uncertainty in Artificial Intelligence , 1994,pp. 399-406, Seattle, WA: Morgan Kaufmann.

5. Huy Anh Nguyen, Deokjai Choi, Application of Data Mining to Network Intrusion Detection: Classifier Selection Model, 2008, pp.399-408, Springer-Verlag.

6. Muda Z, Yassin W, Sulaiman M.N, Udzir N.I , Intrusion Detection based on k-means clustering and Naive Bayes classification, Proc. of 7 th Intl. Conference on IT in Asia, 2011, pp.1-6.

7. Wang, H., Zhang, G., Mingjie, E. *et al.* A novel intrusion detection method based on improved SVM by combining PCA and PSO. *Wuhan Univ. J. Nat. Sci.* **16,** 409 (2011).

8. Cheng Feng, Venkata Reddy Palleti, Aditya Mathur and Deeph Chana: A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems

9. S. Adepu and A. Mathur, "Using process invariants to detect cyber attacks on a water treatment system," in IFIP International Information Security and Privacy Conference. Springer, 2016, pp. 91–104.

10. ——, "From design to invariants: Detecting attacks on cyber physical systems," in IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C 2017). IEEE, 2017, pp. 533–540.

11. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

12. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html#sklearn.ensemble.IsolationForest

13. https://scikit-learn.org/stable/modules/generated/sklearn.covariance.EllipticEnvelope.html#sklearn.covariance.EllipticEnvelope

14. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html#sklearn.neighbors.LocalOutlierFactor

15. Saito T, Rehmsmeier M (2015) The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLoS ONE 10(3): e0118432. https://doi.org/10.1371/journal.pone.0118432