Newly Engineered Energy-based Features for Supervised Anomaly Detection in a Physical Model of a Water Supply System

Andres Robles-Durazno, Naghmeh Moradpoor, James McWhinnie, Gordon Russell, and Zhiyuan Tan

Abstract— ***Industrial Control Systems (ICS) are hardware, network, and software, upon which a facility depends to allow daily operations to function. In most cases society takes the operation of such systems, for example public transport, tap water or electricity, for granted. However, the disruption of those systems might have serious consequences across different sectors. In this paper, we propose a supervised energy-based approach for anomaly detection in a clean water supply system using a new dataset which is physically modelled in the Festo MPA workstation rig. The novelty relies on the set of engineered features collected from the testbed, including voltage, current and power from the sensors that compose the ICS. These values are obtained from independent current sensors that we have physically wired to the testbed. Five machine learning algorithms; Support Vector Machine, k-Nearest Neighbours, Multilayer Perceptron, Decision Tree and Random Forest are employed to evaluate the effectiveness of our proposed features. The metrics used to present the performance of the selected machine learning algorithms are F1-Score, G-Mean, False Positive Rate (FPR) and False Negative Rate (FNR). The results show that machine learning algorithms can classify the variations of energy produced by the execution of cyber-attacks as anomalous by achieving 95.5% F1-Score, and 6.8% FNR with the Multilayer Perceptron classifier.***

Index Terms— Industrial Control Systems, SCADA, Supervised Machine Learning, Anomaly Detection, Energy Monitoring, Novel Dataset

I. Introduction

ICS are broadly used in critical infrastructure and large-scale industrial processes such as transportation, power, water, waste-water treatment, oil, gas, and communication systems, air and sea industries, hospitals, health clinics, fire, police, finance, and public administration services as well as chemical, and pharmaceutical industries [1]. Given that nations are highly dependent on their continuity and operations, any disruptions of these systems could lead to a significant economic loss and can have a substantial impact on public lives, health and safety. The non-disruptive nature of an ICS with the requirement for 24/7 availability and the fact that access to such a system is extremely difficult, hazardous, and sometimes impossible, validate the use of physical, virtual, and hybrid ICS testbeds rather than real systems by cybersecurity researchers.

Historically, the hardware involved in the operation of ICS (such as Programmable Logic Controllers (PLC)s) ran on proprietary hardware and software in physically secure and isolated locations, although, more recently they have implemented Information Technology (IT) capabilities such as network capabilities [2]. Thus, ICS have inherited cyber-vulnerabilities related to IT networks, such as Denial of Service (DoS), Man in the Middle (MITM) and Spoofing [3]. It should be noted that cyber-attacks that target critical industries such as oil, pharmaceutical, nuclear and water might have devastating impacts given that they may put thousands of human lives in danger. A considerable number of cyber-attacks against ICS have been reported in recent years [4]. For instance, Stuxnet [5] is a sophisticated malware that modified the operation of a PLC resulting in the explosion of various centrifuges belonging to an Iranian enriching facility. Another example is BlackEnergy [6], a malware that infected the ICS components of a power facility in Ukraine. The execution of this malicious software resulted in a power outage for a few hours during the coldest month in the region. Detecting cyber-attacks is an ongoing battle given that new attack vectors and malware appear every day. One of the approaches to address this issue is the implementation of traditional security devices such as firewalls and Intrusion Detection Systems (IDS) aimed at detecting anomaly activity [7]–[9]. However, IT and ICS networks have notable differences that have to be considered. The aim for security specialists in an IT network is to protect the information, while, in an ICS network the main objective is to protect the physical process under operation. Therefore, traditional security devices such as firewalls might not be a feasible strategy for ICS networks. For this reason, researchers have explored various alternatives to detect anomalies in ICS, the most popular being the adoption of supervised and unsupervised machine learning techniques. [7]–[9]. Current approaches employ a given machine learning algorithm with the information collected from network packets mostly obtained from virtual implementations rather than physical testbeds. One of the reasons why most of the research is based on virtual scenarios is the cost of implementing and maintaining physical testbeds. For that reason, it may be argued whether the machine learning models that had been trained with information obtained from virtual testbeds are applicable to real implementations. Another point to take into consideration is how trustworthy is the information obtained from the control network, as it has been demonstrated that hackers are capable of modifying network packets aiming to tamper with systems. For instance, in the Stuxnet attack explained above, the monitoring application of the enriching

process did not show any sign of malfunction until the centrifuges were damaged and stopped working.

In this paper, our goal is to address cybersecurity in ICS by proposing an energy-based mechanism for anomalous detection in a clean water supply system. To demonstrate this concept, we implement a Supervisory Control and Data Acquisition (SCADA) testbed using the Festo MPA process control rig. This testbed allows to monitor the energy consumption of the sensors and actuators using the INA219 current sensor. A raspberry pi collects and stores this information in a text file, which, is later used to create the machine learning models used for attack detection. The novelty of the approach proposed in this paper relies on the use of features that are not obtained from a virtual testbed [7-8], nor network traffic [9], instead, it is built in newly engineered energy-based features obtained from the INA219 current sensor which is hard-wired to the actuators/sensors that compose the Festo rig. It should also be noted that in our scenario an intruder might not be able to tamper with the energy-based features that are used to create the machine learning models because they are not accessible from the control network as will be discussed in later sections.

A. *Research Objectives and Hypothesis*

The objective of this paper is to demonstrate the feasibility of detecting cyber-attacks against ICS with a particular focus on a clean water supply system and by using an energy-based machine learning approach. In addition, our aim is to demonstrate the importance of the feature selection process on the performance of the machine learning algorithms. To achieve these objectives, we outline the following hypothesis.

Hypothesis. Newly engineered energy-based features collected from monitoring the energy consumption of the sensors and actuators that compose a model of a clean water supply system in conjunction with well-known supervised machine learning algorithms allow the detection of anomalies that may have a negative impact on the control system.

B. *Contribution*

In this paper, we proposed a set of energy-based features for machine learning classifications that were not obtained from the network traffic nor from a data logger. Given that the information related to an ICS obtained from a network traffic or from a data logger might already be compromised by an intruder. Therefore, our proposed features are immune from tampering as they are captured from the INA219 current sensors. In addition, having a physical implementation allows to face scenarios where noise is present in the data. These types of scenarios cannot be replicated, and they are not present in virtual implementations. Finally, we apply a set of well-known machine learning algorithms, which have been used in related research, to demonstrate the feasibility of our proposed energy-based features in our novel dataset.

C. *Organization of the paper*

The paper is organized as follows. In section II we review the related work in the field. In section III we describe our research approach followed by the experimental design and setup in section IV. In section V we discuss the findings and results. In section VI we discuss the hypothesis stated in this paper. Finally, in section VII we present the conclusions followed by future work and acknowledgements. References are listed at the end of the paper.

II. RELATED WORK

A detailed review of current related work is discussed in two main categories of supervised and unsupervised machine learning techniques for anomaly detection in ICS. The relevance of the quality of the features chosen for training the machine learning algorithm is highlighted to compare with our proposed energy-based features. The related work includes the relevant papers from reliable resources such as IEEEXplore, Elsevier, ACM and Google Scholar.

In [10], the authors proposed an attack detection model for a power system based on supervised machine learning. The features used to build the model are constructed by analysing the relationship between the features and raw data that is obtained from relevant log information and historical data. The original dataset used in their research contains 128 features collected from four Power Management Units (PMU), snort alarms and logs. Their data pre-processing phase involves discarding redundant features that might overfit the model. Afterwards, the dataset is divided into four subsets of data and part of the original features are sent to AdaBoost model for training along with the new features. During the experimentation phase the authors compare their approach with several traditional machine learning algorithms to demonstrate the effectiveness of their model. The metrics for evaluating the model include accuracy, precision, recall, F1 score, ROC curve and AUC. Addressing their results, their proposed model shows the benefits of feature engineering. Although their approach presents good results, it can be argued that historical data and logs might not be a reliable source of features given that they are susceptible to manipulation.

In [11], the authors propose a Multi-Layer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data. The proposed IDS combine signature-based and anomaly-based analysis of host, network and process data. Their mechanism of detection is placed as a second line of defence behind the firewall, and it is composed of data-driven models for cyber-attack detection based on network traffic and system data. The classification models are based on supervised machine learning algorithms such as KNN, Random Forest, Decision Tree and Bagging. Those algorithms can detect well-known attacks only, for that reason, the authors also include an unsupervised approach using the AAKR, which provides flexibility for intrusion detection. The dataset contains 142 features that are related to memory, a computer process, and network behaviour and it includes three cyber-attacks: MITM, DoS attack to the engineering

workstation, and DoS attack to the National Instruments cDAQ (the data acquisition and control hardware). Addressing their captured results, the KNN algorithm outperforms the rest of the algorithms by achieving a score of 98.84% for true positive alarms and 99.46% for true negative. The rate of False Negative alarms achieves 1.16% being the lowest among the rest of the algorithms. Decision tree algorithm has the lowest computing cost; however, the four algorithms remain below one second.

In [12], the authors propose an Industrial Control System Network Intrusion Detection by Telemetry Analysis. They used the honeypot Conpot to simulate the network traffic generated by two Siemens SIMATIC S7-200 PLCs. They then use the python library (pymodbus) to generate the MODBUS protocol stack. Their IDS is implemented as a standalone device that monitors the traffic between the PLC and the rest of the network. They employed REPTtree as a base machine learning algorithm and a set of bagging-aided classifiers for training. They generated a list of features after analysing 838,818 packets, including malicious and benign traces, generated by their virtual ICS. The traffic generated among the devices connected to the ICS network is identified as insider whereas the traffic between the control network and an external network is identified as an outsider. Addressing their results, they achieved a 92.2% accuracy rate for REPTree classifier at the insider classification. For the outsider classification, most of the classifiers achieved high accuracy when classifying packets from different machines, C4.5 and REPTree achieved 99.5% and 99.6% of accuracy respectively. Given that their work is based on a simulation, it is not clear whether the proposed research is applicable to real scenarios for example a power plant or a water treatment system.

In [13], the authors introduce a Machine Learning-based Defence Against Process-Aware Attacks on Industrial Control Systems. They developed a supervised SVM model that can differentiate between disturbances during normal operation and malicious activity. They employ the Tennessee Eastman (TE) Chemical Process as a testbed to assess their approach. They build upon Matlab Simulink model of the TE process, and they incorporate a serial hardware interface between the simulation model and a PLC. The testbed includes 50 states, 41 measured variables with Gaussian noises, 12 manipulated variables and 13 disturbance signals. Their dataset includes information obtained from 12 sensors under normal operation, under attack and various disturbance conditions under normal operation. For the training process, they selected the RBF kernel from the SVM algorithm with parameter N=1 and N=50 for attack detection and to identify the type of attack executed. They run a simulation for the lapse of 2 hours, where a set of attacks were executed during that lapse. Addressing their results, the proposed mechanism of defence model is able to differentiate between a system disturbance and an attack. However, it is not clear whether in their virtual environment they have considered conditions such as: environment, noise and network latency which are present in a real ICS.

In [14], the authors propose a high-performance unsupervised anomaly detection for cyber-physical systems. They used the secure water treatment (SWaT) S3 dataset that contains network traffic with a rate of approximately 11M packets per hour, a public dataset from a power grid control system that consists of 11 network traces. They replace the usual step of feature extraction, usually used in machine learning, by a feature learning approach that is based on current deep learning schemes. They employed a neural network composed of three layers of input, output and hidden. Their proposed framework is implemented in python using the TensorFlow framework for processing and pcap library for packet acquisition. Addressing their results, they achieved 100% of precision and F1 score in the power grid dataset and 0% of false-positive detection. Evaluating the second dataset, it achieves 99% of precision and recall.

Using a similar SWaT testbed, the authors in [15] proposed a mechanism for cyber-attack detection on industrial control systems using convolutional neural networks. They employed a selection of deep neural networks architecture including different variants of convolutional and recurrent networks. They implemented the unsupervised machine learning models using Google's TensorFlow framework. Their dataset is normalized to 0-1 scale and includes 496,800 records in normal operation and 449,919 records under 36 different attacks. Addressing their results, the anomaly detection algorithm achieves the highest AUC by reaching 96.7% for eight layers of CNN. Regarding the training and testing time, the CNN was shorter by a factor 1 to 2 for testing and 1,5 to 4 for training when it is compared to a pure LTSM network. Their mechanism of detection failed in recognizing four types of attacks, however those attacks did not have a considerable impact on the system. The f1 score of the ensemble of four layers 1D CNN model achieved 92.06% with a precision of 1 and recall of 85.29%.

In [16] the authors proposed a Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–based Anomaly Detection Techniques. Their proposed approach involves a four-layer method, where the first layer checks whether the given SCADA observations follow the actuator rules specified for the system, while the second layer finds statistical outliers. The third layer is a neural network that can detect contextual inconsistencies with normal operation and the four-layer uses Principal Component Analysis (PCA) on the entire set of sensors that compose the ICS to classify the samples as normal or abnormal. They used three independent datasets that were obtained from the C-Town WDS, which is a medium-size water distribution network. The dataset contains seven different attacks that were simulated in MATLAB. The performance of their proposed approach is evaluated by adopting the metrics specified in BATDAL. Addressing their captured results, their algorithm can detect the entire set of simulated attacks and only one false alarm was triggered. For the validation dataset, the CSM score achieved 95.3%, while its true negative rate (TNR) reached 94.6%. The overall score of the algorithm is 96.8%, which indicated a satisfactory performance.

In [17], the author proposed an efficient data-driven clustering technique to detect attacks in SCADA systems. Their approach is based on the assumption that normal states can be clustered into finite groups of dense clusters. In addition, critical states in

the n-dimensional space will take the form of noise data. They describe the requirements for developing a SCADA-based IDS: a model able to identify normal/critical states and a proximity-based extraction technique to derive rules. They employed the clustering algorithm: DBSCAN for identifying normal and critical states. To validate their approach, the authors

implemented a virtual ICS that involves five virtual machines, four of them are used as PLC's and they run the MODBUS/TCP-Salve simulator. The fifth virtual machine is used as Master Unit Terminal (MUT), historian server and Human Machine Interface (HMI) client. They used three datasets obtained from their virtual implementation, as well as five datasets publicly available. Addressing their results, the proposed approach achieved an average accuracy of 98% and 0.02% in the detection rate and false positive. The authors proposed the re-labelling technique aiming to reduce the number of false/positive alarms. Addressing the captured results, the number of alarms is reduced by 16%. In this paper, we implemented and fully discuss a Supervised energy-based approach for anomaly detection on a real PLC of a Festo MPS PA Compact Workstation Rig, which is a working model of a clean water supply system, which differs from virtual implementations described in existing work e.g. [12], [16] and [17].

Our proposed technique differs from [10], [11] and [12] because the machine learning algorithms are feed with features that are collected from the INA219 sensor, which is hard-wired to the sensors and actuators. We do not rely on packets obtained from the control network dislike work in [11], [12] and [14] because it might have been compromised by intruders before reaching the machine learning process. Furthermore, the work presented in this paper is different from the existing work, described in Section II, given that our datasets contain malicious and benign traces obtained from a physical testbed in which, unlike [16], a set of real attacks were executed to the testbed when the datasets were collected. Table I provides a summary of the testbeds explained in Section II.

TABLE I TESTBED SUMMARY

Testbed	Type	Components	Attack Vector	ML Approach	ML Algorithms	Reference
CWSS: Clean Water Supply System	Physical	PLC, SCADA, HMI	Packet Crafting, PLC memory corruption	Supervised Machine Learning		This research
Power System	Virtual	Power Management Units, Snort and logs.	Man-In-The-Middle, ARP Spoofing	Supervised Machine Learning	KNN, SVM, GBDT, XGBoost, CNN and Random Forest	[10]
ICS Network traffic	Virtual	Computer process, memory, and network behavior	DoS, Man-In-The-Middle	Supervised Machine Learning	KNN, Random Forest, Decision Tree and Bagging	[11]
ICS Network traffic	Physical	2 SIMATIC S7-1200	Packet delay variation, variable packet loss	Supervised Machine Learning	C4, REPTree	[12]
HITL Testbed: Tennessee Eastman (TE) chemical process	Hybrid	PLC, Virtual process	ARP Spoofing, Man-In-The-Middle	Supervised Machine Learning	SVM	[13]
SWaT	Physical	PLC, MTU, HMI, SCADA	DoS Attack, ARP Spoofing	Unsupervised Machine Learning	Neural Networks	[14]
SWaT	Physical	PLC, MTU, HMI, SCADA	DoS Attack, ARP Spoofing	Unsupervised Machine Learning	TensorFlow framework	[15]
Water Distribution System	Physical	MATLAB	ARP Spoofing, False Data Injection, DoS	Unsupervised Machine Learning	Neural Networks	[16]
Water Distribution System	Virtual	PLC, MUT, HMI	DoS Attack, ARP Spoofing, MITM	Unsupervised Machine Learning	Clustering Algorithms	[17]

III. Research Approach

In this section the research objectives, hypothesis and methodology are described.

A. The experimental Setup

To evaluate the machine learning algorithms proposed in this paper, the computer simulations were performed using the method: stratified 5-fold cross-validation with a suitable data split for training and testing. This method is widely used because the results are less biased and more realistic than other methods such as a simple train/test split. We adopted the following phases to clarify and answer the above hypothesis.

1. Pre-processing Phase.
    a. Smoothing the voltage signal collected from the ultrasonic sensor by applying a digital filter.
    b. Applying three different feature selection techniques for discarding redundant or low informative features.
    c. Balancing the dataset by applying oversampling techniques such as SMOTE.
    d. Splitting the data into training and testing datasets by using 5-fold cross-validation.
    e. Normalizing or Standardizing the dataset depending on the selected Machine Learning algorithm.

2. Training & Testing Phase.

    a. Training the selected machine learning algorithm with the training dataset.
    b. Obtaining the prediction results using the testing dataset.
    c. Performance evaluation of the selected machine learning algorithms

B. ICS Datasets

SWaT [18] and WADI [19] are the most common physical testbeds employed for the cybersecurity analysis of water treatment/clean water supply systems. Most of the research in the field are based on these two physical systems, either by having a direct access to them or by having access to the associated datasets generated under malicious and benign scenarios. The two testbeds are also the closest existing work to our research in this paper. Given that we have also generated our very own dataset, named Clean Water Supply System (CWSS), to advance the research in the field, the review comparison of the three datasets (SWaT, WADI and CWSS) is as follows.

The SWaT testbed was developed by the iTrust Center for Research in Cyber Security at the Singapore University of Technology and Design (SUDU) [20]. SWaT represents a scaled-down version of a water treatment plant that produces 5 gallons of water per minute. The SWaT dataset is composed of the network traffic of 51 sensors and actuators during seven days of normal operation. The normal operation corresponds to the starting and stabilization of the plant. A total of 41 attacks were executed during four days of operation.

The WADI is a testbed that simulates a scaled-down water distribution system. It was developed and implemented by the same creators of SWaT. The WADI testbed includes a large number of tanks that supply water to customer tanks. The dataset contains values obtained from 123 sensors and actuators during fourteen days of normal operation over which a total of 15 attack scenarios were executed.

Our CWSS testbed simulates a model of a clean water supply system in the Festo MPA Compact Workstation rig. The CWSS testbed includes 7 sensors and actuators that operate for one day. Further, 7 attacks were executed against the testbed during 11 hours of operation. Our dataset contains energy features obtained from the INA219 current sensor and hard-wired between the PLC and sensors/actuators composing the physical system.

In terms of network protocol, CWSS testbed implements Profinet [21], which is an industrial standard for data communication over TCP/IP, while SWaT employs Modbus TCP [22] and WADI devices CIP over Ethernet/IP. Modbus TCP is a protocol with vulnerabilities [23] e.g. it lacks adequate security checks in communication between two endpoints which could allow an unauthenticated remote attacker to send random commands against any slave device using the MODBUS master. However, Profinet protocol provides more secure communication and is the most widely used standard in ICS. Therefore, from an attacker's point of view, it is more difficult to issue cyber-attacks against a system which implements Profinet (i.e., CWSS) rather than Modbus TCP (i.e., SWaT).

Furthermore, SWaT and WADI datasets are based on basic and traditional network-based attacks such as ARP spoofing and Man-In-The-Middle attacks for which we already have many protections [24]. For example, static ARP entries, encryption, VPN, packet filters, HTTPS, public key pair authentication and many Instruction Detection Systems (IDS) can easily stop these attacks. However, in CWSS testbed, we implemented a novel set of attacks against the input/output and working memory of Siemens S7-1500 [25]. Siemens S7-1500 is one of the popular PLCs available on the market and used in industry at the moment. The features in the CWSS dataset are energy-based collected from the INA219 current sensor hard-wired between the PLC and sensors/actuators on a model of a clean water supply systems. Additionally, WADI does not provide details regarding network implementation over which malicious and benign scenarios have been issued and dataset has been generated while in CWSS we fully explained this specification. The implementation of cyber-attacks against both WADI and SWaT is also unclear while this is fully detailed in CWSS. These makes the CWSS dataset more understandable and more realistic in terms of collected features and events in comparison with SWaT and WADI datasets.

In general, although the SWaT and WADI are bigger datasets captured over longer periods in comparison with CWSS, CWSS

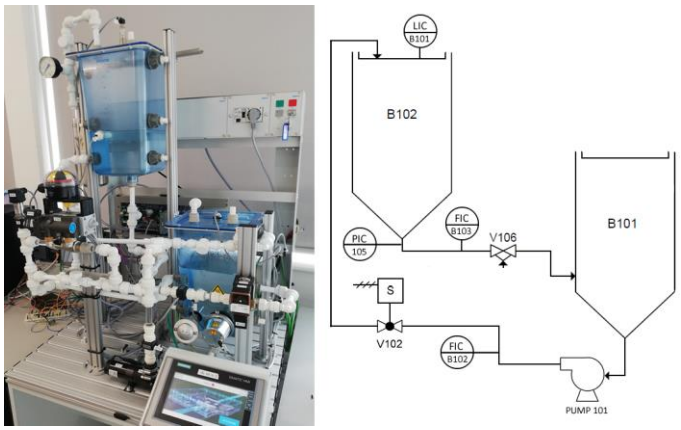

Figure 1. Testbed

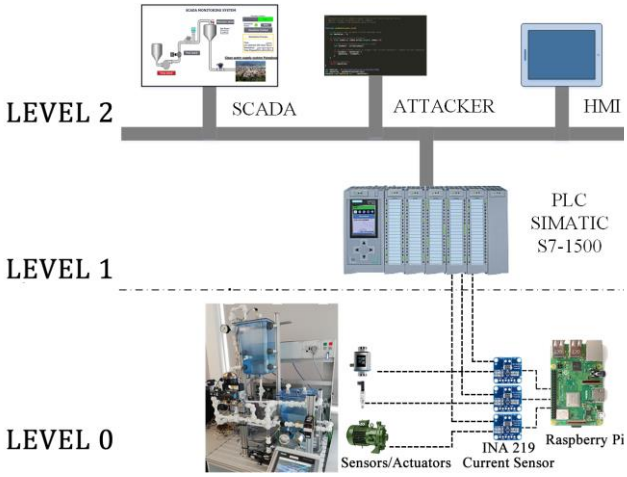

Figure 2. Dataset Collection.

dataset has been collected under novel attacks against the input/output and working memory of a PLC currently used in industry, having more severe consequences on ICS, is more realistic in terms of attack novelty, consisting more difficult attacks from an attacker's point of view, and does not need an attacker to have a full knowledge of the system.

C. CWSS Dataset Description

This section provides an overview of the CWSS datasets used for the experiments implemented in this paper. A testbed simulating an uninterrupted clean water supply system was modelled using the Festo MPS PA Compact Workstation Rig [26]. The control process implementation is depicted in Figure 1 where the tank B101 contains the water that supplies the reservoir tank B102 through the variable speed PUMP 101. The water demand from customers was modelled and implemented using the proportional valve V106 of the Festo Rig. In normal operation, the water level in the reservoir tank B102 is maintained at a setpoint defined by an operator. The full description of the control implementation can be found in our previous work [27]. For the attack scenario, we implemented a set of attacks to the memory of the PLC aiming at overwriting the input memory of the PLC; hence the normal operation of the control system is affected. For instance, the attacker might modify the input memory of the ultrasonic sensor pretending that the current water level is lower than it is. Consequently, the control system will increase the speed of the pump, resulting in an increase of the water level above the setpoint for the tank B102. This might result in a tank overflow. The set of the executed attacks to the ICS are listed in Table III. The full implementation of the attacks and threat model can be found in our previous work in [27] and our own implemented source code for this can be found in [28]. The dataset for this paper that contains the energy traces of the sensors involved in the testbed are shown in Table II. To achieve this, we extended our previous work [29] by wiring the current sensor INA 219 [30] to each one of the sensors and then collecting the data using a Raspberry PI 3 [31]. Figure 2 shows the architecture of the testbed employed in this research according to the ICS reference model suggested by NIST Special Publication 800-82 [32] which defines four levels. Level 0: Input/Output refers to the physical process. It includes hardware, such as sensors and actuators that are directly connected to the control process

Level 1: Control Network involves the equipment used to monitor and control the physical process. At this level, the information from the sensors is obtained and processed to then generate the outputs that will be sent to the actuators. Equipment at level 1 comprises PLCs and RTUs. Level 2: Supervisory Control, includes the systems used to monitor and control the physical process. This level includes devices such as HMI, workstations, and servers. Lastly, Level 3: Corporate Network denotes the equipment involved in the business-related activities.

The dataset was collected when the ICS was in operation for over 8 hours. Figure 3 shows the number of collected malicious and benign instances during the operation. The number of instances that corresponds to the malicious class is 35.72% of the entire data set, while 64.28% corresponds to the benign class, therefore we can say that the dataset is unbalanced. One of the major problems of using machine learning on imbalanced datasets is obtaining a biased and inaccurate model [33]. To overcome the imbalance problem, we use Synthetic Minority Over-Sampling Technique (SMOTE) on the original dataset, a method that uses the k-nearest neighbour to produce new synthetic instances of the minority class [34]. As it can be seen in Figure 3, Dataset I has an equal number of Malicious and Benign instances, after employing SMOTE on our original dataset,. Furthermore, SMOTE is used to create two more datasets labelled as Dataset II and Dataset III which are also depicted in Figure 3. These datasets will aid us to evaluate the performance of the machine learning algorithms in the following sections.

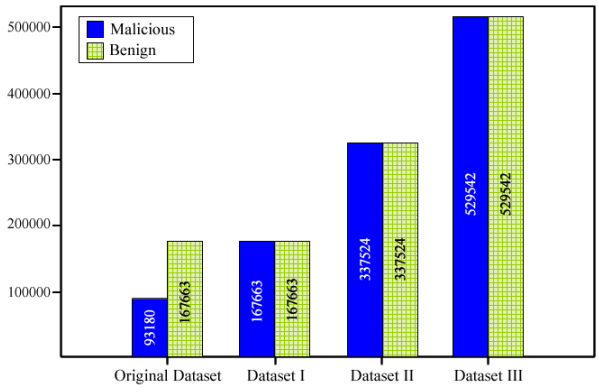

Figure 3. Dataset Details.

TABLE II FEATURES

Feature	Sensor	Feature	Sensor
1. sh_ultra	Ultrasonic Sensor	13. sh_fo	Flowmeter Out
2. v_ultra		14. v_fo	
3. c_ultra		15. c_fo	
4. p_ultra		16. p_fo	
5. sh_pump	Pump	17. sh_pi	Pressure In
6. v_pump		18. v_pi	
7. c_pump		19. c_pi	
8. p_pump		20. p_pi	
9. sh_fi	Flowmeter In	21. sh_po	Pressure Out
10. v_fi		22. v_po	
11. c_fi		23. c_po	
12. p_fi		24. p_po	

D. Pre-processing

Data pre-processing is a data mining technique which is used to improve the quality of the raw data [35]. This stage has a significant impact on the performance of supervised learning models because unreliable input could lead to obtaining incorrect results. For instance, in our scenario, the data collected from the ultrasonic sensor contains undesirable noise that might be misclassified in cases where it is not removed. In the following section, we describe the pre-processing stage which includes the de-noising phase and the feature selection process followed by an overview of the selected machine learning algorithms employed in this paper.

E. Dataset Filtering Process

In our scenario, the capacity of the water tank is 10 litres, and the water is poured from the top of the tank. When the control system starts and the tank is empty, the water bounces at the bottom of the tank. This process generates noise in the readings obtained from the ultrasonic sensor. The noise decreases as soon as the water in the reservoir tank starts to increase. However, it should be noted that the noise is always present in the signal obtained from the ultrasonic sensor, in smaller or bigger quantities Therefore, in this section, we explain the noise removal of the ultrasonic sensor involved in our implemented testbed. Further, external factors such as noise and temperature could lead a sensor to fail to recognize the correct water level which adds noise to the dataset. The analogue sensor, which is labelled as B101 in Figure 1, is fitted on the top of the reservoir tank. It uses sound waves above 20000 Hz, which is beyond human hearing, to measure the distance between the sensor and the water. The analogue signal is converted by means of a transducer into a standard (0-10v) electrical signal

TABLE III SET OF ATTACKS EXECUTED TO THE CONTROL SYSTEM

Attack	Effect
Changing Setpoint in the Working Memory	Water Level Increases/Decreases 2-2.5 litres. It depends on the value sent from attacker to the Input Memory of the PLC.
Attack on Ultrasonic Sensor	Water Level Increases/Decreases. It depends on the value sent from attacker to the Input Memory of the PLC.
Attack on Flow In	Affects Pump Operation, consequently the water level in the reservoir tank.
Attack on Pump	Water level decreases 0.5-1 Litres.
Attack on Flow Out	Affects the Control Operation when using feedforward Controller.
Attack on Pressure In	Slightly affects the normal operation of the control system. The water level increases/decreases 0.1 - 0.2 litres.
Attack on Pressure Out	Affects the control operation when using a PI controller that takes the Pressure Out as Input for calculating the water level, otherwise this does not affect the control operation.

The machine learning algorithms may miss out patterns and provide wrong results when noise is present in a given dataset. One common technique in signal pre-processing is the design and the use of filters to remove unwanted frequencies from electrical signals. There are a considerable number of filters for signal processing such as Low Pass Filter (LPF), High Pass Filter (HPF) and Band Pass Filter (BPF). Although these filters are electrical circuits composed by resistors, amplifiers, and capacitors, they can be digitally implemented by mathematical equations. In this paper, we apply a LPF on the data collected from the ultrasonic sensor as its success has been proven in similar research such as [36], [37]. Low-pass filters allow the low-frequency components of an input signal to pass through while reducing the high-frequency components. Measurement noise falls into the high-frequency range of the signal spectrum, while the underlying process signal is generally toward the low-frequency end [55].

The blue line in Figure 4 shows the signal obtained from the ultrasonic sensor without filtering. The signal contains a considerable amount of noise that might affect the performance of our selected machine learning algorithms. In Figure 4, the yellow line shows the Ultrasonic sensor signal that we filtered with a normalised passband frequency of $0.001\pi\,{}^{r}/_{s}$ and a stopband attenuation of 60dB.

As it is shown, this signal contains less noise than the original one, however, there are still remanences of noise. Please refer to [29] for comprehensive explanations on normalised passband frequency and stopband attenuation. To remove as much noise as possible, we apply Butterworth LPF [36][38] which is a digital filter that has a flat response in the passband [38] and sucessfully applied in similar researches [27][45]. Butterworth LPF smooths the electrical signals with a frequency higher than the cutoff frequency. The cutoff frequency is the boundary between the desired and undesired frequencies. It should be noted that the cut-off frequency does not define good or bad frequencies. The orange line in Figure 4 shows the ultrasonic

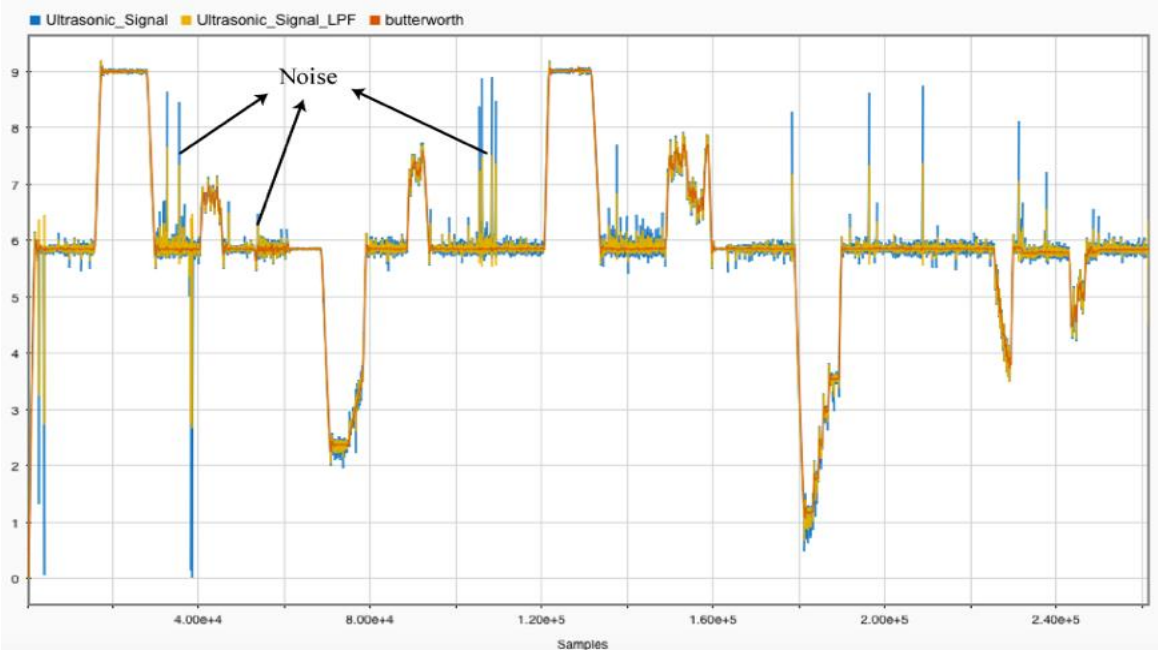

Figure 4. Raw Ultrasonic Sensor Signal

sensor signal when the Butterworth LPF is applied which shows Butterworth filter removes more noise than the simple LPF.

F. Feature Selection

In machine learning techniques, feature selection is a process of choosing the most relevant features that are useful in predicting the desired response [39]. In this paper, we obtained twenty-four features from six sensors located on our testbed, as it is shown in Table II. The main aim of using feature selection techniques is to reduce the number of features to the most relevant ones for later use in building models based on machine learning algorithms. It should be noted that feature selection and feature extraction are two different concepts. Both techniques have the same aim of reducing the dimensionality of the dataset, however, the main difference is that feature selection keeps a subset of original features, while feature extraction creates new sets of features from the available ones [40].The benefit of using feature selection before training a machine learning algorithm is the reduction of the dataset dimensionality, as a result, the time taken to build a machine

learning model will be reduced. It has been widely studied before by the following authors[41], [42]. Further, another benefit worth pointing out is that feature selection will improve the machine learning metrics such as accuracy and precision [35]. There is a considerable number of feature selection techniques to perform feature selection such as lasso regression, step wise forward and backward selection [43] [44] [9] [45], however, in this paper, we chose the most suitable ones for our dataset and based on their popularity in the similar researches. These techniques are Information Gain, Chi-Square and Correlation Based.

Information Gain (IG) measures the amount of information that a feature gives about a class [45]. It measures the reduction in entropy, which can be defined as the information and the degree of uncertainty of random variables. IG tells how important an attribute is and it will be used for discriminating between the classes to be learned [39]. Chi-Square is another popular method of feature selection technique. It applies the statistical X^2 to measure the independence of two events. In feature selection, these two events are an occurrence of the feature and occurrence of the class[46]. The value of X^2 is high when the two events are dependent. It means that the feature is correlated with the class and it should not be discarded. The higher the value of X^2, the more relation that the feature has with the class [39]. Correlation-Based is a feature selection technique for classification tasks in Machine Learning. It examines each feature individually in order to determine the relationship of the feature with the corresponding class [46]. Each feature is ranked according to the achieved correlation score.

G. Selected Features

Each feature selection method measures the relevance of the features depending on its correlation with the dependent variable. Figure 5 shows the features that obtained the highest scores in each one of the feature selection techniques described above. This representation shows the features that each algorithm has in common. For instance, the features 14. v_fo, 4. p_ultras, 3. c_ultras, 22. v_po, 8. p_pump, 2. v_ultras, 10. v_fi, 5. sh_pump are among those that obtained higher scores in the three feature selection techniques. Furthermore, the features that IG and Correlation Base have in common are 18. v_pi, 6. v_pump and 1. sh_ultrasonic. The Chi-Square is the only one among the other two feature selection techniques that chose the feature 12. p_fi. A condition for evaluating the relation of features with the dependent variable is analysing density curves for the malicious and benign traces [47].

Figure 6 and Figure 7 shows the density of malicious and benign events in the features: voltage in the ultrasonic sensor (2. v_ultras) and power in the pump (8. p_pump). These features are ranked with high scores according to our three feature selection techniques (IG, X2, and Correlation-Based). Both features are suitable for feature classification because the peak of the curve for malicious and benign traffic are opposite of each other. Fig 8 and Fig 9 show features with a low score such as Voltage in the shunt resistor that monitors the Pressure Out and Pressures In sensor (21. sh_po, 17. sh_pi). The malicious and benign distributions are completely overlapped; hence, these features are not suitable to be considered for classification.

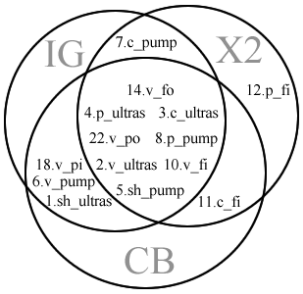

Figure 5. Features Selected.

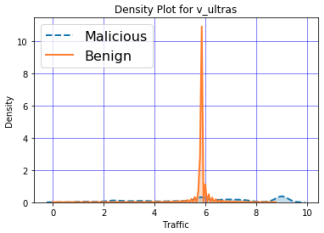

Figure 6. Density plot for Voltage in the Ultrasonic Sensor.

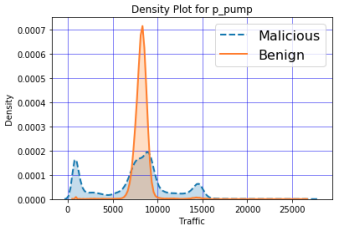

Figure 7. Density plot for Power in the Pump.

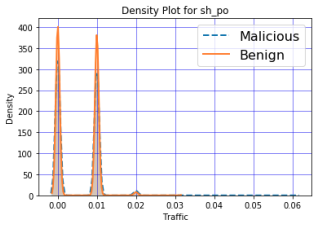

Figure 8. Density plot for Voltage in Shunt in the Pressure Out Sensor.

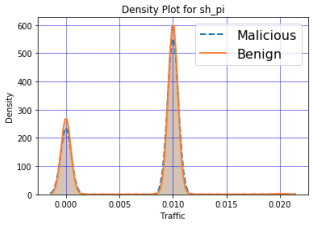

Figure 9. Density plot for Voltage in the Shunt in the Pressure In Sensor.

H. *Overview Machine Learning Algorithms*

For the experimental process conducted in this paper, we employ five machine learning algorithms that are widely used. They were addressed and analysed in the related work. These algorithms are Decision Tree, Gaussian Naïve Bayes, Multilayer Perceptron, KNN and SVM. They are briefly explained as follows.

Decision Tree

Decision Tree (DT) is a type of supervised Machine Learning algorithm in which the entire dataset is divided into smaller datasets by taking into account the descriptive features until the set is small enough to contain the points that fall under one label [48]. Instances from the dataset are sorted down the tree from the root to a designated leaf node. DT provides a classification for each instance in a given dataset [49].

Gaussian Naïve Bayes

Naïve Bayes (NB) is one of the most used classifiers given that it provides a simple approach with clear semantics to represent. It uses learning probabilistic knowledge and is an algorithm for binary and multi-class classification problems. NB can be extended to real-value attributes by assuming a Gaussian Distribution [50] which is then called Gaussian Naïve Bayes (GNB). GNB takes each data point and assigns it to the nearest class. The assumption of this classifier is that the data from each labelled class (e.g., Malicious or Benign) is represented by a simple Gaussian Distribution.

Multilayer Perceptron

Multilayer Perceptron (MLP) is a type of Artificial Neural Network, that is inspired by how the biological neural networks are capable of processing information in the human brain [51]. Figure 10 shows the representation of an MLP that is composed of one hidden layer, one input layer, and output layer, the input layer (i) feeds the layer (j), which is a multidimensional perceptron with a weight matrix W.

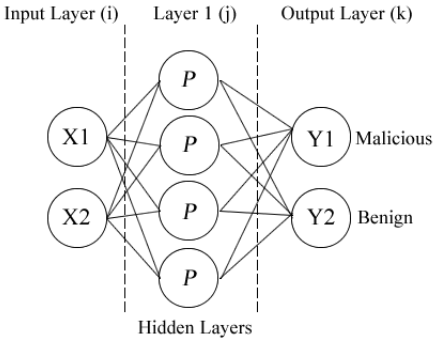

Figure 10. Multilayer Perceptron.

K-Nearest Neighbour

K-Nearest Neighbour (KNN) is one of the most straight forward algorithms which is frequently used for classification tasks. It can also be used for estimation and prediction [48]. KNN is an example of instance-based learning given that it stores the training dataset in the memory. The classification for an unclassified record is based on comparing it with similar records in the training dataset. The distance between the new record and the closest class is measured by a distance function such as Euclidean, Manhattan or Minkowski for continuous variables and Hamming distance for categorical variables [52].

Support Vector Machines

Support Vector Machine (SVM) is a supervised machine learning algorithm based on the concept of decision planes that define decision boundaries for different classes [52]. The task of SVM is to find out the hyperplane that maximises the distance margin between the malicious and benign classes. The support vectors are the extreme points that outline the hyperplane.

IV. Experimental Setup

The experiments proposed in this paper are benchmarked against five popular classifiers used in similar research: Decision Tree, Naïve Bayes, MLP, KNN and SVM. The results obtained from the classifiers are evaluated in order to verify Hypothesis 1 discussed in Section III. The dataset used in this paper is collected from a real testbed which includes equipment such as: Siemens S7-1500 PLC, sensors, and actuators, all currently used in industry. Further, two more datasets are created to evaluate the results obtained from the selected machine learning algorithms when the dataset size grows. The datasets are described in section 3.3. Moreover, we assess three well-known feature selection techniques in order to validate Hypothesis 2 discussed in Section III. In this paper, the experiments were executed in a Laptop MacBook Pro with 2.9 GHz Intel Core i7 and 16 GB 1600 MHz DDR3 of RAM memory. The five selected machine learning algorithms were implemented in the python-based web application called Jupyter [53]. To estimate the performance of the ML algorithms we used the statistical method called k-fold cross-validation procedure (i.e. 5-fold cross validation for our experiments) where the given dataset is to be split into k smaller dataset and then average value is computed. The experimental design considered the features from our very own CWSS dataset that obtained the highest scores in each of the feature selection methods described in the previous sections. The metrics used to evaluate the performance of the machine learning algorithms are F1-Score, Geometric Mean (G-Mean), False-Positive Rate (FPR), False-Negative Rate (FNR), Time Taken to Build the Model, and Time Taken to Test the Model.

F1-Score is a harmonic balance of the precision and recall

We chose F1-Score metric over Accuracy given that unlike Accuracy, F1-Score is not contributed by the large number of true negatives that our model might provide. G-Mean is a performance metric that combines the True Negative Rate (TNR) and True Positive Rate (TPR). A low G-Mean score indicates that the performance of the machine learning algorithm is poor. Additionally, given that triggering a false positive alarm or a false negative alert in critical infrastructure might have a more significant impact in comparison with traditional computer networks, we consider False Positive Rate (FPR) and False-Negative Rate (FNR) as important metrics to evaluate the performance of our machine learning models.

Moreover, it is also important to evaluate two important metrics of: Time Taken to Build the Model and the Time Taken to Test the Model. They have been chosen as it is vital to predict an attack on ICS as fast as possible in order to avoid irreversible damage. For instance, attacking a water treatment system may involve the manipulation of the water chlorination. Modifying the dosage of chlorine in the water would put lives in great danger. On the other hand, the time taken to build the model may not be required to be quick, except in circumstances where it is required to update the model on the fly.

V. Result Analysis

This section presents the analysis and discussion of the results obtained from the five selected machine learning algorithms and three feature selection methods given six performance metrics as discussed before. Table IV shows the results obtained from the machine learning algorithms after employing all the features and also once hiring the ones chosen by each feature selection technique.

TABLE V RESULTS OBTAINED FROM DATASET II

Feature Selection Technique	Algorithm	F1 Score	G-Mean	FPR	FNR	Time to Build the Model (s)	Time Taken to Test the Model (s)
Information Gain	Decision Tree	0.939	0.94	0.018	0.099	2.952	0.006
	Naïve Bayes	0.908	0.912	0.019	0.152	0.257	0.045
	Multilayer Perceptron	**0.955**	**0.956**	**0.025**	**0.063**	**20.794**	**0.013**
	KNN	0.945	0.946	0.038	0.071	6.376	8.683
	SVM	0.954	0.955	0.015	0.075	3499.471	421.428
Chi Square	Decision Tree	0.939	0.94	0.018	0.099	2.512	0.006
	Naïve Bayes	0.885	0.891	0.016	0.194	0.272	0.044
	Multilayer Perceptron	**0.958**	**0.959**	**0.017**	**0.065**	**20.639**	**0.013**
	KNN	0.945	0.946	0.038	0.071	6.119	8.519
	SVM	0.953	0.954	0.014	0.077	3313.695	428.312
Correlation Based	Decision Tree	0.939	0.94	0.018	0.099	3.072	0.007
	Naïve Bayes	0.908	0.912	0.019	0.153	0.282	0.049
	Multilayer Perceptron	**0.958**	**0.959**	**0.017**	**0.065**	**21.405**	**0.014**
	KNN	0.945	0.946	0.038	0.071	6.997	9.241
	SVM	0.954	0.955	0.014	0.076	3482.824	443.72
No Feature Selection Method	Decision Tree	0.939	0.94	0.018	0.099	4.323	0.009
	Naïve Bayes	0.894	0.899	0.017	0.178	0.393	0.079
	Multilayer Perceptron	**0.952**	**0.953**	**0.015**	**0.078**	**24.045**	**0.017**
	KNN	0.946	0.947	0.037	0.069	9.467	10.298
	SVM	0.95	0.951	0.013	0.084	6100.745	733.347

TABLE VI RESULTS OBTAINED FROM DATASET III

Feature Selection Technique	Algorithm	F1 Score	G-Mean	FPR	FNR	Time to Build the Model (s)	Time Taken to Test the Model (s)
Information Gain	Decision Tree	0.94	0.942	0.018	0.097	4.81	0.008
	Naïve Bayes	0.908	0.912	0.019	0.153	0.352	0.066
	Multilayer Perceptron	**0.955**	**0.956**	**0.029**	**0.059**	**30.295**	**0.02**
	KNN	0.959	0.959	0.029	0.053	12.181	13.126
	SVM	0.954	0.955	0.013	0.075	7117.861	865.213
Chi Square	Decision Tree	0.94	0.942	0.018	0.097	4.071	0.009
	Naïve Bayes	0.885	0.891	0.016	0.193	0.367	0.061
	Multilayer Perceptron	**0.96**	**0.96**	**0.017**	**0.062**	**30.509**	**0.02**
	KNN	0.959	0.96	0.029	0.052	11.639	12.777
	SVM	0.953	0.954	0.013	0.077	7391.045	866.156
Correlation Based	Decision Tree	0.94	0.942	0.018	0.097	5.019	0.009
	Naïve Bayes	0.908	0.912	0.019	0.153	0.385	0.071
	Multilayer Perceptron	**0.958**	**0.958**	**0.018**	**0.064**	**30.701**	**0.021**
	KNN	0.959	0.96	0.029	0.052	13.485	14.051
	SVM	0.954	0.955	0.013	0.076	140920.919	909.082
No Feature Selection Method	Decision Tree	0.94	0.942	0.018	0.097	7.228	0.014
	Naïve Bayes	0.894	0.899	0.017	0.179	0.598	0.114
	Multilayer Perceptron	**0.953**	**0.954**	**0.015**	**0.076**	**35.172**	**0.025**
	KNN	0.961	0.961	0.028	0.05	17.67	15.826
	SVM	0.95	0.951	0.012	0.085	12946.696	1476.491

Table V and Table VI show the same results as described above but obtaining from Dataset II and Dataset III, both respectively. According to the results shown in Table IV, the Correlation-Based and IG, as two feature selection techniques, slightly improve the performance of the Naïve Bayes algorithm in terms of F1-Score from 89.4%, when the entire dataset is used, to 90.8%, with only chosen features. However, the Time Taken to Build the Model and the Time Taken to Test the Model, do not show a significant difference for Naïve Bays algorithm in all the scenarios.

Moreover, the F1-Score for the MLP algorithm is improved from 95%, when the entire dataset is used, to 95.4%, when only selected features by the chi-square are employed. The Time Taken to Build the Model for this algorithm is reduced by 2 seconds and the Time Taken to Test the Model remains below 1 second. It should be noted that the SVM does not improve in terms of F1-Score or G-mean metrics, however reducing the number of features aids to reduce the computational time to 6 minutes for the Time Taken to Build the Model and 1 minute for the Time Taken to Test the Model.

TABLE IV RESULTS OBTAINED FROM DATASET I

Feature Selection Technique	Algorithm	F1 Score	G-Mean	FPR	FNR	Time Taken to Build the Model (s)	Time Taken to Test the Model (s)
Information Gain	Decision Tree	0.935	0.936	0.019	0.106	1.106	0.003
	Naïve Bayes	0.908	0.912	0.019	0.152	0.134	0.021
	Multilayer Perceptron	**0.95**	**0.951**	**0.029**	**0.068**	**10.144**	**0.006**
	KNN	0.916	0.918	0.051	0.113	2.103	4.112
	SVM	0.95	0.951	0.015	0.082	819.341	105.368
Chi Square	Decision Tree	0.935	0.936	0.019	0.106	0.914	0.003
	Naïve Bayes	0.886	0.891	0.016	0.193	0.135	0.019
	Multilayer Perceptron	**0.954**	**0.955**	**0.017**	**0.072**	**9.776**	**0.005**
	KNN	0.916	0.918	0.051	0.113	2.01	4.058
	SVM	0.956	0.957	0.014	0.072	720.915	96.698
Correlation Based	Decision Tree	0.935	0.936	0.019	0.106	1.139	0.003
	Naïve Bayes	0.908	0.912	0.019	0.152	0.141	0.021
	Multilayer Perceptron	**0.953**	**0.954**	**0.019**	**0.072**	**10.336**	**0.006**
	KNN	0.916	0.918	0.051	0.113	2.356	4.285
	SVM	0.957	0.957	0.014	0.07	743.715	101.966
No Feature Selection Method	Decision Tree	0.935	0.936	0.019	0.106	1.435	0.005
	Naïve Bayes	0.894	0.899	0.017	0.178	0.181	0.038
	Multilayer Perceptron	**0.95**	**0.954**	**0.015**	**0.076**	**11.901**	**0.008**
	KNN	0.916	0.918	0.051	0.112	3.299	4.847
	SVM	0.961	0.962	0.013	0.063	1181.682	162.983

The results obtained from the machine learning algorithms on dataset II are shown in Table V. The KNN algorithm shows considerable improvement on dataset II. F1-Score and G-mean metrics on dataset I, achieved 91.6% and 91.8% both respectively while on dataset II it achieves 94.5% and 94.6%. It can be seen in Table V that the Time to Build the Model and the Time to Test the Model are increased by a factor of 2 or even sometimes more for the five algorithms. Table VI shows the results obtained from dataset III. Both F1-Score and G-mean metrics obtained by the MLP algorithm on dataset III with the features provided by the Chi-Square achieved 96%. This outperforms the scores of 95.3% and 95.4% achieved by the MPL algorithm when the entire dataset is used. The Time to Build the Model and the Time to Test the Model are increased by the factor of 2 and 4, both respectively, compared to dataset II and dataset I. For instance, it can be seen in Table I that the F1-Score and G-mean metrics from the Naïve Bayes and MLP algorithm both show an improvement compared to the results obtained when the algorithms are trained with the entire dataset.

As Table VI shows, the results obtained from the rest of the algorithms do not show an improvement, however, the F1-Score and G-mean metrics obtained from the algorithm trained with the feature selection techniques are equal to the results obtained from the entire dataset.

VI. Discussion

In this section, the scientific hypothesis described at the beginning of this paper are discussed and the challenges in implementing the mechanism of attack detection in real operational plants.

Hypothesis 1. Newly engineered energy-based features collected from monitoring the energy consumption of the sensors and actuators that compose a model of a clean water supply system in conjunction with well-known supervised machine learning algorithms allow the detection of anomalies that may have a negative impact on the control system.

The evaluation of the machine learning algorithms described in the previous section demonstrates the feasibility of classifying anomalous activity on a model of a clean water system by monitoring the energy of the actuator/sensors that compose the control system. The algorithms that show the best performance regarding F1-Score are MLP and SVM for three datasets. Although, SVM requires significantly more time than MLP in building the machine learning model.

The concept of energy monitoring explained in this manuscript shows the feasibility of using a set of energy-based features for the detection of anomalies in a model of a clean water supply system. It should be considered that the proposed attack detection mechanism operates at layer 0 of the ICS architecture and it is independent of the equipment that compose the control system. For this reason, our detection mechanism cannot be fairly compared to the detection mechanisms explained in related work. We use energy readings at level 0, while the related work uses information obtained from the control network at level 2 of the ICS architecture shown in Figure 2.

Although this implementation is feasible in the testbed explained in this paper, the unknowns of the feasibility of implementing this concept on a large scale arise. The authors in [54] describe the challenges of implementing proofs of concept, such as the one explained in this manuscript, in real operational plants. Our anomaly detection system can be implemented in any type of ICS because it only requires information from sensors or actuators, regardless of the type of process. We encourage research to analyse the feasibility of employing the attack detection mechanism suggested in this paper in other ICS implementations such as power plants, transportation and so on.

VII. Conclusions

This paper proposed an approach for anomaly detection in an ICS based on monitoring the energy of the sensors and actuators. Most of the research is focused on applying supervised and unsupervised machine learning techniques using features extracted from the traffic collected from an ICS network. Unlike this, we propose monitoring the energy consumption of actuators and sensors using a hard-wired INA219 current sensor. Another point to highlight in this paper is that the datasets, that contain benign and malicious traffic, are obtained from a physical testbed which was exclusively implemented for this research.

The results obtained from the experimentation show the feasibility of using this approach for anomaly detection using a wide range of machine learning algorithms. Further, the feature selection techniques applied to the dataset did successfully remove features that did not contribute to the machine learning model. One of the visible advantages of feature selection is the reduction of computational time for heavy algorithms such as SVM and KNN. For instance, on SVM the Time Taken to Build the Model is reduced by 37% when the correlation-based technique is applied to the dataset. The performance of the machine learning algorithms achieved an F1 Score of 90% overall.

In our scenario, we focus on obtaining a high detection rate along with the lowest FPR and FNR. Bearing that in mind, the algorithm that meets those requirements is Multilayer Perceptron (MLP) which achieves an F1 score of 95%, 2.9% FPR and 6.8% FNR, when Information Gain is applied on the dataset.

A. *Future Work*

In future work, we plan to implement an online detection system considering the results presented in this paper. Moreover, we will analyse adversarial scenarios that could affect our implementation.

REFERENCES

[1] B. Esmaeilian, S. Behdad, and B. Wang, "The evolution and future of manufacturing: A review," *J. Manuf. Syst.*, vol. 39, pp. 79–100, 2016, doi: 10.1016/j.jmsy.2016.03.001.

[2] L. A. Maglaras *et al.*, "Cyber security of critical infrastructures." 2018, doi: 10.1016/j.icte.2018.02.001.

[3] E. Byres, "The myths and facts behind cyber security risks for industrial control systems," *Proc. VDE Kongress*, pp. 1–6, 2004, doi: 10.1.1.579.3650.

[4] Cybersecurity Insiders, "Insider Threat 2018 Report," p. 41, 2018.

[5] T. Chen and S. Abu-Nimeh, "Lessons from Stuxnet," *Computer (Long. Beach. Calif).*, vol. 44, pp. 91–93, 2011, doi: 10.1109/MC.2011.115.

[6] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine Blackout: Implications for False Data Injection Attacks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3317–3318, 2017, doi: 10.1109/TPWRS.2016.2631891.

[7] X. Clotet, J. Moyano, and G. León, "A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of Critical Infrastructures," *Int. J. Crit. Infrastruct. Prot.*, vol. 23, pp. 11–20, 2018, doi: 10.1016/j.ijcip.2018.08.002.

[8] A. A. Cárdenas, S. Amin, and Z. Lin, "Attacks Against Process Control Systems : Risk Assessment , Detection , and Response Categories and Subject Descriptors," *Security*, pp. 355–366, 2011, doi: 10.1145/1966913.1966959.

[9] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, 2017, doi: 10.1016/j.jksuci.2015.12.004.

[10] D. Wang, X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning," *J. Inf. Secur. Appl.*, vol. 46, pp. 42–52, 2019, doi: 10.1016/j.jisa.2019.02.008.

[11] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, J. B. Coble, W. Hines, and J. B. Coble, "Multi-Layer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System and Process Data," *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1–1, 2019, doi: 10.1109/tii.2019.2891261.

[12] S. Ponomarev and T. Atkison, "Industrial Control System Network Intrusion Detection by Telemetry Analysis," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 2, pp. 252–260, 2016, doi: 10.1109/TDSC.2015.2443793.

[13] A. Keliris, H. Salehghaffari, B. Cairl, P. Krishnamurthy, M. Maniatakos, and F. Khorrami, "Machine learning-based defense against process-Aware attacks on Industrial Control Systems," *Proc. - Int. Test Conf.*, pp. 1–10, 2017, doi: 10.1109/TEST.2016.7805855.

[14] P. Schneider and K. Böttinger, "High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks," pp. 1–12, 2018, doi: 10.1145/3264888.3264890.

[15] M. Kravchik and A. Shabtai, "Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks," 2018.

[16] A. A. Abokifa, K. Haddad, C. Lo, and P. Biswas, "Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–Based Anomaly Detection Techniques," *J. Water Resour. Plan. Manag.*, vol. 145, no. 1, p. 04018089, 2018, doi: 10.1061/(asce)wr.1943-5452.0001023.

[17] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, and A. Y. Zomaya, "An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 5, pp. 893–906, May 2016, doi: 10.1109/TIFS.2015.2512522.

[18] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," *2016 Int. Work. Cyber-physical Syst. Smart Water Networks, CySWater 2016*, no. Figure 1, pp. 31–36, 2016, doi: 10.1109/CySWater.2016.7469060.

[19] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," *Proc. - 2017 3rd Int. Work. Cyber-Physical Syst. Smart Water Networks, CySWATER 2017*, pp. 25–28, 2017, doi: 10.1145/3055366.3055375.

[20] iTrust, "Secure Water Treatment (SWaT) Testbed," no. October, 2018.

[21] J. Feld, "PROFINET - scalable factory communication for all applications," in *IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.*, pp. 33–38, doi: 10.1109/WFCS.2004.1377673.

[22] Qing Liu and Yingmei Li, "Modbus/TCP based Network Control System for Water Process in the Firepower Plant," in *2006 6th World Congress on Intelligent Control and Automation*, 2006, pp. 432–435, doi: 10.1109/WCICA.2006.1712353.

[23] O. Kwon, Taeyean and Lee, Jaehoon and Yi, "Vulnerability Analysis and Security Modeling of MODBUS," *Adv. Sci. Lett.*, vol. 22, no. 9, pp. 2246–2251, 2016, doi: 10.1166/asl.2016.7793.

[24] J. Singh, S. Dhariwal, and R. Kumar, "A detailed survey of ARP poisoning detection and mitigation techniques," *Int. J. Control Theory Appl.*, vol. 9, no. 41, pp. 131–137, 2016.

[25] Siemens, "Our fastest controller for automation," 2018. [Online]. Available: https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc/simatic-s7-1500.html. [Accessed: 09-Nov-2015].

[26] FESTO, "MPS PA Compact Workstation with level, flow rate, pressure and temperature controlled systems," 2015. [Online]. Available: https://www.festo-didactic.co.uk/gb-en/learning-systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and-temperature-controlled-systems.htm?fbid=Z2IuZW4uNTUwLjE3LjE4Ljg4Mi40Mzc2. [Accessed: 07-Jul-2018].

[27] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, G. Russell, and I. Maneru-Marin, "PLC Memory Attack Detection and Response in a Clean Water Supply System," *Int. J. Crit. Infrastruct. Prot.*, May 2019, doi: 10.1016/j.ijcip.2019.05.003.

[28] A. Robles, "Packet Crafting, Scapy and S7-1500 PLC." [Online]. Available: https://github.com/andrex17/ics. [Accessed: 06-Apr-2019].

[29] A. Robles-Durazno, N. Moradpoor, J. Mcwhinnie, and G. Russell, "A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system," in *In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2018)*, 2018, pp. 1–8, doi: 10.1109/CyberSecPODS.2018.8560683.

[30] Adafruit, "INA219 HIGH SIDE DC CURRENT SENSOR BREAKOUT - 26V ±3.2A MAX," 2018. [Online]. Available: https://www.adafruit.com/product/904. [Accessed: 15-Jan-2019].

[31] R. Pi, "Raspberry Pi 3 Model B," 2019. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 08-Nov-2018].

[32] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Guide to Industrial Control Systems (ICS) Security," 2015.

[33] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *Advances in Intelligent Computing*, 2005, pp. 878–887.

[34] K. W. P. Chawla, N.V., Bowyer, K.W., Hall, L.O., "SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[35] J. Miao and L. Niu, "A Survey on Feature Selection," *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 919–926, 2016, doi: 10.1016/j.procs.2016.07.111.

[36] M. Hansen, M. Haugland, T. Sinkjær, and N. Donaldson, "Real time foot drop correction using machine learning and natural sensors," *Neuromodulation*, vol. 5, no. 1, pp. 41–53, 2002, doi: 10.1046/j.1525-1403.2002._2008.x.

[37] Y. Hirai *et al.*, "A Biomedical Sensor System with Stochastic A/D Conversion and Error Correction by Machine Learning," *IEEE Access*, vol. 7, pp. 21990–22001, 2019, doi: 10.1109/ACCESS.2019.2898154.

[38] R. G. Lyons, *Understanding Digital Signal Processing*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1996.

[39] J. Novaković, P. Strbac, and D. Bulatović, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugosl. J. Oper. Res.*, vol. 21, no. 1, pp. 119–135, 2011, doi: 10.2298/YJOR1101119N.

[40] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: A survey," *Appl. Comput. Informatics*, no. xxxx, 2019, doi: 10.1016/j.aci.2019.01.001.

[41] S. Shrivastava, S. Adepu, and A. Mathur, “Design and assessment of an Orthogonal Defense Mechanism for a water treatment facility,” *Rob. Auton. Syst.*, vol. 101, pp. 114–125, 2018, doi: 10.1016/j.robot.2017.12.005.

[42] S. Adepu, S. Shrivastava, and A. Mathur, “Argus: An Orthogonal Defense Framework to Protect Public Infrastructure against Cyber-Physical Attacks,” *IEEE Internet Comput.*, vol. 20, no. 5, pp. 38–45, 2016, doi: 10.1109/MIC.2016.104.

[43] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *J. Comput. Sci.*, vol. 25, pp. 152–160, 2018, doi: 10.1016/j.jocs.2017.03.006.

[44] H. H. Pajouh, A. Dehghantanha, R. Khayami, and K. K. R. Choo, “Intelligent OS X malware threat detection with code inspection,” *J. Comput. Virol. Hacking Tech.*, vol. 14, no. 3, pp. 213–223, 2018, doi: 10.1007/s11416-017-0307-5.

[45] S. Chandra, Z. Lin, A. Kundu, and L. Khan, “Towards a Systematic Study of the Covert Channel Attacks in Smartphones,” *Int. Conf. Secur. Priv. Commun. Syst.*, pp. 427–435, 2014.

[46] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1200–1205, doi: 10.1109/MIPRO.2015.7160458.

[47] P. O’Kane, S. Sezer, K. McLaughlin, and E. G. Im, “SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 3, pp. 500–509, Mar. 2013, doi: 10.1109/TIFS.2013.2242890.

[48] H. Qu, J. Qin, W. Liu, and H. Chen, “Instruction Detection in SCADA/Modbus Network Based on Machine Learning,” in *Machine Learning and Intelligent Communications*, 2018, pp. 437–454.

[49] D. Wu *et al.*, “Cybersecurity for digital manufacturing,” *J. Manuf. Syst.*, 2018, doi: 10.1016/j.jmsy.2018.03.006.

[50] M. C. Belavagi and B. Muniyal, “Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection,” in *Procedia Computer Science*, 2016, doi: 10.1016/j.procs.2016.06.016.

[51] R. Kabore, Y. Kermarrec, and P. Lenca, “Performance Comparison For Multi Class Classification Intrusion Detection In SCADA Systems Using Apache Spark.” p. ., Sep-2018.

[52] S. Jain, S. Shrivastava, Z. Saquib, S. Shah, and A. Rodrigues, “Identification of High Pressure Critical Links in Water Distribution Systems,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–7, doi: 10.1109/ICCUBEA.2018.8697566.

[53] M. Ragan-Kelley *et al.*, “The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication.,” in *AGU Fall Meeting Abstracts*, 2014, vol. 2014, pp. H44D-07.

[54] S. Adepu, N. K. Kandasamy, and A. Mathur, “EPIC: An electric power testbed for research and training in cyber physical systems security,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11387 LNCS, no. November, pp. 37–52, 2019, doi: 10.1007/978-3-030-12786-2_3.

[55] MATLAB, “Noise Reduction GUI using Low Pass Filter,” 2018. [Online]. Available: https://ww2.mathworks.cn/matlabcentral/fileexchange/15329-noise-reduction-gui-using-low-pass-filter?s_tid=FX_rc2_behav. [Accessed: 09-Nov-2020]