Detecting Anomalous Behavior of PLC using Semi-supervised Machine Learning

Ken Yau, KP Chow, SM Yiu, CF Chan
Department of Computer Science
The University of Hong Kong
Hong Kong, China
{kkyau, chow, smyiu, cfchan}@cs.hku.hk

Abstract— **Industrial Control System (ICS) is used to monitor and control critical infrastructures. Programmable logic controllers (PLCs) are major components of ICS, which are used to form automation system. It is important to protect PLCs from any attacks and undesired incidents. However, it is not easy to apply traditional tools and techniques to PLCs for security protection and forensics because of its unique architectures. Semi-supervised machine learning algorithm, One-class Support Vector Machine (OCSVM), has been applied successfully to many anomaly detection problems. This paper proposes a novel methodology to detect anomalous events of PLC by using OCSVM. The methodology was applied to a simulated traffic light control system to illustrate its effectiveness and accuracy. Our results show that high accuracy of identification of anomalous PLC operations is obtained which can help investigators to perform PLC forensics efficiently and effectively.**

Keywords— Programming logic controller, forensics, machine learning

I. Introduction

Industrial Control System (ICS) system is used to monitor and control industrial and infrastructure processes such as chemical plant and oil refinery operations, electricity generation and distribution, and water management [1]. If any undesirable incidents happened to the systems, it may hazard human's lives, cause serious damage to our environment and enormous financial loss. It is important to protect the systems from any undesired incidents such as hardware failure, malicious intruders, accidents, natural disasters, accidental actions by insiders [5].

Traditionally, the control systems have been operated as isolated systems with no network connection to the world. Threats against these systems were limited to physical damage attacks or data tampering that originated inside the system. Nowadays, such systems are connected to the corporate networks and Internet over TCP/IP and wireless IP for improving performance and effectiveness [2]. As a result, the closed systems have been exposed to various Internet threats and attacks.

Programmable Logic Controller (PLC) is an essential component of ICS. It is a special computer, which can be used to construct an automation system (from very simple one to a rather complicated one). An example of a simple automation system is Lighting Control System. The system is used to turn lights on automatically when the area becomes occupied and turn them off when the area becomes unoccupied. On the other hand, a group of PLCs can form a complex automation control system such as power generation system. PLCs in electricity generation system are responsible for automating numerous tasks that keep the electricity flowing to our home, offices and factories [3].

Because of the special architecture of PLC such as limited memory and proprietary operating system, it is difficult to apply contemporary tools and techniques for security protection and digital forensics. This paper proposes to adopt a semi-supervised machine learning algorithm, One-class Support Vector Machine (OCSVM), to detect PLC anomalous events. Although OCSVM has previously been applied successfully to anomaly detection problems such as detecting anomalous Windows registry accesses [25], it seems that it has not been used to detect PLC anomalous behavior. Compared to supervised machine learning, semi-supervised machine learning may be a better solution for PLC anomaly detection (see the followings for more elaboration).

In our experiment, we selected a popular PLC, Siemens Simatic S7-1212C, and set up a common critical PLC application: simulated traffic light control system. Anomalous operations of traffic light control system were created in order to prove the effectiveness and accuracy of the methodology. The proposed methodology is an initial step for us to create a generic model to detect anomalous behavior of any PLC and other control programs even with limited domain knowledge of PLC applications.

II. Programmable Logic Controller

Programmable Logic Controller (PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes (Fig.1) [4].

When designing and implementing control applications, PLC programming is an important task. All PLCs have to be

978-1-5386-0683-4/17/$31.00 ©2017 IEEE

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.

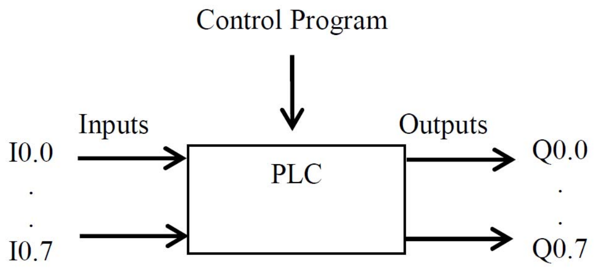


Fig. 1. Programmable Logic Controller

loaded with user program to control the status of outputs according to status of inputs. PLC can identify each input and output by address. For Siemens PLC, the inputs and outputs have their addresses in terms of the byte and bit numbers. For example, I0.7 is an input at bit 7 in byte 0 and Q0.7 is an output at bit 7 in byte 0.

A PLC generates anomalous operations in the following situations [15]: (i) hardware failure; (ii) incompatible firmware version; (iii) control program bugs created by an authorized programmer or attacker; (iv) stop and start attacks; and (v) memory read and write attacks. In order to detect these kinds of anomalous operations, we do the followings. We first capture relevant values of memory addresses used by PLC control program in normal situation. The captured values are used to train a model for the normal behavior of PLC using the semi-supervised machine learning. The trained model can be used to classify whether the PLC events are in normal operation or not.

To demonstrate our proposed methodology, we developed a control program by STEP 7 (Siemens programming software for S7 PLC programming, communication and configuration) for controlling traffic light control system (Fig. 2).

A. *Traffic Light Control System*

The setup of a simulated traffic light control system that we used in our experiment is shown in Fig 2. PLC Input I0.0 and I0.1 were connected with switches. PLC Output Q0.0, Q0.1, Q0.5, Q0.6, and Q0.7 were connected with lights. The traffic light control program (TLIGHT) was from the user guide “SIEMENS SIMATIC S7-300 Programmable Controller Quick Start” [6]. The control system is constructed by a set of instructions which are Inputs, Outputs, Memory Bit, and Timers. The instruction details are listed in Table I [6].

III. Challenges of PLC Protection and Forensics

Traditional tools and techniques are not easy to apply directly to PLCs for security protection and forensic investigation because of its unique architectures, such as special operating systems and limited memory [9]. For example, there is no software can be installed to PLC to prevent and detect malicious software. Followings are PLC forensic challenges [8]:

- Lack of documentation: Insufficient low-level documentation available for PLC with serious implications for forensic investigations.
- Lack of domain specific knowledge and experience: There is no comprehensive knowledge for performing PLC forensics.
- Lack of security mechanisms: No logging systems for security and forensic purposes.
- Lack of forensic tools: No dedicated forensic tools for PLC to perform a comprehensive investigation.
- Availability / Always-On: The availability of PLC in ICS environment is always top priority. Therefore, it is not easy to shut down a PLC for forensic investigation.

IV. Machine Learning

Machine learning is a method of data analysis. It builds an automated analytical model by using algorithms to learn from data iteratively. Based on the model, machine learning allows computers to find hidden insights without being explicitly programmed [10]. Supervised learning trains a model on known input and output data so that it can predict future outputs. Unsupervised learning finds hidden patterns or intrinsic structures in input data without knowing the corresponding labels of each input [11]. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data) [7]. One-class Support Vector Machine(OCSVM) is a semi-unsupervised algorithm.

A. *One-class Support Vector Machine (OCSVM)*

In machine learning, OCSVM is an One-class classification, also known as unary classification, tries to identify objects of a specific class amongst all objects, by learning from a training set containing only the objects of that class [13] (Fig. 3).

This paper utilizes OCSVM to train a model using data of normal situations (Training set), and classify PLC anomalous behavior that deviates from the trained model. This approach is suitable to deal with PLC anomalous behavior detection because OCSVM is suitable to deal with large amount of training data, since class labelling is not necessary. Also, it is relatively easy to gather training data of normal situations. On the other side, it is relatively difficult or impossible to collect data with a faulty system state. Even a faulty system state could be simulated, there is unlikely to guarantee that all the faulty state are simulated [12].

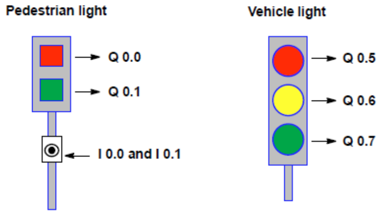


Fig. 2. PLC Inputs / Outputs connection with traffic lights

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.

TABLE I. INSTRUCTIONS OF TRAFFIC LIGHT CONTROL SYSTEM

Instruction	Address	Description
Outputs	Q 0.0	Red for pedestrians
	Q 0.1	Green for pedestrians
	Q 0.5	Red for vehicles
	Q 0.6	Yellow for vehicles
	Q 0.7	Green for vehicles
Inputs	I 0.0	Switch on right-hand side of street
	I 0.1	Switch on left-hand side of street
Memory Bit	M 0.0	Memory bit for switching the signal after a green request from a pedestrian
Timers (on-delay timer)	T 2	Duration (3 sec) of yellow phase for vehicles
	T 3	Duration(10 sec) of green phase for pedestrians
	T 4	Delay (6 sec) red phase for vehicles
	T 5	Duration (3 sec) of red/yellow phase for vehicles
	T 6	Delay (1 sec) next green request for pedestrians

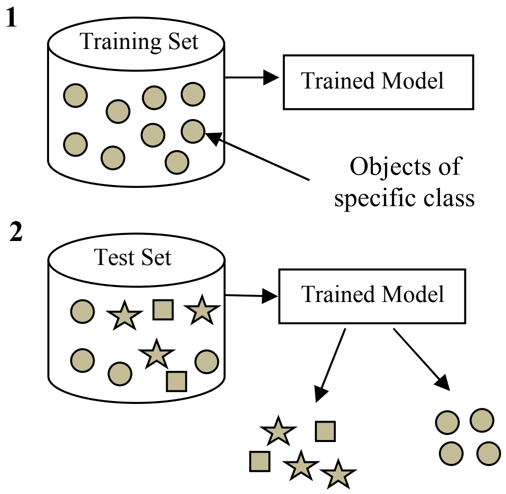


Fig. 3. One-class Classification

V. LITERATURE REVIEW

There are many research works focusing on ICS and PLC security protection and forensics after STUXNET malware attack discovered in 2010. STUXNET's target was to infect Siemens programming device (i.e., PC running Step 7 on Windows environment). The objective of the malware is to reprogram ICS by modifying code on the PLCs to make them work in a manner the attacker intended and to hide those changes from the operator of the equipment [17].

An example is the research work of Jamie et al. [22], they present a new methodology for the development of a transparent expert system for the detection of wind turbine pitch faults utilizing a data-intensive machine learning approach. The expert system for the classification and detection of wind turbine pitch faults, as validated by the 85.50% classification accuracy achieved.

Tina Wu and Jason Nurse have proved that PLC attacker's intentions can be determined by monitoring the memory addresses of user control program [16]. They identified the memory addresses used from the program code, and then monitored and recorded the values of the addresses by PLC Logger as a file (stored with normal PLC behavior). Based on the clear file, they can determine if the PLC is running normally or being attacked.

Ken Yau and KP Chow have proposed two solutions to perform PLC forensics. The first solution was that they developed a Control Program Logic Change Detector (CPLCD) [14]. It worked with a set of Detection Rules (DRs) to detect and record undesired incidents, the incidents were interfering with the normal operations of PLC. The DRs were defined based on the PLC user control program. CPLCD program worked with the defined DRs to monitor memory variables of the control program to detect "PLC Control Program Change Attack" and "PLC Memory Read and Write Logic Attack".

Their second solution was that, they proposed to capture values of relevant memory addresses used by PLC control program as a data log file. Based on the log file, supervised machine learning was applied to identify anomalous PLC operations [15].

All the solutions mentioned above are able to detect malicious behavior of a specific PLC, and some solutions use supervised machine learning. However, they are not generic solutions. Investigator must fully understand PLC control program logics before applying these solutions to determine anomalous PLC behavior. Since each PLC installed with different control programs for different applications and some programs are extremely complicated, therefore, investigators are not easy to apply the above solutions to the real PLC control systems. Furthermore, it takes time to label large set of training data when using supervised machine learning.

VI. EXPERIMENTAL SETUP AND METHODOLOGY

This section describes the experimental setup and the proposed methodology for identifying PLC anomalous operations.

A. Experimental Setup

The experiments used a Siemens S7-1212C PLC loaded with the traffic light control program (TLIGHT) (Section IIA). The values of relevant memory addresses used by TLIGHT were captured in a log file via a program using the libnodave open sources library [18]. In particular, the program monitored the PLC memory addresses over the network and recorded the values along with their timestamps. One computer was installed with Snap7 to create anomalous PLC operations by altering some values in address locations. Snap7 is an open source, 32/64 bit, multi-platform Ethernet communication suite for interfacing natively with Siemens

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.

S7 PLCs [19]. The overview of hardware experimental setup is shown in Fig. 4.

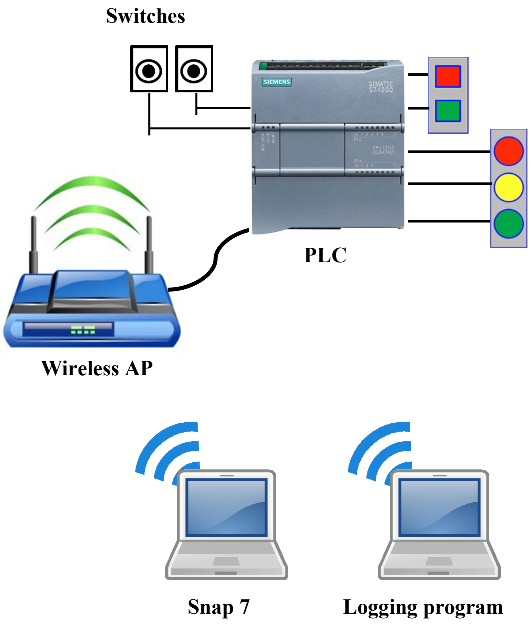


Fig. 4. Overview of Hardware Experimental Setup

B. *Classifying Anomalous Behavior*

A machine learning technique typically splits the available dataset into two components: (i) training set for learning the properties of the data; and (ii) testing set for evaluating the learned properties of the data. The accuracy of the response prediction was evaluated based on the testing set [23]. An overview of PLC anomaly detection using OCSVM is shown in Fig 5 and the details are as follows:

- Step 1: To set up a simulated traffic light control system.

  The setup details are shown in Fig. 2.

- Step 2: To collect values of relevant memory addresses used by PLC program.

  To capture the values of relevant memory addresses used by PLC program in a log file. (Fig. 6). The memory addresses of traffic light control system are shown in Table I. The captured data in the log file was used for OCSVM model training.

- Step 3: To normalize the collected values as training set.

  To simplify the semi-supervised machine learning process, all the non-binary values of memory addresses (e.g., timers) were converted to binary values.

- Step 4: To train an OCSVM model by using the normalized values.

  To train a learning model, One-class SVM (sklearn.svm.OneClassSVM) of Scikit-learn is adopted. Scikit-learn is a free software machine learning library for the Python programming language [20]. Based on the training set of the captured data, OCSVM was applied to train a model. There are four kernel functions used in OCSVM which are Linear, Polynomial, Gaussian, and Sigmoid/Logistic. The kernels are functions used to define a similarity measure between two data points.

  After comparing the performance of the four kernel functions in our experiments, we found that the kernel Polynomial function provided higher accuracy of classification for the simulated traffic light control system.

  Polynomial Kernel:

  K(x,y) = (gamma*x*y + coef()) ^ degree, the parameter settings are shown in Table II.

- Step 5: To create and collect PLC anomalous events for performance evaluation of the model.

  One computer was installed with Snap7 to create anomalous PLC operations by altering some values in address locations.

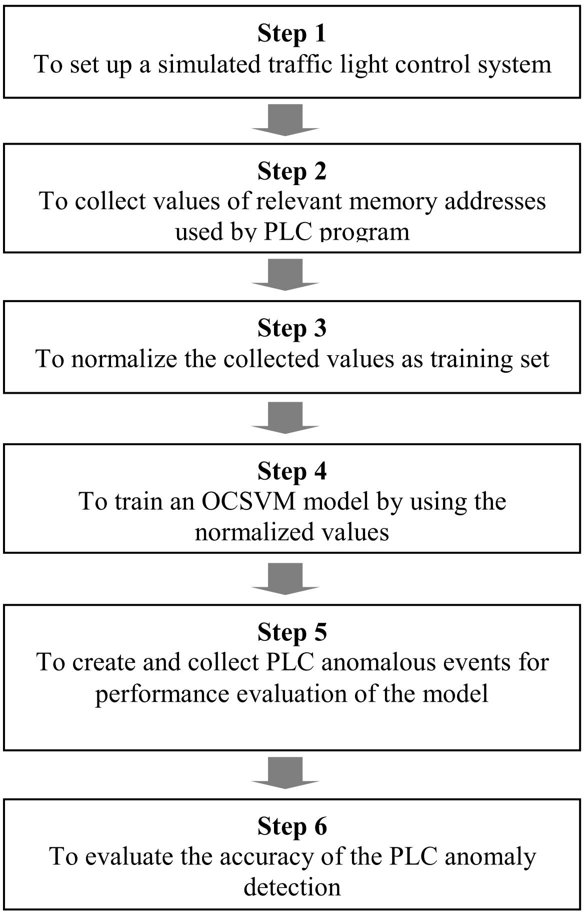
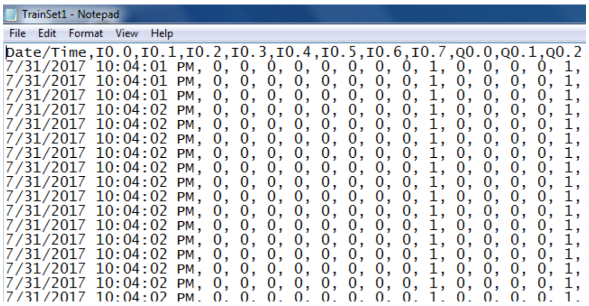

Step 1
To set up a simulated traffic light control system
Step 2
To collect values of relevant memory addresses used by PLC program
Step 3
To normalize the collected values as training set
Step 4
To train an OCSVM model by using the normalized values
Step 5
To create and collect PLC anomalous events for performance evaluation of the model
Step 6
To evaluate the accuracy of the PLC anomaly detection

Fig. 5. Overview of PLC Anomaly Detection using OCSVM

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.



```
Date/Time,I0.0,I0.1,I0.2,I0.3,I0.4,I0.5,I0.6,I0.7,Q0.0,Q0.1,Q0.2
7/31/2017 10:04:01 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:01 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:01 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
7/31/2017 10:04:02 PM, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
```

Fig. 6. Data log file

TABLE II. INPUT PARAMETER SETTINGS OF SCIKIT-LEARN ONE-CLASS SVM (OCSVM)

Para-meter	description	Value
degree	Degree of the polynomial kernel function	3
coef0	coefficients	4
nu	An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. Should be in the interval (0, 1].	0.1
gamma	gamma defines how much influence a single training example. The larger the gamma is, the closer other examples must be to be affected.	0.1

Test sets were created by capturing the values of the PLC memory addresses while performing the simulated attacks. The test sets contained normal and anomalous PLC events.

- Step 6: To evaluate the accuracy of the PLC anomaly detection.

  To evaluate the accuracy of the One-class SVM classification, one training set and three test sets were collect from the simulated traffic light control system. The trained model was evaluated by sklearn.metrics [24] and the classification results with five performance metrics are shown in Table III. The brief descriptions of the metrics are as follows:

  Accuracy: The accuracy is the ratio (tp + tn) / (p + n) where tp is the number of true positives and fn is the number of false negatives. P is the number of real positive cases in the data and n is the number of real negative cases in the data.

  Precision: The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp is the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.

  Recall: The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn is the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0.

  F1: Score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

  AUC: Area Under the Curve (AUC) is prediction scores which measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test.

VII. DISCUSSION

In the experiment, we made an assumption that the training set data collected from the traffic light system was in normal operations (without any anomalous events). This assumption is not unreasonable as we can collect data of normal behavior of the PLC during testing and maintenance. From the experimental results, high accuracy and high AUC of PLC anomalous operation detection were obtained.

Since our logging program captures memory addresses of PLC with time stamps, OCSVM together with the time stamps information can help forensic investigators to carry out investigation efficiently. OCSVM was able to detect the simulated traffic light anomalous behavior in a dataset after OCSVM model was trained. Since each dataset was recorded with time stamps, we could know the date and time about the PLC anomalous events. According to the time stamps and the values of memory addresses in the dataset, the scope of investigation can be narrowed down. For example, if any firmware or user control program was updated, or any attack during a particular period of time, the proposed solution can identify the date and time about the anomalous events.

According to the experiments, we found that it is important to select a correct kernel function with appropriate values of the function parameters in order to obtain a more accurate result of PLC anomaly detection. In our experiments, we chose kernel function Polynomial and adjusted the parameter values as Table II for classifying anomalous operations of the simulated traffic light system. As different control systems have different operational behavior, therefore, we believe that kernel type and values of parameters may be different for different kinds of PLC control systems.

Comparing with supervised machine learning for PLC anomaly detection, OCSVM may be a better solution when the training set data is large and complicated because the training data for OCSVM is not necessary to be labelled. Class labelling is not an easy task for large set of data

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.

because it is time consuming and always need to be performed by control system's experts.

VIII. Conclusion and Future Work

To overcome the challenges of PLC protection and forensic investigation, this paper proposes to use semi-supervised machine learning, One-class SVM (OCSVM), to detect PLC anomalous behavior based on the captured values of PLC memory addresses. Our experiment demonstrates that our solution is feasible and practical to apply to the traffic light control system. This paper is an initial step of applying semi-supervised machine learning for PLC anomaly detection. In future, we will evaluate the feasibility and increase the accuracy to detect PLC anomalous behavior by applying semi-supervised algorithm on various PLC applications in ICS. In addition, we will try to create a generic model for PLC anomaly detection even when the PLC control program is not provided.

TABLE III. OCSVM Classification Results of Traffic Light Control System

	No. of Rec	Accuracy	Precision	Recall	F1	AUC
Training Set	41580	0.96	1	0.96	0.98	n/a
Test Set 1	5000	0.78	1	0.78	0.88	0.89
Test Set 2	7000	0.75	1	0.75	0.86	0.83
Test Set 3	13130	0.82	1	0.82	0.90	0.88

References

[1] Irfan Ahmed, Sebastian Obermeier and Martin Naedele, Golen G. Richard III: SCADA System: Challenges for Forensics Investigations, IEEE Computer, Vol. 45 No. 12, pp 44–51, USA, 2012.

[2] T. Spyridopoulos , T. Tryfonas , J. May ,Incident analysis & digital forensics in SCADA and industrial control systems, System Safety Conference incorporating the Cyber Security Conference, 8th IET International, 2013.

[3] Dillon Beresford, Exploiting Siemens Simatic S7 PLCs, Black Hat USA, 2011.

[4] W. Bolton, Programmable Logic Controllers (4th Edition), 2006.

[5] Keith Stouffer,Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, Adam Hahn, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82 Revision 2, U.S. Department of Commerce, 2015.

[6] Siemens, SIMATIC S7-300 Programmable Controller Quick Start, Primer, Preface, C79000-G7076-C500-01, Nuremberg, Germany, 1996.

[7] Semi-supervised learning (https://en.wikipedia.org/wiki/Semi-supervised_learning), 2017.

[8] H. Patzlaff, D 7.1 Preliminary Report on Forensic Analysis for Industrial Systems, CRISALIS Consortium, Symantec, Sophia Antipolis, France, 2013.

[9] Fabro, M: Recommended Practice: Creating Cyber Forensic Plan for Control Systems, Department of Homeland Security (2008), Idaho National Laboratory (INL), USA, 2008.

[10] Machine Learning: What it is and why it matters (www.sas.com/it_it/insights/analytics/machine-learning.html), 2017.

[11] Machine Learning in MATLAB (www.mathworks.com/help/stats/machine-learning-in-matlab.html), 2017.

[12] Introduction to One-class Support Vector Machines (rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/), Last accessed on 2 May 2017, 2017.

[13] One-class classification.com (en.wikipedia.org/wiki/One-class_classification), 2017.

[14] Ken Yau and Kam-Pui Chow, PLC Forensics based on control program logic change detection, Journal of Digital Forensics, Security and Law, Vol. 9(2), 2015.

[15] Ken Yau and Kam-Pui Chow, Detecting Anomalous Programmable Logic Controller Events using Machine Learning, (to be appeared in the proceedings of) The 13th Annual IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL., February 2017.

[16] Tina Wu and Jason R.C. Nurse, Exploring the use of PLC debugging tools for digital forensic investigations on SCADA system, Journal of Digital Forensics, Security and Law, Vol. 9(2), 2015.

[17] Nicolas Falliere, Liam O Murchu, and Eric Chien: W32.Stuxnet Dossier, Version 1.4, Symantec Corporation, 2011.

[18] T. Hergenhahn, libnodave (sourceforge.net/projects/libnodave), 2014.

[19] D. Nardella, Step 7 Open Source Ethernet Communication Suite, Bari, Italy (snap7.sourceforge.net), 2016.

[20] sklearn.svm.OneClassSVM (scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html), 2017.

[21] Novelty and Outlier Detection (scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection), 2017.

[22] Godwin, J.L. and Matthews, P.C. and Watson, C., Classification and detection of electrical control system faults through SCADA data analysis, in Chemical engineering transactions. Volume 33. , pp. 985-990, 2013.

[23] scikit-learn Project, An Introduction to Machine Learning with scikit-learn (scikit-learn.org/stable/tutorial/basic/tutorial.html), 2016.

[24] sklearn.metrics: Metrics (scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics), 2016.

[25] Katherine Heller, Krysta Svore, Angelos D. Keromytis, Salvatore Stolfo, One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses, Columbia University Academic Commons, 2003.

Authorized licensed use limited to: Duale Hochschule Mannheim (FH). Downloaded on March 14,2023 at 11:33:52 UTC from IEEE Xplore. Restrictions apply.