UNIVERSITY OF AMSTERDAM

Advantages of Anomaly Detection Between the Controlling Unit and the Process Devices of an Industrial Control System

Boukema, Anouk
anouk.boukema@os3.nl

Lahaye, Rick
rick.lahaye@os3.nl

February 12, 2017

Supervisor:
Dima van de Wouw
dvandewouw@deloitte.nl

Abstract

Industrial Control Systems (ICSs) have recently been embedding common IT solutions for cost-performance reasons. This made them more accessible for the outside world, and more prone to its problems. As malware becomes more advanced and selective[4], this research proposes anomaly detection at the direct data going into- and coming from the process devices. This results in an Intrusion Detection System (IDS) which monitors independent, raw and integer data. This IDS contains unsupervised machine learning in order to inspect different types of data and is implemented on a relatively cheap Raspberry Pi 3. Therefore, it creates the possibility to implement this in front of every process device in an ICS and achieving overall anomaly detection. The advantages of this IDS are examined through a literature study and answer the research question: *"What are the advantages of anomaly detection between the controlling unit and its process devices?"*. Thereafter, the feasibility of the IDS is tested in the proof of concept and proven by the results. Further research should be conducted to transform this suggested IDS into a business case. Hence, the IDS can be a positive addition to the currently available security solutions.

Contents

1 Introduction

Historically Operational Technologies (OT) within a factory were mainly proprietary optimized for the specific production. Because of the constant need to improve on cost-performance, this production process started embedding more general IT solutions to achieve broader monitoring, connectivity, and interoperability within the Industrial Control Systems (ICSs). Therefore, less human interaction was needed and the production relied mostly on its control systems, process devices and standardized communication protocols over the network. This however opened these systems to security issues known to the Internet nowadays [27].

The main priority of an ICS is the constant availability of the system. If downtime occurs, it could result in financial loss, reputational damage or human injury.

Intrusion in these systems may affect the availability and functioning of an ICS by inserting malware, which could alter the actuators behavior.

These actuators create and provide the final product of an ICS. In the *Purdue Model for Control Hierarchy* they are segmented in the lowest level and called process or level 0 devices. In this paper, we will refer to the levels of the Purdue model in the same way as shown below in figure 1.

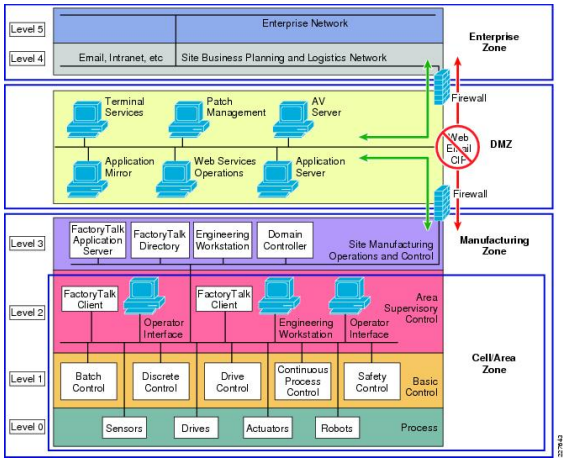

Figure 1: Purdue Model for Control Hierarchy

There is a broad variety of security available for ICSs nowadays offered by external security companies. They mainly supply IT security like firewalls and Intrusion Detection Systems, which is usually implemented at level 2 or higher, as this is where the data of the lower devices will congregate.

As malware becomes more advanced and selective[4], security also needs to evolve and explore new protection possibilities. This resulted in the research question: *"What are the advantages of anomaly detection between the controlling unit and its process devices?"*. If any advantages are found, an IDS will be proposed and tested in a proof of concept. Since many process devices are present in an Industrial Control System, multiple IDSs need to be present to achieve overall detection. Therefore the goal of the proof of concept is to create a cheap and efficient IDS that is capable of analyzing different types of data. If this is the case, an IDS at this lower level would be a positive addition to the currently available security solutions.

2 Literature Study

Attacks on Industrial Control Systems (ICS) are not uncommon and are becoming more feasible to perform nowadays as they are more often connected to the internet and the global IT infrastructure[12]. Some well-known attacks that were done are Stuxnet, Ukraine Power Grid attack, and PLC-blaster.

Stuxnet is a malicious computer worm that specifically attacks Programmable Logical Controllers. When infected it is able to reprogram the PLC, spread among other Programmable Logic Controllers, and propagates in the network. Initial introduction into a targeted network is done with the use of an infected USB stick. This was also the case with the attack on the Iran's Nuclear program where Stuxnet was used to do damage to the fast-spinning centrifuges to tear themselves apart[4].

The Ukraine Power Grid attack was an attack on Ukrainian energy companies that left 230.000 people without power for up to 6 hours. The initial intrusion into the targeted network was done with the use of stolen credentials received by phishing emails. Hackers then took over the SCADA network by accessing the Human Management Interface (HMI) systems remotely which resulted in the shutdown of 230

power switching stations, damage to critical infrastructure components, and the destruction of logs and files on systems[32].

PLC-Blaster is a worm designed to attack Siemens S7 PLCs. This worm attacks PLCs by using the PLC's management console portal. Access could be gained due to the fact that integrity safeguards were missing in the management console software by default, this allowed hackers to read, write, and modify the code. This resulted in hackers rewriting passwords, and being able to modify the program[21].

2.1 Detection along with Prevention

These three examples all target the controlling devices of an Industrial Control System. Relating this to the Purdue Model in figure 1, the control devices at level 1 and higher are tampered with and are sending out falsified data to the supervisory systems. Therefore these higher systems will not notice the changes made to the controller by an intrusion. The changes done to the controlling devices will only be noticed once the final product contains malfunctions. Examples of malfunctions are a broken centrifuge, power outage, or a more general example is a production batch that needs to be revoked after distribution.

An example of a malfunction that resulted in an altered final product was the callback of Chocolate bars by Mars[16]. This callback was initiated after a customer found plastic in a Mars Chocolate bar. As the company could not be certain that there was no plastic in any other bars from this product line, a callback was initiated in 55 countries which resulted in an estimated loss of tens of millions of dollars.

To prevent malfunctions, and the hacks explained previously, security is needed within Industrial Control Systems. If security is in place, this is often placed at central points where traffic flows by[18]. The advantage of this placement is that a security device like an Intrusion Detection System (IDS) is capable of inspecting traffic of multiple devices. The disadvantage of this placement is that the validity of this traffic is dependent on the lower level devices that forward this traffic upwards, and the data it monitors is an abstraction of the real data used to create the final product. The result of this disadvantage is that it might not be noticed when level 1 devices sent out falsified data.

When detection is placed before level 1, and therefore not dependent on any other levels, it can capture the raw data instead of the summarized data that is sent upwards by level 1 devices. This data is integer as it contains the direct input to the process devices like actuators and sensors, and therefore the direct cause of the quality of the final product.

To answer the research question *"What are the advantages of anomaly detection between the controlling unit and its process devices?"*. The data flowing between these two levels is raw and integer data. Meaning that this is the only level where data cannot be falsified, and if anomaly detection is implemented well, it is possible to detect malfunctions at an early stage.

2.2 Related Work

Summarizing, the succeeded hacks described previously, suggest that many Industrial Control Systems do not have an IDS monitoring the data below the control devices like the PLC. Whether this IDS might already be implemented by modern security supplies or not is further investigated. The assumption is made that the most know modern security suppliers are supporting the "2016 Industrial Control Systems (ICS) Cyber Security Conference"[17]. For each company supporting this conference, its closest security and detection solution to level 0 is investigated by looking at their portfolio. Many of these companies do not supply any detailed information on how they approach security between layer 1 and 0. Therefore, only the companies which do indicate their methods are summarized below[1].

Claroty According to their whitepaper[5] they provide monitoring over level 1 to level 4. The closest monitoring towards level 0 they provide is between level 1 and 2 where they monitor authentication with the PLC and its actions with use of the Modbus protocol.

[1] investigated as well: Ultra Electronics 3eTI, Honeywell, Nexdefence and NEXTNINE Connected ICS

Waterfall Waterfall provides security for ICS environments. They use Unidirectional Gateways as they are more secure then Firewalls due to the widespread deployment of it[30]. This gateway operates between level 3 and 4. Also, Waterfall is able to inspect Modbus traffic from PLCs at level 1 to higher devices[31].

BEDROCK open secure automation BEDROCK shows in its whitepaper that it provides security by refurbishing the whole ICS. They start from the bottom by first replacing all plastic module housing by metal and then sealing the metal constructs, like PLC's. Then provide secure communication, secured firmware, insert an anti-tamper component and secure everything with proven secure encryption and hashing algorithms[25].

Check Point Checkpoint provides an Intrusion Prevention System with a built-in VPN solution that is used to secure traffic from the plant to the enterprise. They have support for most of the industrial traffic protocols including Modbus, therefore is able to inspect traffic coming from the PLC[22].

Indegy Indegy also provides a solution that supports monitoring communication protocols like Modbus. This gives the possibility to monitor the traffic between level 1 and 2[13].

DarkTrace DarkTrace uses a machine learning algorithm to detect emerging cyber-threats, from within the network. According to DarkTrace this appliance is placed between level 3 and 4[6].

Of the above listed companies, none of them provide any details if they provide security[2] between level 0 and 1 or not. Most of them provide some sort of security like an Intrusion Detection System or Firewall that is able to inspect common protocols coming from the levels below[10][17]. Though no specific details could be found at which level this security is placed.

Reasons for the limited details regarding their security solutions could be a competitive advantage over other companies, thus releasing details could impact this advantage. Also, these security companies are often hired by organizations and governments, and releasing these details could result in a potential risk to their production or critical infrastructures.

2.3 Anomaly Detection Techniques on raw ICS data

As no clear indication could be found of an Intrusion Detection System between level 1 and 0, its possibility will be further researched in this paper. First of all, it is important to understand what kind of data an IDS between level 1 and 0 might have to deal with. The in- and output of a PLC and its process devices can either be analog, digital discrete or digital logic signals and are usually a combination of the last two[1].

When assumed that a regular PLC in an Industrial Control System has to deal with both discrete and logic signalling, a proposed IDS would require the ability to inspect and understand these signals. The purpose of inspecting these signals is to find anomalies in the Industrial Control System. In order to find anomalies, the IDS has to be able to compare the input with knowledge on what the data should look like when no anomalies would occur. Therefore, it needs to have an idea of the normal behavior of the in- and output data. Assuming that an ICS produces a specific product over and over again, the conclusion is drawn that the in- and output signals will conform to some iterative pattern. When the IDS would learn the patterns under which these signals conform, it can compare new data with this learned pattern and conclude whether this new data conforms to it or not. If it does not conform, it should be classified as an anomaly.

Within the field of anomaly detection there exist two types of anomalies: point anomaly and contextual anomaly[3]. Point anomaly is the simplest type of anomaly to detect as it counts for only one data point that needs to be compared with the rest of the data. For example, the maximum average temperature in the Netherlands in Juli is 22 degrees, if one day the maximum is 35 degrees this can be seen as an

[2] apart from physical security

anomaly. Point anomalies can be found with the use of classifications and boundary setting.

Finding contextual anomalies is a more difficult task. Since detection can only be done when the new data point does not seem to relate to the rest of the data, also known as the context. Usually, contextual anomalies can be found within time series or spatial data. An example of this can be seen in figure 2 where the temperature at t_2 does not exceed the maximum or minimum over the year but is not conform to the context of the rest of the data. Therefore t_2 can only be detected as a contextual anomaly and not as a point anomaly.

In order for an IDS to find the point and contextual anomalies at the repeating of an ICS its process device data, there are a few techniques possible[8]. When the specific pattern is known to the user, knowledge-based techniques can be used where the boundaries are set manually. This technique is data specific and might therefore not be the best solution when proposing an IDS for Industrial Control Systems in general.

More generic techniques need to build up knowledge so they can be applicable for many kinds of data input. For building up this knowledge they need to train on training data. Collecting training data in an ICS can be done by storing the data flowing through the IDS for several minutes, hours, or days dependent on how often the production process iterates in time. With this training data, and the fact that the data is autocorrelated (conform to a pattern), the best techniques are time series classification and recurrent neural networks[20]. The gathered training data might not always contain autocorrelated data. If so, then simple and less computational heavy classifiers can be used[20].

3 Proof of Concept

To proof, the feasibility of an Intrusion Detection System between level 1 and 0, a proof of concept is created. This proof of concept has to meet the following requirements in order to be a realistic example for an implementation within an ICS:

1. **Operates real-time** in order to detect anomalies as soon as possible and therefore decrease damage costs[16];
2. **Detects point and contextual anomalies** as explained in Chapter 2.3;
3. **Affordable** in comparison to general security suppliers and the possible prevented damage costs;
4. **Generic** so it will be able to monitor data signals from several types of process devices.

For this proof of concept a water thermostatic continuous closed loop environment is chosen[29] that can be seen in figure 3. The reason for choosing this environment is that it represents a generic closed circuit process control setup. Therefore a realistic simplification of the lower level devices in an Industrial Control System.

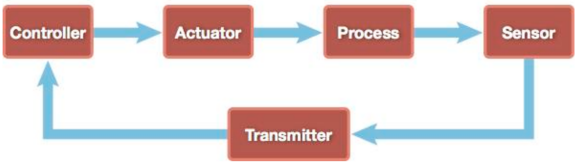

Figure 3: closed loop

3.1 Setup Components

The components used in this Proof of Concept are a digital temperature sensor[7] and 12 Volt heatercartridge[11] as a sensor and actuator. The sensors output is a discrete digital signal, and the heater input a logic digital signal. The combination of these two signals, as mentioned in Chapter 2.2, is a realistic comparison of the regular in- and output of a PLC. The controlling unit used is a Raspberry Pi which regulates the temperature by running a script. This script is based on the same kind of syntax as ladder logic for a PLC, meaning it only uses boolean logic [14]. For the IDS a Raspberry Pi 2 model B was used. It has a 900MHz quad-core ARM Cortex-A7 CPU and 1GB of RAM[23].

This Raspberry Pi was chosen because of its performance its relatively cheap price (req. 3) and the good support and available drivers (req. 4). It is running Raspbian GNU/Linux 8 (Jessie) as the operating system, which is officially supported by the Raspberry Pi foundation[24], and Python 2.7.9. This Raspberry Pi has 40 General Purpose Input Output (GPIO) pins, of which 26 can be used to control or read

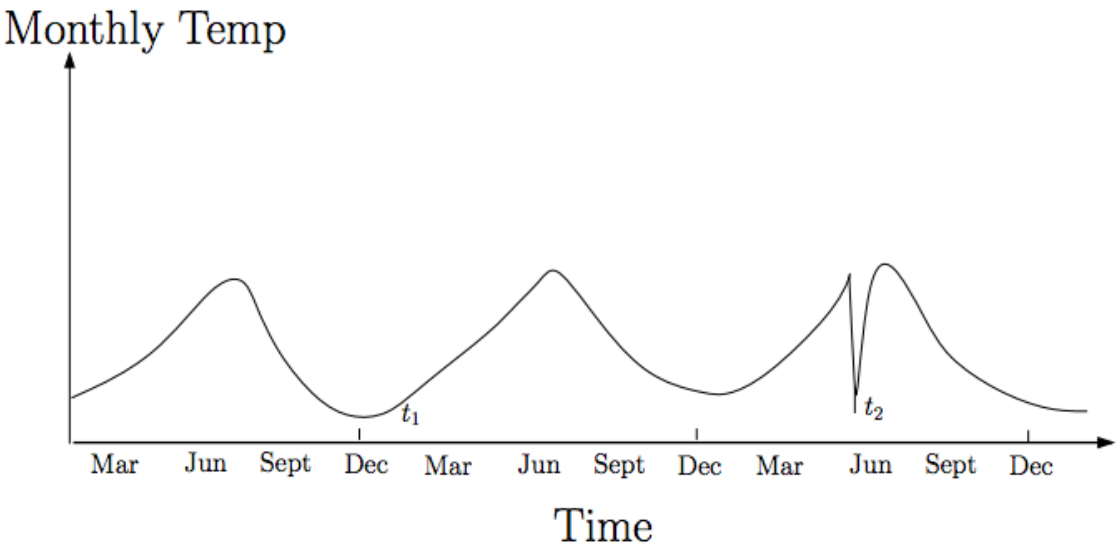

Figure 2: Contextual anomaly t_2 in a temperature time series

external devices e.g. sensors or actuators[9] (req. 4). 1 GPIO pin will be used to read the sensor, another to read the heating element.

3.2 Anomaly Detection Techniques used for IDS

As researched in section 2.3 finding anomalies in data coming from process devices within an ICS can best be done with time series classification or recurrent neural networks regarding the generic requirement. For this poc the decision was made to combine two open source machine learning (req. 3&4) algorithms, in order to achieve also the other three requirements.

For the logic digital signal flowing to the heater point anomalies do not exist since the value is either True or False. By prepossessing the data and calculating how long the heating element has received a True value or False value point anomaly detection becomes feasible. To find these anomalies a Support Vector Machine (SVM) algorithm is used, which finds real-time point anomalies without much computational effort (req. 1).

To find contextual and point anomalies within the discrete digital signal coming from the sensor a Long Short Term Machine Learning (LSTM) algorithm is used. This is computational heavy algorithm because it needs to calculate for every data point whether it fits the context of the rest of the data, but is considered particularly useful for learning sequences containing longer term patterns of unknown length[15] which is the more generic approach but also heavier on computation time and therefore possible slower than the SVM.

Both of the machine learning algorithms will perform novelty detection because it trains their network on a flat file containing non-anomalous data[19]. This file is created by the main script IDS.py. This script requires two arguments. Both specifying for how many minutes the train and test datasets needs to written to file respectively. Once the training data is written, both machine learning algorithms will be started on training their network. Then the test data will be collected which is needed for the LSTM, and concurrently the SVM will start checking each new data value if it is anomalous or not. By using multithreading both of the algorithms can run concurrently.

Support Vector Machine (SVM) The One-Class Support Vector Machine is a module in the open source Scki-kit framework for machine learning[26]. This method is an unsupervised machine learning algorithm and therefore trains on unlabeled data. In the case of the proof of concept, it learns on one class which is the amount of time that the heater is enabled. It then compares the observations (test

data) with the trained data set, and checks if the observation is regular (alike) or not. If not regular, it will be classified as anomaly[19]. One-Class SVM was chosen as it is the lightest method for detecting anomalies in just the test data[19].

Long Short Term Memory Neural Network (LSTM) The LSTM is run on the Raspberrry Pi using the open source Keras libraries with TensorFlow as backend. The LSTM network is trained on the training data containing no anomalies. After training, it can predict a value for every data point based on the previous data points. When the same training model is used but anomalous data is inserted it will predict worse than at data without anomalies. Creating the model and making the predictions is based on the code of Jason Brownlee[2]. Such a wrong prediction is shown in figure 4, where the red line is the prediction on the test data containing an anomaly.

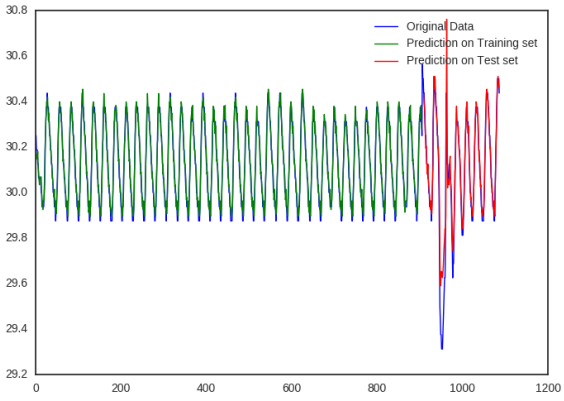

Figure 4: plot of training and test data and their predictions

The *error vector* **EV** for each predicted *data point* **dp** can then be calculated by taking the absolute difference between the real value and the predicted value of that data point. The data point with the highest error vector on the training set will be set as a *threshold* for anomalous data. For each data point in the test set it's error vector will be calculated and compared with this threshold. When bigger, it will be classified as anomaly.

$$EV = |Realvalue - Predictedvalue|$$

$$Threshold = Max(EV_{Train})$$
$$Anomaly = \{dp | EV_{Test}(dp) > Threshold\}$$

To illustrate how the **EV** of every datapoint in the test data relate to the calculated threshold can be seen in figure 5 where the anomaly seen in 4 will also be detected by the LSTM framework.

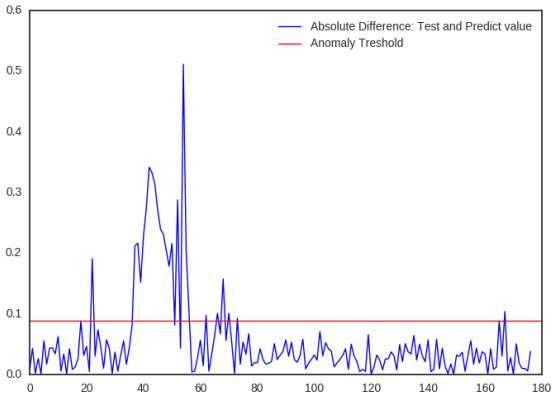

Figure 5: plot of EV on training data with threshold

3.3 Experiments

The two machine learning algorithms on the IDS are put through several tests to detect anomalies. The machine learning algorithms were trained on a training data set which was the collection of all sensor and actuator's data of 50 minutes. After those 50 minutes, they were conducted on 5 different tests, all with a length of 10 minutes.

Test 1 - Baseline To be sure the machine learning algorithms are not overfitted during the training phase.
Test 2 - Faulty Software Here the program of the PLC script was slightly altered in comparison to the PLC script which was running during the collection of the training data. Therefore a possible hack or fault in the software was simulated. This was done by every time the PLC activated the heater, to leave it on 5 seconds longer than usual.
Test 3 - Malfunction In this test a possible small malfunction is simulated in the process devices. First after two minutes, the sensor will be kept out of the water for 10 seconds.

Secondly, the heater will be kept out of the water for 10 seconds during minute 6 when it is activating.
Test 4 - Human Error or vandalism is simulated by adding an ice cube to the water at minute 2.
Test 5 - Damage Since the components of an ICS usually have to last for 20 to 30 years, they might wear. Such a wear might be a leakage, this is simulated by slowly removing 10 % of the water starting from minute 2.
Each of these tests is done 5 times. The results are displayed in table 1.

3.4 Results and Observations

The experiments gave the following results:

Experiment	SVM	LSTM
Test 1 - Baseline	5/5	5/5
Test 2 - Malfunction	5/5	5/5
Test 3 - Faulty Software	3/5	5/5
Test 4 - Human Error	5/5	5/5
Test 5 - Damage	5/5	5/5

Table 1: Experiment Results

A test is seen as passed when it alerted within this 10 min as an anomaly (except for test 1).

Test 1 shows that 5 out of the 5 tests were successful. Meaning that the Proof of Concept did not show any anomalies when nothing is altered. Therefore it can be concluded that the Machine Learning frameworks are not overfitted and too sensitive.

Test 2 shows that 3 out of the 5 tests were successfully for the Support Vector Machine method and 5 out of 5 by the Long Short Term Memory method to detect potential faulty software. A possible reason for this difference is the different input and mechanisms both Machine Learning algorithms base their classification on. SVM classifies based on activation time of the heating element whereas the LSTM bases its classification of history and patterns of the temperature data. During training, the heater was enabled for different time lengths but always resulted in the same pattern. If then the water needed to be warmed for a shorter period but because of the faulty PLC script would be heated 5 sec longer than needed, this might still be in shorter then of another heating period within the training data. Therefore the SVM might not always notice the difference. The LSTM however, would always notice the difference because the result of the heater being enabled to long will result in the temperature raising to high in comparison to the training data.

Test 3 shows that 5 out of the 5 tests were successful. This means that potential malfunctioning heater or sensor can be detected as an anomaly by the Machine Learning frameworks. When the sensor is taken out and later placed back it brings sudden drops to the temperature which will be noticed by LSTM as a contextual anomaly since it differs from the regular pattern from the training data. When the heater is taken out and later placed back during its activation this will be noticed by SVM because it will always be enabled for a longer time because feedback loop to disable the heater was extended with 10 seconds.

Test 4 was successful with 5 out of 5 to detect potential human errors. The water temperature went down when the ice cube was added, this change was detected by the Machine Learning frameworks since the heater needs to be enabled longer than usual (SVM) and the temperature does not conform to its usual pattern anymore (LSTM).

Test 5 was successful with 5 out of 5. This shows that damage to components like a water leak is detected by the Machine Learning frameworks. Removing water is shown as an anomaly as it results in the heater being enabled less long to heat the smaller volume of water to the same temperature (SVM) therefore also resulting in a faster iteration of the patterns compared to the training data (LSTM).

4 Conclusion

During this project, the following research question was answered: *"What are the advantages of anomaly detection between the controlling unit and its process devices?"*.

With use of the literature study it can be concluded that the advantage is that an Intrusion Detection System can inspect the data directly, this in contrary with anomaly detection at a higher level.

Anomaly detection at a higher level would

have the disadvantage that it inspects summarized and abstracted data that is being forwarded from the lower levels, and therefore being dependent on these levels. If the lower levels are intruded or infected, the rightness of the forwarded data can be questioned as it can be altered. This could result in false classification.

With the use of the proof of concept, it can be concluded that Machine Learning can be used to detect anomalies in the data coming from and going to the process devices on a Raspberry Pi. One way to use Machine Learning for this detection is by combining the Support Vector Machine method with the Long Short Term Memory method. The anomalies that were tested and found are malfunctions, vandalism, and intrusions.

The costs of a single IDS made of a Raspberry Pi equipped with open source and free adequate machine learning algorithms would be around hundred 100 dollars. However, ICS owners still need to make the trade-off between the proposed IDS versus the possible prevented costs that could occur when damage is detected at a later stage. When comparing this to the callback Mars had to make of tens of millions of dollars, implementing an IDS made of a Raspberry Pi equipped with adequate machine learning algorithms would not be on the same scale and could pose a valuable addition.

It can also be concluded that further development and research is needed before this solution can be included in a realistic business case. More information regarding shortcomings and discussion points can be found in section 5.

5 Discussion and Future Work

Because of the limited time available for this project, there are some discussion points and shortcomings. These shortcomings are recommended for future work.

5.1 Discussion Points

This project has the following discussion points: Raspberry Pi instead of PLC and the usability.

5.1.1 Raspberry Pi

The PLC was simulated by a Raspberry Pi that is running python code to control the sensor and the actuator. One could debate if this comes close to a realistic scenario, however, due to the limited amount of time and the focus of the project, which is anomaly detection and not PLC's, there was chosen for the fastest and easiest solution. Our simulated PLC setup has the following similarities with an actual PLC: ladder logic with use of Python if, elif, else statements and digital signalling to control sensor and actuator.

The main difference between a Raspberry Pi and a PLC is that PLC is designed for special and uncommon environments and therefore have a longer life expectations and more reliability, where a Raspberry PI is not. However, they both control the sensor and actuator with Boolean logic.

5.1.2 Usability

One could debate if the written software could be used in another environment. In general, it would be relatively easy to use other devices than the ones used in the Proof of Concept. The only requirements are that the actuator is controlled by enabling (True signal) or disabling (False signal), and that the sensor is supported by the W1 subsystem and has discrete data. The W1 subsystem is a framework to read generic digital temperature sensors[28].

5.2 Future Work

This project has the following recommendations for future work: testing and further development.

5.2.1 Testing

Testing was only limited to the test environment with one heater as actuator and one temperature sensor. To implement this Intrusion Detection System in a real life production environment it would be recommended to further test the implementation and do fine tuning towards the production environment.

5.2.2 Further Development

It is recommended to further develop the software by making the Machine Learning methods learn on both the temperature and heater data instead of only 1 of them. Right now the Support Vector Machine method learns only on temperature data, and Long Short Term Memory on the heater data. By learned on both data, deeper patterns could be recognized which could result in better detection.

Another point for further development is making a central management console that can keep track of multiple Intrusion Detection Systems. Right now anomalies are only printed in the terminal locally on the IDS itself. It is recommended to connect the Raspberry Pi and the management on a separate independent network to prevent intrusions if the Industrial Control System environment gets compromised.

References

[1] William Bolton. *Programmable logic controllers*. Newnes, 2015.

[2] Jason Brownlee. *Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras*. 2016. URL: http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[4] Thomas M Chen and Saeed Abu-Nimeh. "Lessons from stuxnet". In: *Computer* 44.4 (2011), pp. 91–93.

[5] Claroty. *Claroty Solution Brief*. Tech. rep. URL: https://s3.amazonaws.com/claroty-public/Claroty_Solution_Brief.pdf.

[6] DarkTrace. *Immune system cyber-security for SCADA systems*. Tech. rep. 2016.

[7] *DS18B20 Temperatuur sensor probe*. URL: https://opencircuit.nl/Product/10369/DS18B20-Temperatuur-sensor-probe.

[8] P Garcia-Teodoro et al. *Anomaly-based network intrusion detection: Techniques, systems and challenges*. 2009. URL: http://www.sciencedirect.com/science/article/pii/S0167404808000692.

[9] *GPIO: Models A+, B+, Raspberry Pi 2 B and Raspberry Pi 3 B*. URL: https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/.

[10] Derek Harp and Bengt Gregory-Brown. *The State of Security in Control Systems Today*. 2015. URL: https://www.sans.org/reading-room/whitepapers/analyst/state-security-control-systems-today-36042.

[11] *Heatercartridge 12V/40W*. URL: https://iprototype.nl/products/3d-printing/ceramic-heatercartridge.

[12] *IBM Reports Significant Increase in ICS Attacks*. 2016. URL: http://www.securityweek.com/ibm-reports-significant-increase-ics-attacks.

[13] Indegy. *Indegy's Core Technologies*. URL: http://www.indegy.com/technology/.

[14] Tony R. Kuphaldt. "Chapter 6 - Ladder Logic". In: IV - Digital ().

[15] Pankaj Malhotra et al. "Long short term memory networks for anomaly detection in time series". In: *Proceedings*. Presses universitaires de Louvain. 2015, p. 89.

[16] "Mars recalls chocolate bars in 55 countries after plastic found in product". In: (2016). URL: https://www.theguardian.com/lifeandstyle/2016/feb/23/mars-chocolate-product-recalls-snickers-milky-way-celebrations-germany-netherlands.

[17] 2016 Wired Business Media. *2016 Industrial Control Systems (ICS) Cyber Security Conference*. 2016. URL: http://www.icscybersecurityconference.com/.

[18] *Network IDS & IPS Deployment Strategies*. URL: https://www.sans.org/reading-room/whitepapers/detection/network-ids-ips-deployment-strategies-2143.

[19] *Novelty and Outlier Detection*. URL: http://scikit-learn.org/stable/modules/outlier_detection.html.

[20] Srinath Perera. *Introduction to Anomaly Detection: Concepts and Techniques*. Nov. 2015. URL: https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/.

[21] *PLC-BLASTER Worm targets Industrial Control Systems*. URL: https://threatpost.com/plc-blaster-worm-targets-industrial-control-systems/119696/.

[22] Check Point. *Check Point Industrial Control Systems Cyber Defense Solution Brief*. URL: https://www.checkpoint.com/downloads/product-related/solution-brief/sb-critical-infrastructure-and-industrial-control-systems.pdf.

[23] *Raspberry Pi 2 Model B*. URL: https://www.raspberrypi.org/products/raspberry-pi-2-model-b.

[24] *Raspbian*. URL: https://www.raspberrypi.org/downloads/raspbian/.

[25] The Bedrock TM Revolution. *Chapter Three: Intrinsic Cyber Security Fundamentals*. Tech. rep. 2016.

[26] *Scikit-learn: Machine Learning in Python*. 2011. URL: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[27] Keith Stouffer, Joe Falco, and Karen Scarfone. "Guide to industrial control systems (ICS) security". In: *NIST special publication* 800.82 (2011), pp. 16–16.

[28] *The 1-wire (w1) subsystem*. URL: https://www.kernel.org/doc/Documentation/w1/w1.generic.

[29] Vance VanDoren. *Open- vs. closed-loop control*. Aug. 2014. URL: http://www.controleng.com/single-article/open-vs-closed-loop-control/f8d8023a15738d0fcfe78d6a2d71dd60.html.

[30] Waterfall. *NERC CIP V5 Standards Position Unidirectional Security Gateways as Secure Alternatives to Firewalls and Network Intrusion Detection Systems*. Tech. rep. 2015. URL: http://waterfall-security.com/wp-content/uploads/2015/12/wf-cipv5-ugw-details-v1.pdf.

[31] Waterfall. *Turbine Monitoring and Diagnostic Threats and Solutions*. Tech. rep. 2015. URL: http://waterfall-security.com/wp-content/uploads/2015/12/turbine-monitoring-threat-d4.pdf.

[32] Kim Zetter. "Inside the Cunning, Unprecedented Hack of Ukraines Power Grid". In: (2016).