Monitoring Approaches for Security and Safety Analysis: Application to a Load Position System

Zujany Salazar
Research & Development Dept.
Montimage EURL
Paris, France
zujany.salazar@montimage.com

Ana Rosa Cavalli
Research & Development Dept.
Montimage EURL
Paris, France
ana.cavalli@montimage.com

Wissam Mallouli
Research & Development Dept.
Montimage EURL
Paris, France
wissam.mallouli@montimage.com

Filip Sebek
RD Motion & Control
Marine & Ports, ABB AB
Västerås, Sweden
filip.sebek@se.abb.com

Fatiha Zaidi
CNRS,ENS Paris-Saclay, Laboratoire Méthodes Formelles 91190
Université Paris-Saclay
Gif-sur-Yvette, France
fatiha.zaidi@universite-paris-saclay.fr

Monika Ewa Rakoczy
Research & Development Dept.
Montimage EURL
Paris, France
monika.rakoczy@montimage.com.com

***Abstract*—Safety monitoring of Industrial Control Systems (ICS) is a must for optimal operation of safe manufacturing facilities. Failures and miss-behaviours seldomly occur without prior warning, but these warnings are often subtle, requiring careful analysis of data by experienced personnel for early detection. Monitoring function allows to promptly take adequate corrective actions in order to maximize uptime and increase trust of running industrial systems. In this paper, we present two main approaches of monitoring techniques implemented in the Montimage MMT tool. The first approach is a signature-based approach, where there are safety properties to be checked on the ICS logs, and the other relies on Machine Learning (ML) to detect anomalies. Both methods have been applied to check safety on an industrial system: a crane load position system provided by ABB. Several experiments have been performed to check if the information provided by a system's PLC is correct, guarantying the safety of the system.**

***Index Terms*—Monitoring techniques, Industrial Control Systems (ICS), signature-based analysis, anomaly detection, ML-based analysis, MMT tool.**

I. Introduction

Industrial Control Systems (ICS) is a generic term that refers to different types of control systems that operate and/or automate industrial processes. These systems consist of a combination of devices, software, and networks that together achieve an objective, such as manufacturing a product, water treatment, energy generation, etc.

Managing safety during operation is one of the main objectives of evolving ICS systems. Besides, cybersecurity is playing an essential role in process safety systems. In this sense, monitoring of security and safety properties has become an important part of today's software development projects where different pieces of code from several providers can be used and integrated into evolving ICS environments [1]. In particular, in these systems, the cyber-physical infrastructures must guarantee the protection of the traditional (physical) elements, such as sensors, controllers and actuators; as well as the novel (cyber) capabilities, in terms of computing and communication protection [2]. The topic has been attracting increasing attention since the Stuxnet incident when the successful cyber-physical sabotage of a uranium enrichment plant in Iran took place [3]. More recently, the coronavirus pandemic has increased the interest of industry actors on cybersecurity operation technology [4].

Procuring cybersecurity in ICS becomes more critical over time because of the imperfections of the existing protection tools, and the increasing presence of vulnerabilities. Compared with the previous year's data, the proportion of vulnerabilities that have a high or critical severity score has grown. More than half of the vulnerabilities identified in ICS systems were assigned CVSS v.3.0 [5] base scores of 7 or higher, corresponding to a high or critical level of risk. Kaspersky has also reported the main challenges according to industrial companies: their results show that companies are concerned of safety of employees, as well as damaging of products and services, and loss of proprietary or confidential information, due to cybersecurity incidents [4]. Therefore, in today's context, cybersecurity is a priority in industrial systems, and it affects factors as important as the lives of employees, or the quality of a company's products.

Nevertheless, ensuring safety requires that relevant actors inside the company acquire proper knowledge and skills to guarantee security at deployment and operation phases, such that a system can resist attacks and handle security errors appropriately [6]. They also need to be supported by tools to ensure dynamic risk management techniques [7], [8], for the automatic detection of missbehaviours and vulnerabilities and their mitigation at runtime.

Currently, there are two main monitoring approaches that are widely used to identify possible breaches in computer systems: signature-based and anomaly-based intrusion detection. Both methods attempt to verify security properties, which also aim to guarantee safety in the systems under monitor. However, both methods have limitations. In the case of a

specific case study like the one we will describe in this paper, signature-based monitors may be difficult to configure, since there is no clarity on the security properties that must be monitored to avoid the situation that compromises the safety of the system. system. Whereas anomaly detection algorithms tend to always be a bit less accurate, and having important rates of false-negatives results.

The research described in this paper was designed as a case study to investigate the use of two different security monitoring approaches, in order to provide a solution to a safety problem in an Industrial Control System. There were three research objectives for this case study:

1) To explore the applicability of signature and ML based monitoring techniques to solve a safety problem in a Load Position System
2) To gain understanding of the drawbacks and benefits of each of the proposed monitoring techniques
3) To measure the accuracy of the proposed monitoring techniques in the classification task proposed in the case study described in Section II

The paper is organised as follows. Section II describes the case study proposed by ABB. Section III presents related work. Section IV shows the monitoring solution in terms of architecture and functionalities. Section V presents the experimentation environments and the results in the context of the industrial crane LPS system. Finally, section VI gives the conclusion and future work.

II. Case Study Description

The following subsections describe the context of this research, which is the European project VeriDevOps, and its concrete case study that motivated this research.

A. VeriDevOps projet

VeriDevOps [1] project addresses automation of verification methods and monitoring for prevention and protection of industrial control systems. The innovation resides in using Natural Language Processing (NLP) to automatically generate formal specifications of security requirements, that normally are written in natural language. The idea is to provide a way of preventing inconsistencies from propagating into operations and identifying the faults that could be introduced in the requirements. Therefore, this formalized security requirements will be leveraged to perform monitoring activities at the operations stage of the system, in order to verify security and safety.

B. ABB Case Study

ABB Load Position System (LPS) is a camera-based tracking system that determines a hanging crane load position with help of attached LED markers on the load. As can be seen in Figure 1, the system consists on a camera on top of a crane that keeps tracks of LED markers in a load that is in motion. In addition, there is single Programmable Logic Controller (PLC) that analyses the camera images.

The workflow of ABB LPS is as follows:

1) The camera captures images of the moving load
2) The camera performs a local analysis of the picture, and it produces an array of coordinates of what the camera thinks are markers a.k.a. *marker candidate list*, that include the real markers to monitor and noise in the image
3) The PLC receives the marker candidate list
4) The PLC, based on statistical algorithms (proprietary to ABB, and therefore confidential), makes a selection of the real markers and discard the noise
5) The maker spatial coordinates are used for the system to know the current position of the load, and calculate the following movements of the crane

However, it has happened that noise detected by the camera is tagged as markers by the PLC. This situation might lead to incorrect movements of the crane and compromise the safety of the system.

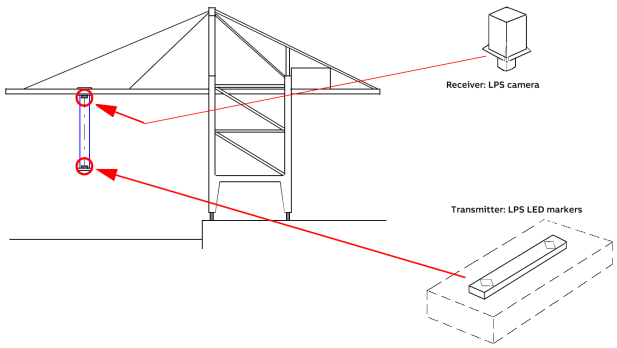

Fig. 1. The Load Position System

The monitoring techniques utilized in this paper aim to perform a classification task, in order to indicate if the selection of markers made by the PLC was valid or not. For such a purpose, the two techniques aim to find the real markers based on a group of features that are related to the LPS configuration, and the spatial coordinates of the candidates.

It is important to notice, that the utilized techniques do not attempt to find the real markers themselves, but to verify the PLC results, based on a group of features that are related to the LPS configuration, and the spatial coordinates of the candidates.

For the purposes of this research work, two LPS configurations were simulated by ABB. Both configurations differs in terms of internal parameters of the LPS system, for example the model of the ABB crane, or the positions of the markers in the load. Finally, system configuration number two was recorded two times in different days, which varies the load and environmental conditions, such as the weather. More details about the data recorded from the simulations is provided in Section V-A and in Table I.

[1] https://www.veridevops.eu/veridevops

Figures 2 and 3 depict examples of three different records performed in the two simulated LPS system configurations. In the pictures, it is possible to see that spatial information is not enough to determine the validity of a marker; therefore, another features must be considered.

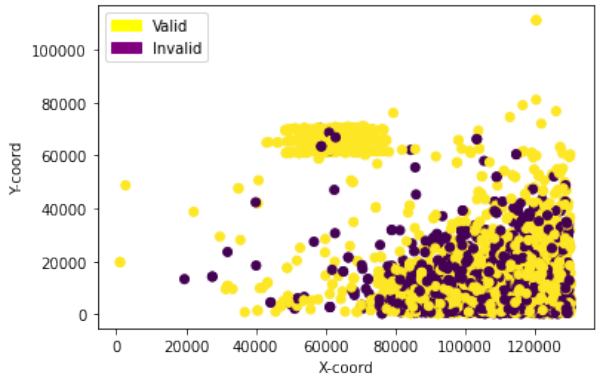

Fig. 2. Spatial representation of the markers: System configuration 1

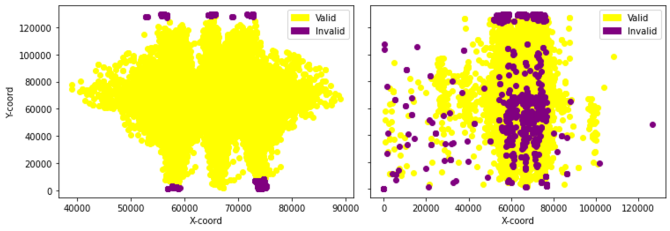

Fig. 3. Spatial representation of the markers: System configuration 2-a (right), 2-b (left)

III. Related Work

The monitoring of approaches are generally divided into two main categories. The first one relies on formal properties to be checked on collected traces, and the second one uses artificial intelligence algorithms to detect drifts and anomalies. These two categories will be presented in the next two subsections.

A. Signature-based monitoring

Signature-based Intrusion Detection Systems (SIDS) are monitor tools that compare captured system or network events to a set of predefined malicious activity patterns. Therefore, if one of these previously recorded behaviors is detected, an alert is triggered.

One of the main problems with this approach is that attacks are already described/modelled in a specific manner; therefore, a simple mutation could make them undetectable. Furthermore, the choice of algorithms to accurately and efficiently detect malicious behaviors and intrusions [9], and how to identify attacks that span across several events [10] remain also open issues.

Regarding Industrial Control Systems, Richey's thesis [11] aims to leverage the static topology of ICS networks and those programs that define them to enhance the IDS's knowledge of the environment in which it is deployed. The author describes a method for automatically generating rules and signatures to detect possible intrusions, by parsing PLC ladder logic to extract address register information, data types and usage. Moreover, a Ladder Logic Parser program was created to test the proposed method, showing that it is not only applicable to a controlled test environment, but can also create a significant number of Snort rules that define abnormal behavior using real-world ladder files. Using a smaller test case ladder file, the functionality of this method was proven accurate and a sampling of the larger real-world files were found to be thorough and valid.

B. ML based anomaly detection

Anomaly-based Intrusion Detection Systems (AIDS) work by comparing the actual comportment with a previously-established "normal" model of the behavior of the system. Any substantial deviance between the observed behavior and the model is considered as an anomaly, which can be translated as an intrusion or attack into the system. AIDS has drawn interest from a lot of scholars due to its capacity to overcome the limitation of the Signature-based Intrusion Detection Systems (SIDS) [10].

Normally, in AIDS, the normal model of the behavior of a computer system is created using machine learning, statistical-based, or knowledge-based methods. We will consider only the machine learning method.

The applications of machine learning techniques in the design of Intrusion Detection Systems (IDS) have remained a trend in the last few years [12]. Therefore, there have been numerous anomaly-based IDS prototypes that implement these techniques.

Recently, there is an increasing interest in the use of Deep Learning techniques as a prospective method for the next generation of IDSs due to their capability of automatically finding correlations in data [12], [13]. Several author have summarized various intrusion detection mechanisms using a combination of machine learning and deep learning approaches [14], [15].

IV. Monitoring Approaches used in ABB case study

The following subsections describe the existing monitoring approach techniques that we used to solve the problem stated in the ABB case study, described in Section II.

A. Global monitoring architecture

The monitoring approaches are intended to be performed during ICS operations. Figure 4 depicts the global workflow of the monitoring activities.

First, application logs from the LPS system are taken as inputs. Logs can be analyzed both online and offline. Therefore, near real-time monitoring or analysis of previously saved logs is possible.The feature extraction module analyzes the data, and extracts the selected features to be considered in both monitoring approaches. The data is then processed by the SIDS and AIDS components. Finally, the results are

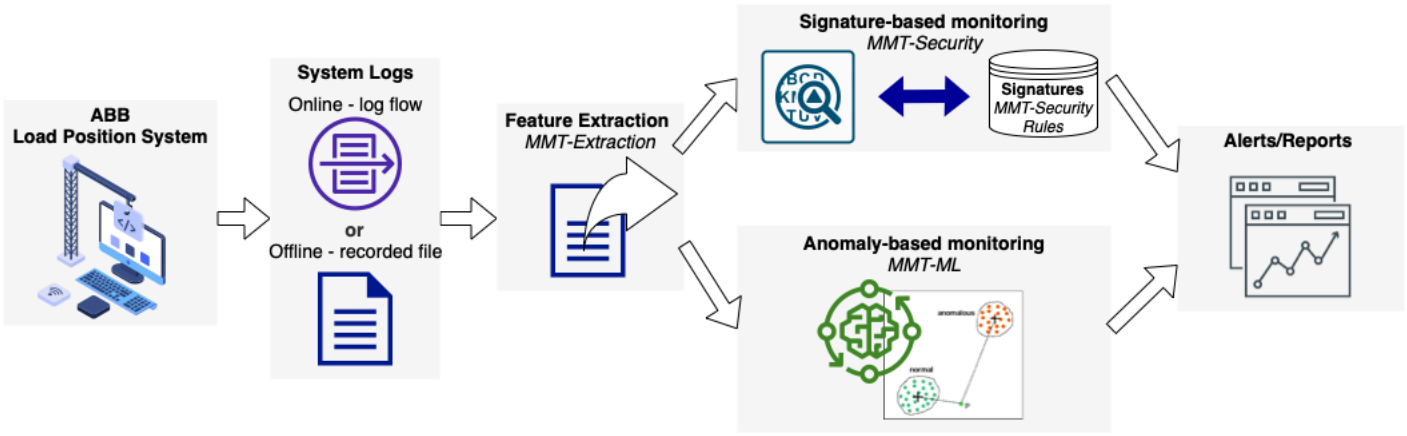

Fig. 4. Monitoring architecture

displayed in a dashboard with relevant graphs that summarize the status of the system and display alerts when security flaws are suspected or detected. Section 4.1 describes each of the modules that compose the main architecture.

For the experimentation described in this paper, only offline analyzes were performed.

a) Features Extraction: The feature extraction task is performed by MMT-Extract [16], a C library that analyzes network traffic and application logs, to extract network and application-based events. Extraction is powered by a plugin architecture that allows adding new protocols or application message formats to parse. In this case study, new plugins were added to the library to analyze data from the ABB system, which uses internal ABB protocols.

b) Signature-based monitoring: MMT-Security [16] is a signature-based monitoring solution, that allows analysing network traffic according to a set of properties called MMT-Security properties. These properties contain signatures that formally specify security goals, or malicious behaviours related to the monitored system.

The MMT-Security property model is inspired by Linear Temporal Logic and can refer to two types of properties:

- Properties that describe the normal, legitimate behaviour of the application or protocol under analysis. In consequence, the non-respect of the property indicates a potential violation of a safety or security requirement; e.g., all the ports in a computer must be closed unless they are being used by an authorised application.
- Attacks that describe malicious behaviour corresponding to an attack model, a vulnerability or misbehaviour. In this case, the respect of the property indicates the detection of a potential incident; e.g., a big number of requests in a short period of time could be a denial-of-service attack.

XML format was chosen as the language of MMT-Security properties, due to its simplicity and straightforward structure verification. A property is a general ordered tree as shown in Figure 5, where the leaf nodes are the atomic events captured in the traces. Each property is composed of a context, in the left branch, and a trigger, in the right branch. Then a property is valid when the trigger is valid, and the trigger is inspected only if the context is valid. Figure 9 show an example of one MMT-Security property implemented in the context of ABB case study, according to the strategy *NoOfCandidates* detailed below in this Section. Syntax of MMT-Security properties is detailed in [16].

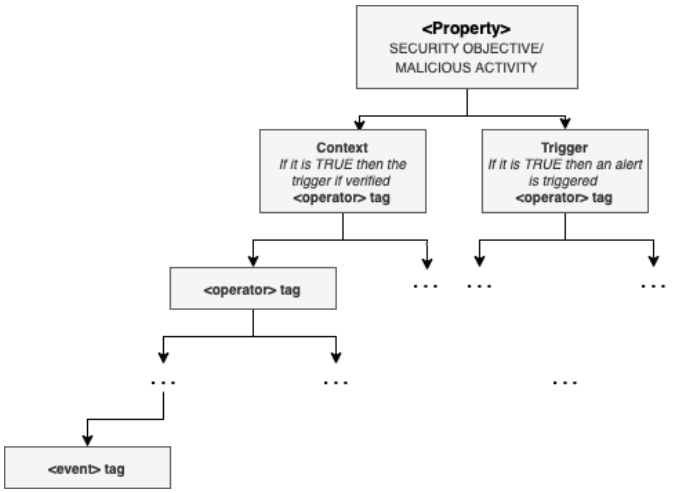

Fig. 5. Security property structure [16]

Determining MMT-Security properties using the PLC features, such as the trolley and hoist positions and speeds, depicted in Figure 7, is not a trivial task. There is not a direct relation between the spatial coordinates, trolley position and speed, main hoist position and speed, nor the gantry speed and the validity of the markers. Therefore, for a non-expert on the LPS system, it would be a difficult task to create MMT-Security properties using these features that allow to monitor and validate the PLC verdict.

However, the relationship between the number of marker candidates and the validity of markers seems more direct, once border values are determined, as shown in Figures 6 and 8. Moreover, considering the hypothesis that since the

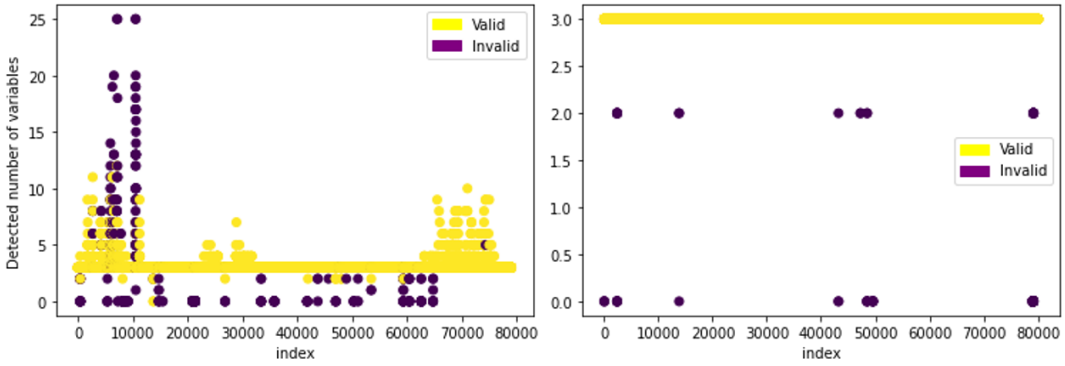

Fig. 6. Relationship between the number of marker candidates and their validity (Right: System conf. 2-a, Left: System conf. 2-b)

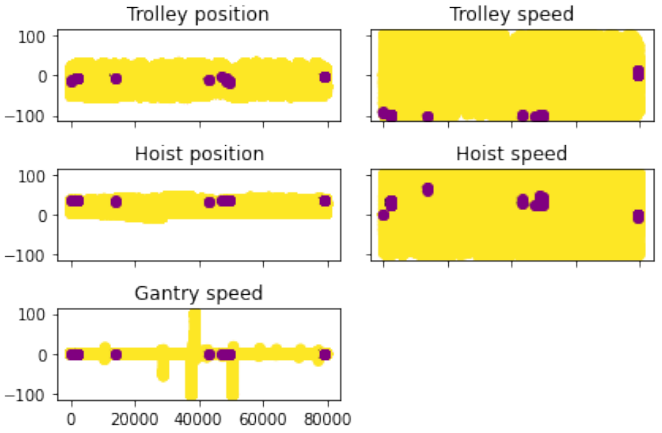

Fig. 7. Data features calculated by the PLC (System conf. 2-b), and their relation with the validity of the markers (Yellow: valid markers, Purple: invalid markers)

markers are in constant motion there should be no sudden jumps between one marker and another, we propose three other strategies that aim to detect these jumps. Below we present a list of the proposed strategies, which we specify as MMT-Security properties to verify them on the ABB data, and provide a solution to the marker classification problem:

1) **Number of marker candidates (NoOfCandidates):** Figure 6 and 8 show that all valid verdicts occurred in a specific region of number of marker candidates, outside of with the verdicts were invalid. Therefore, the number of marker candidates must be within a certain range. Violation of this property would mean that the selected markers are potentially noise.
2) **Speed variation:** The variation speed of a marker (displacement/ time interval) must be less than the specified threshold. Violation of this property would mean that the selected markers are potentially noise.
3) **Displacement magnitude:** The magnitude of the displacement between the markers must be less than the specified threshold. Violation of this property would mean that the selected markers are potentially noise.
4) **Uneven displacement:** The magnitude of the displacement of the markers must be even. Violation of this property would mean that the selected markers are potentially noise.

All the strategies defined above require determining threshold values. For that purpose, we use the optimization algorithm Dual Annealing [17]. This is a well-known stochastic global optimization algorithm, intended for objective functions that have a non-linear response surface, as was the case for all objective functions of the strategies mentioned above. It is a stochastic optimization algorithm, therefore a candidate solution is randomly modified, and new solutions probabilistically replace the current candidate solution. Consequently, worse solutions may replace the current candidate solution. The probability of this type of replacement rises at the beginning of the search and decreases with each iteration, controlled by the hyperparameters of the algorithm. One of the main limitations of Dual Annealing is its computational cost, but it is able to find a global maximum and not get stuck in local minima in problems where the exact algorithms fail, although it usually provides an approximation.

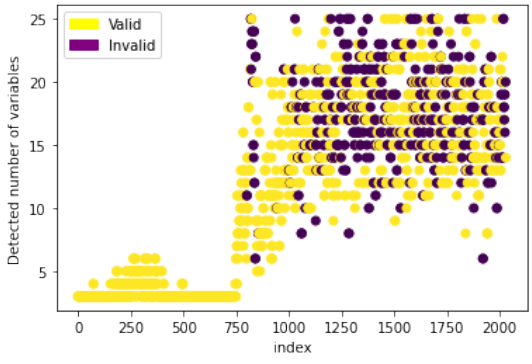

Fig. 8. Relationship between the number of marker candidates and their validity (System conf. 1)

c) Machine Learning based anomaly detection: For performing the anomaly-based monitoring task, we used a Multilayer Perceptron (MLP) classifier [18], with two hidden layer with 4 neurons, that we determined based on the size of

```
<beginning>
<!-- Property 90: Number of markers different of 3.-->

<property value="THEN" delay_units="ms" delay_min="0" delay_max="0" property_id="91" type_property="ATTACK"
    description="Number of markers different to 3">
<event value="COMPUTE" event_id="1"
    description="Marker frame"
    boolean_expression="(ABB_CAMERA.type == 7)"/>
<event value="COMPUTE" event_id="2"
    description="Number of markers different to 3"
    boolean_expression="(ABB_CAMERA.marker_number != 3)"/>
</property>
</beginning>
```

Fig. 9. Example of MMT-Security property, to detect three marker candidates

the input layer and output layer, according to the empirical results of [19]. Regarding the regularization parameter alpha, which contributes to avoid overfitting by penalizing weights with large magnitudes, we used $\alpha = 0.001$, based on our own experimentation. As for the activation function, we used the widely-used activation function: the Rectified Linear Unit (ReLU), defined as $f(x) = max(x, 0)$. One of the main advantages of ReLU function is that it does not activate all the neurons at the same time, therefore Neural Networks (NN) that use ReLU have been proved to be more easily optimizable than neural networks that use other activation functions, such as sigmoid or tanh units [20]. For the solver for weight optimization, we used Adam solver [21], a first-order gradient-based optimization algorithm that is straightforward to implement, requires little memory, is computationally efficient, and is suitable for problems that are large in terms of data with thousands of training samples or more, as it was our case. Finally, we used a gradually decreasing learning rate to decrease the overfitting chances. Figure 10 resumes the utilized MLP model. The used features are described in Section V-A and summarized in Table II.

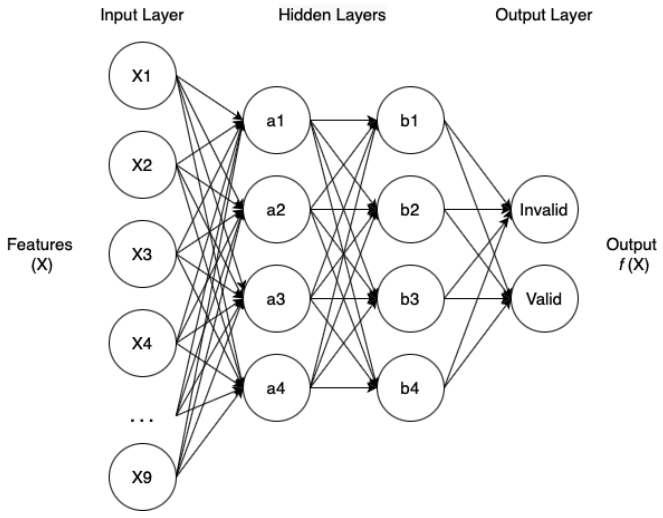

Fig. 10. Two hidden layer MLP

V. Experimentation

In the following subsections we describe the experimentation we did applying the two proposed monitoring approaches.

A. Data

Table I shows the datasets utilized in this experimentation. Data was extracted of two different ABB LPS system configurations. Each entry of the datasets contain the group of candidate markers, the PLC verdict, that indicates which of the candidates are the real markers, and the validation of the PLC verdict, performed manually by experts on the system from ABB. Assuming that in a stand still trolley, the load will not be able to suddenly jump several meters, experts used an ABB visualization tool, that allowed to see jumpy and jerky positions that were physically impossible for a 30 ton container to do. Consequently, in such a cases if the PLC selected those type of markers, they could determine its verdict was wrong.

All the datasets contain features calculated by the PLC of the corresponding system, Figures 7, 6, and 8 depict the relation between these features and the validity of the markers.

Besides, relevant attributes were extracted from the original data, and correlated to compute new features: displacement between consecutive markers (euclidean distances), and speed variation between two consecutive markers. All the features are summarized in the Table II.

For performing the ML experiments, we used the three datasets described in Table I. Machine Learning applications require important amounts of data containing fair representations of each of the classes, in this case: valid and invalid markers. Table I depicts that the three studied datasets were highly unbalanced. Therefore, in order to generate new samples of under-represented class, we utilized random oversampling with replacement of the current available samples, in both training and testing dataset. Moreover, training dataset was divided using proportions 75% and 25% to create the training and verification datasets respectively. Finally, as the chosen ML model is sensitive to feature scaling, we scaled each attribute to values between $[-1, +1]$.

B. Signature-based analysis results

Table III shows the partial results we obtained by using the proposed strategies on the dataset *System conf.1*. Though

TABLE I
LPS DATASETS

Name	Description	Total number of entries	Number of valid PLC verdicts	Number of invalid PLC verdicts
System conf.1	Recording of the system configuration 1. In this record the system behave in a erratic way, more noise in the camera image were detected, so the marker selection process made by the PLC was more difficult than in the datasets 2-a and 2-b	2031	1547	484
System conf.2-a	1st recording of the system configuration 2	79102	78327	775
System conf.2-b	2nd recording of the system configuration 2, registered under the same crane and LPS configuration as the dataset conf.2-a, but in a different day, therefore different weather conditions	80057	79942	115

TABLE II
FEATURES

ID	Features name	Description	Type
X1	Coordinates	Spatial coordinates of markers	PLC
X2	TrolleyPos	Trolley position	
X3	Hoistpos	Hoist position	
X4	TrSpd	Trolley speed	
X5	GaSpd	Gantry speed	
X6	MhSpd	Hoist speed	
X7	NoOfCandidates	Number of marker candidates found including noise	
X8	Speed	Variation speed of a marker (displacement/ time interval)	Calculated
X9	Displacement	Magnitude of the displacement between the markers	

still insufficient, the number of marker candidates (NoOfCandidates) probed to be the most accurate method. The other proposed strategies that aimed to detect jumps in the spatial coordinates of the markers, were not effective and must be substituted. The best accuracy obtained with signature-based analysis did not exceed 73%.

C. ML-based analysis results

The results on both system configurations, using different features are summarised in Tables III.

For the first system configuration, we were disposed only of a single dataset, therefore we have performed validation (on the divided $75\%-25\%$ dataset), without a possibility of testing the model on a new dataset (testing set). However, the results demonstrate that the model is able to predict the validity of PLC verdicts with over 74% accuracy.

Regarding the second system configuration, we were able to validate and then test the model. Finally, Table III also shows the results of the testing of the model using a completely new, unseen dataset, that was recorded on the same system configuration but at different time. Then, it contains different environmental conditions than the training dataset. In this case, the model reach accuracy close to 100%, using the marker candidate number, and the spatial coordinates of the markers as features.

As for the results in the cases when training and testing data came from different system configurations, we did not get good prediction accuracy. Therefore, the proposed model is not adequate to these scenarios. Nevertheless, occurrence of such a situation would be very rare in real-life scenarios, as normally the chosen ML model can always be tuned and adapted to a specific system configuration.

D. Discussion

We have proposed two main approaches to validate ABB PLC verdicts, and guarantee security and safety at system operations. The signature-base monitoring approach presented important limitations, and in general we observed that in this type of case studies a deep knowledge of the system, and in particular of its variables. This is not always possible, and even with the required system knowledge, the task of studying all the variables involved and the use of them to write a property that is ready to be checked on the system is very hard. In this specific case study, for humans it was possible only to find single-features properties, as showed in Table III. However, while using ML techniques which correlated the features in an automatic manner, we were able to obtain much better results. In general, using multi-features relationships, generally difficult to obtain manually, the design of security and safety properties has become easier.

ML results were promising, in particular when the system behave normally, as it is the case of the datasets *System conf.2-a* and *2-b*. In more erratic scenarios, as *System conf.1*, the results are less good, but the model have been proved to predict validity of the markers with an accuracy over the 74%.

As for the features, for the signature-based analysis performed on the system configuration 1, best results were obtained by using the number of marker candidates (*NoOfCandidates* feature) found including noise. This is expected, because this feature measures in someway the amount of noise that the sensors process in a certain moment. Therefore, when there is a big amount of marker candidates, it implies a big amount of noise, and the markers are more probable to be incorrect. On the other hand, if the marker candidates are below a minimum, this can also indicate a problem in the sensor lecture. The optimization process to find the corresponding maximal and minimum values, based on evidence, is simple, as well as the design of a safety rule to monitor this variable. Nevertheless, this feature proved to not be enough for classifying all the markers, and the rest of the tried features that aimed to detect jumps in the markers were not effective. Finally, manually designing properties based on the PLC features was not a trivial task, Figure 7 depicts how the relation between the validity of markers and PLC features is not straightforward.

Using ML techniques *NoOfCandidates* was very effective when testing on system conf.2, but in the erratic behaviour of system conf.1, the feature did perform well enough. As for the spatial information (i.e *Coordinates* feature), it had

TABLE III
SUMMARY OF RESULTS

Monitoring approach	Dataset		Features	Accuracy	Recall invalid markers	Recall valid markers	Precision invalid markers	Precision valid markers
Signature-based detection	System conf.1		NoOfCandidates	**0.73**	**0.79**	**0.71**	**0.46**	**0.92**
			Speed variation	0.60	0.19	0.73	0.18	0.74
			Displacement magnitude	0.59	0.77	0.54	0.34	0.88
			Uneven displacement	0.63	0.63	0.63	0.35	0.84
	Training data	**Validation/Testing data**						
ML Anomaly detection	System 1 (0.75)	System 1 (0.25)	NoOfCandidates, Coordinates	0.74	0.84	0.64	0.70	0.80
			NoOfCandidates, Coordinates, Speed	0.74	0.83	**0.65**	0.70	0.79
			Coordinates, Displacement, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	**0.78**	**0.97**	0.59	0.70	**0.95**
			NoOfCandidates, Coordinates, Displacement, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	**0.78**	0.95	0.61	**0.71**	0.93
			NoOfCandidates, Coordinates, Speed, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0.74	0.87	0.60	0.68	0.83
			NoOfCandidates, Coordinates, Displacement, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0.76	0.92	0.61	0.70	0.88
ML Anomaly detection	System 2-a	System 2-b	NoOfCandidates	**1,00**	**1,00**	**1,00**	**1,00**	**1,00**
			Coordinates	0,97	**1,00**	0,95	0,95	**1,00**
			TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,58	0,69	0,48	0,57	0,61
			NoOfCandidates, Coordinates	**1,00**	**1,00**	**1,00**	**1,00**	**1,00**
			NoOfCandidates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,98	0,96	**1,00**	**1,00**	0,96
			Coordinates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,51	0,06	0,96	0,59	0,50
			NoOfCandidates, Coordinates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,92	0,88	0,96	0,95	0,89
ML Anomaly detection	System 2-b	System 2-a	NoOfCandidates	0,90	0,80	**1,00**	**1,00**	**0,84**
			Coordinates	0,58	0,16	**1,00**	0,98	0,54
			TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,50	0,00	**1,00**	0,00	0,50
			NoOfCandidates, Coordinates	0,90	**0,81**	**1,00**	**1,00**	**0,84**
			NoOfCandidates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,90	0,80	**1,00**	**1,00**	**0,84**
			Coordinates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	0,58	0,16	0,99	0,94	0,54
			NoOfCandidates, Coordinates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd	**0,91**	**0,81**	**1,00**	**1,00**	**0,84**

a performance below the 50% of accuracy for the system conf.1 and of 58% for the system conf.2-b. However, for the system conf.2-a, it performed extremely good. Nevertheless, when testing on system 2-a, the prediction accuracy is very poor, therefore this feature is not very trustworthy. As a matter of fact, in Figure 3 it is possible to see that even though the system's configuration remains equal, the meteorological conditions can radically change its spatial information, and hence the invalid markers can appear in different coordinates.

Certain combination of the features was only possible using ML techniques. In comparison, for the signature-based approach the problem of correlating the features was too complex to make it manually, and the results by using only the PLC features (*TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd*) without the number of marker candidates proved to be very inaccurate. Adding spatial information (i.e *Coordinates* feature) did not improved the results neither, due to the limitations explained before. But combining *NoOfCandidates, Coordinates, Displacement, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd* features, on the case of the erratic behaviour of system conf.1, proved to be the most accurate strategy. As for the system conf.2, *NoOfCandidates, TrolleyPos, Hoistpos, TrSpd, GaSpd, MhSpd* features donate the best and more reliable results, as they were constantly good in the two performed tests.

VI. Conclusion

This paper presents the preliminary results of the application of monitoring techniques to an industrial system: ABB Load Position System (LPS). The proposed monitoring solutions have shown to increase security and safety by corroborating PLC results, based on two main strategies: signature-based monitoring and ML-based anomaly detection techniques (i.e. neural networks).

The use of signature-based analysis showed its limitation in the context of a complex system where the thresholds of different features are not specified. The accuracy did not exceed 73%. The usage of ML algorithms improved this accuracy, Multi-layer Perceptron (MLP) classifier showed excellent results if the Crane LPS system is under the same configuration for the training and testing datasets. As expected we also noticed that the most features we have, the better accuracy we have.

Although the results obtained are promising, we plan to continue the application of MMT modules presented in this paper in order to improve the accuracy and precision of our analysis by using other ML algorithms and relying on more features. Finally, we plan to study the feasibility and performance of using both monitoring approaches combined, in an industrial context. ML results could be used as a first indication that a potential security breach has occurred, and signature-based monitoring could be used to rule out false positives and further identify the type of breach.

Acknowledgement

This work was made possible with funding from the European Union's Horizon 2020 research and innovation programme, under grant agreement No. 957212 (VeriDevOps). The opinions expressed and arguments employed herein do not necessarily reflect the official views of the funding body.

References

[1] D. Prince, "Cybersecurity: The security and protection challenges of our digital world," pp. 16–19, 2018.
[2] P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*. Springer, Jul. 2017.
[3] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.
[4] "Common Vulnerability Scoring System Version 3.0 Calculator," https://www.first.org/cvss/calculator/3.0.
[5] "Kaspersky ARC ICS 2020 Trend Report," https:ics-cert.kaspersky.commedia/Kaspersky_ARC_ICS-2020-Trend-Report.pdf, accessed: 2022-01-13.
[6] C. Thompson and D. Wagner, "A Large-Scale study of modern code review and security in open source projects," 2017.
[7] X. Franch and A. Susi, "Risk assessment in open source systems," 2016.
[8] A. Salamai, O. Hussain, and M. Saberi, "Decision support system for risk assessment using fuzzy inference in supply chain big data," 2019.

[9] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems," *Appl. Soft Comput.*, vol. 92, p. 106301, 2020.
[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," 2019.
[11] D. J. Richey, *Leveraging PLC Ladder Logic for Signature Based IDS Rule Generation*. Mississippi State University, 2016.
[12] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," 2020.
[13] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," 2017.
[14] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, Mar. 2019.
[15] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, Mar. 2019.
[16] B. Wehbi, E. Montes de Oca, and M. Bourdelles, "Events-based security monitoring using mmt tool," in *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, 2012, pp. 860–863.
[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.220.4598.671
[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533–536, 1986.
[19] J. Heaton, *Introduction to Neural Networks for Java, 2nd Edition*, 2nd ed. Heaton Research, Inc., 2008.
[20] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017.
[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.