University of Kent

Kent Academic Repository

Moura, Ralf Luis de, Franqueira, Virginia N. L. and Pessin, Gustavo (2021) ***Towards Safer Industrial Serial Networks: An Expert System Framework for Anomaly Detection.*** **In: Proceedings: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence ICTAI 2021. . IEEE**

Downloaded from

https://kar.kent.ac.uk/90285/ The University of Kent's Academic Repository KAR

The version of record is available from

https://doi.org/10.1109/ICTAI52525.2021.00189

This document version

Author's Accepted Manuscript

DOI for this version

Licence for this version

UNSPECIFIED

Additional information

Conference: https://ictai.computer.org/Item has to be updated when publication details become available.

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Towards Safer Industrial Serial Networks: An Expert System Framework for Anomaly Detection

***Abstract*—Cyber security is a topic of increasing relevance in relation to industrial networks. The higher intensity and intelligent use of data pushed by smart technology (Industry 4.0) together with an augmented integration between the operational technology (production) and the information technology (business) parts of the network have considerably raised the level of vulnerabilities. On the other hand, many industrial facilities still use serial networks as underlying communication system, and they are notoriously limited from a cyber security perspective since protection mechanisms available for TCP/IP communication do not apply. Therefore, an attacker gaining access to a serial network can easily control the industrial components, potentially causing catastrophic incidents, jeopardizing assets and human lives. This study proposes a framework to act as an anomaly detection system (ADS) for industrial serial networks. It has three ingredients: an unsupervised K-means component to analyse message content, a knowledge-based expert system component to analyse message metadata, and a voting process to generate alerts for security incidents, anomalous states, and faults. The framework was evaluated using the Profibus-DP, a network simulator which implements a serial bus system. Results for the simulated traffic were promising: 99.90% for accuracy, 99,64% for precision, and 99.28% for F1-Score. They indicate feasibility of the framework applied to serial-based industrial networks.**

***Index Terms*—Cyber security, Anomaly Detection Systems, Industrial Serial-based networks, Expert Systems, Profibus-DP**

I. Introduction

Cyber security is an increasing topic in industrial researches due to the intense, intelligent use of the data generated in many varied sources, which brings, as a consequence, greater integration between the communication elements. This phenomenon is also happening in the industry shop-floor through data generated on the Industrial Control Systems (ICS) [1]. ICSs are being connected to business networks to enable intelligent decision-making through the correlation among operational data extracted from plants in real-time and corporate or public databases. Industrial networks are how data generated in ICS can be integrated and consumed [2].

Industrial plants typically use industrial networks to integrate ICS elements. The up-to-date ones are based on ethernet-like networks, also known as routable industrial networks. These networks have cyber security protection mechanisms inherited from business networks such as identity and access control mechanisms, firewalls, IDS (Intrusion Detection Systems), and IPS (Intrusion Protect Systems) [3]. However, not all industrial networks are routable, and many industrial facilities make use of serial networks. Serial-based networks do not have similar mechanisms as routable networks and are very limited from a cyber security perspective. The limitations come from the fact that these networks were created when ICSs were isolated. There was no concern about security issues at that time, as an attacker would have to be physically connected to the network to carry out an attack [4].

Serial industrial networks were created to be deterministic, repeatable, and performing appropriately for their application. There are dozens, perhaps hundreds of different serial networks in industrial plants, the vast majority of them without any mechanism to protect against cyber-attacks [4]. Even so, these networks are being integrated into business networks, being considered essential sources of process data [3]. The integration increases vulnerabilities and opportunities for attacks. Because they do not have adequate security mechanisms, an attacker can easily control the network and cause catastrophic incidents [4]. In addition, critical infrastructures such as power systems, nuclear power plants, and even autonomous vehicles could be susceptible to these invaders, jeopardizing assets and human lives [1].

This study presents an expert system that combines knowledge-based and unsupervised technics to improve protection in industrial serial-based networks. ADS is a protection mechanism widely used in routable networks but with little or no application in serial-based networks [5]. The generic framework proposed is the base of an expert system that combines a knowledge-based process in parallel with unsupervised technics to detect anomalies. Furthermore, a voting process is employed to reduce false-positive rates and improve anomaly detection precision. The main objective of the expert system is to improve the identification of anomaly states, faults in network operation, and cyber security incidents. The main contributions of the proposed approach are: (1) We propose and evaluate a generic framework based on an expert system implemented in any industrial serial-based network as additional cyber security protection; (2) We propose an approach using knowledge-based combined with unsupervised technics to detect anomalies in industrial serial-based networks; (3) We present an efficient voting process to reduce false-positive rates and improve anomaly detection in industrial serial-based networks.

II. Serial-Based Industrial Networks and Cyber security Limitations

Industrial networks connect equipment in the industry to control and monitor physical actions and conditions. Many devices and equipment in an ICS are commonly segregated in layers that form different automation subsystems [40].

Distinct networks may support each subsystem according to its characteristics, such as the number of elements, real-time responses, throughput, and type of devices [4].

Serial-based industrial networks are usually deterministic and repeatable with regular cycles. Determinism means that there is a known maximum time in which data is sent and reaches its destination. Repeatable means that the intervals between sending the data are repeated in consecutive cycles. There can be no drop in performance or interruptions in these networks, as this would directly impact production processes [6]. Serial-based industrial networks commonly support lower-level network layers such as sensor networks (Sensorbuses), device networks (Devicebuses), and other field instruments (FieldBuses). The equipment can range from simple digital or analogic sensors plugged into an I/O module to intelligent devices, controllers, and PLCs (Programmable Logic Controller) [7]. Thus, there is a most suitable network for each type of equipment.

Serial-based networks were built on the assumption that all entities operating in the network are legitimately installed, perform the intended logic, and follow the protocol's rules [41]. In general, serial-based networks do not implement robust cyber security defense mechanisms. Cyber security professionals usually implement defense layers through firewalls between security zones and DMZs (Demilitarized Zones) to avoid access to the lower industrial networks. However, they do not have specific protection mechanisms for serial-based industrial networks, not even those from default [3].

The devices applied in serial-based networks, in general, do not implement cryptography; all the data transferred over the network is plaintext [41] because of hardware and network bandwidth limitations [9]. Network components do not verify the identity and permissions of other components associated with them [41], authentication and authorization are not usual, and the control is limited to the address checks. Network components do not verify the messages' content and legitimacy (Data Integrity) [41].

Updates in hardware and firmware are sometimes nonviable – some manufactures do not even support upgrades in cyber security features [10]. In addition, almost no devices or software specialized in serial-based industrial networks on the market, not even asset monitoring. ADS, IDS, or IPS are only available for routable networks, significantly reducing the possibilities for improvements in cyber security in serial-based networks [3]. If attackers could inject messages into the serial-based network, they could cause significant damage [41]. As there are usually no intrinsic cyber security mechanisms, network monitoring using an ADS seems to be an additional security measure. A critical aspect is that it is not intrusive; i.e., it does not interfere with the network.

III. Related works

Several defense techniques may be applied in industrial networks. Anomaly and intrusion detection systems are recognized as strong lines of defense, mainly in situations in which there are no other efficient cyber security mechanisms [11].

Intrusion detection is a process of detecting and tracking anomalous activity in computing and network resources [12]. IDS are based on the fact that an intruder's behavior will be noticeably different from that of a legitimate user [13]. Differently, anomaly detection methods assume that something abnormal is suspicious and tracks behavior, learning from continuous monitoring and data collecting [12]. Anomalies are patterns in data not conform to a notion of normal behavior [8]. IDS and ADS may be applied to improve cyber security defenses; however, ADS can detect additional events like other anomalous states and faults in the network operation.

There are several studies related to these subjects. Rubio et al. [11] review the threats that affect elements in an ICS and analyzed the applicability of IDS mechanisms. Chandola et al. [14] provides an overview of the research of anomaly detection systems, creating categories, and discussing the computational complexity of the anomaly detection techniques. Hu et al. [15] presented a taxonomy of IDS for ICS based on different methods and analyzed the advantages and disadvantages of various categories of IDS. Finally, Mitchell and Chen [16] classify IDS techniques based on detection technique and audit material, summarizing the benefits and drawbacks of each one.

Some industrial networks were covered in many studies that propose multiple techniques more adherents with their specificities. For example, Goldenberg and Wool [17] suggested deploying an intrusion detection system on Modbus /TCP networks based on deterministic finite automation (DFA), assuming that the traffic is highly periodic. Gao and Morris et al. [18] discuss the need for intrusion and detection systems for some vulnerabilities in the serial-based network Modbus RTU / ASCII. Liang et al. [19] proposed an intrusion detection using a multi-feature data clustering optimization model to improve detection accuracy and reduce false positive detection rates. Khan et al. [20] proposed a hybrid model that uses k-nearest neighbor and takes advantage of the consistent nature of communications patterns in SCADA (Supervisory Control and Data Acquisition Systems) networks.

Morris et al. [22] propose a deterministic intrusion detection for Modbus protocols (TCP and serial) derived from vulnerabilities analysis of these protocols. There are many ways to implement an anomaly or intrusion detection system. Multiple algorithms to be applied in industrial networks were proposed over the years. For example, Tomlin et al. [22] proposed an unsupervised learning algorithm based on clustering in SCADA system networks. Machine learning is the most popular technique in recent studies, as in Zolanvari et al. [23], Mehmod and Rais [24] and, Anton et al. [25]. Deep learning algorithms are also applied in anomaly and intrusion detection systems for industrial networks such as Hijazi et al. [26] and Javaid et al. [27]. Martinez et al. [28] and Alvaro and Emilio [29] proposed using multi-agent systems for intrusion detection that implement agents to collect and analyze information from multiple sources, not only from the industrial networks. This approach is considered a hybrid intrusion detection system such as in Shen et al. [30], Ullah and Mahmoud [31], and Anton et al. [32], in which various

techniques are applied to improve the robustness and the effectiveness of the ADS and IDS.

The studies that relate ADS or IDS to industrial networks in their majority are concentrated in techniques and algorithms for detecting anomalies or intrusions in SCADA networks based on routable protocols and the traditional methods already applied in business networks. Some studies propose solutions for networks based on serial protocols but are limited to specific protocols, like Modbus and CAN [9]. Although these works deal with serial networks, they are not generic to the point of being implemented in other well-known protocols and used in industry such as Profibus DP / PA, Interbus, DeviceNet, among others. Thus, there is a notorious gap that needs to be addressed in networks based on serial protocols with the primary objective of strengthening the cyber security defenses in ICS where these networks are still applied.

IV. PROFIBUS-DP

Profibus-DP is a serial-based network, and it implements a serial bus system. Profibus is a multi-master system and offers different services for automation technology as cyclic data exchange for process data and acyclic data exchange for configuration and diagnostics [33]. There are two types of devices; the masters are an active station that determines data traffic on the bus and can assume two functions (or classes) [34]: DP master, class 1: Control the system and the slaves, typically are controllers, PLC, or computer-based systems. DP master, class 2: These masters are tools for commissioning, engineering, and maintenance. Typically, they are PC based systems.

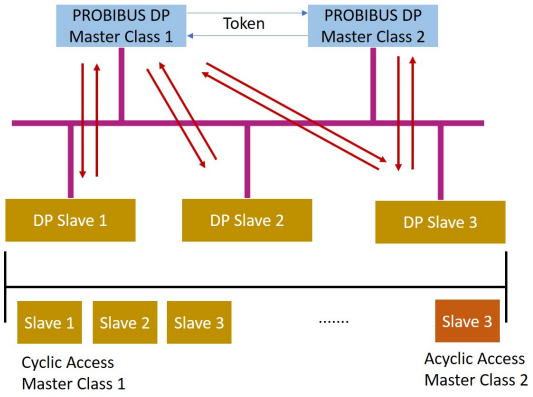

Fig. 1. Structure of Profibus-DP system

Each Profibus-DP device must have a unique address for communication; these addresses comprise the range 0-127 [35]. Profibus-DP has well-defined cycle phases, as shown in Figure 2. During the startup phase, addresses are checked to assess if they are valid, and slave diagnostics, configuration, and parametrization are carried out. After all validations, the data exchange starts [33].

During the data exchange phase, master class 1 exchanges messages with slaves cyclically. The message exchange routine made by master class 1 includes requesting information for

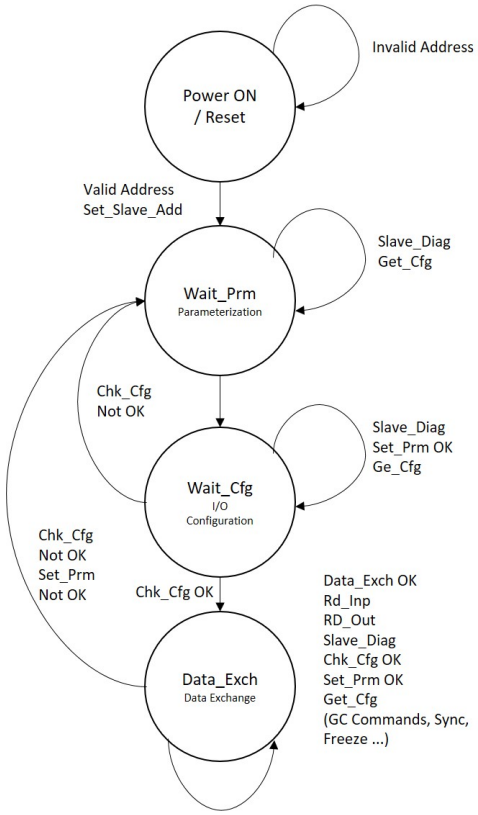

Fig. 2. Cycle phases [4]

slaves who are part of the network and assessing whether new slaves have been included. The network master exchange occurs cyclically between class 1 and class 2 masters; however, Class 2 masters do not follow a routine, as their action occurs in periods of configuration and diagnosis, which can vary according to the need [33].

Telegrams and telegram sequences are uniform, and their formats are fixed. There are five distinct telegram formats, as shown in Figure 3 [33]:

- SD1: Telegram without data field.
- SD2: Telegram with variable length (payload range from 4 to 249 bytes).
- SD3: Telegram with fixed data length (8 bytes).
- SD4: Token telegram.
- SC: Short confirmation.

SD means, start delimiter, and assume different values as shown in Figure 3. ED stands for end delimiter and marks the end of a telegram [35].

The message sequence during the data exchanges is: a master receives the token from another master (SD4), sends a telegram without a data field (SD1), and gets a short confirmation (SC) [33]. Then, the master sends another telegram without a data field (SD1) and receives a telegram with variable length in reply (SD2). Finally, it sends this telegram to all active slaves. After the cycle, the master passes on control and sends a token to another master (SD4) [35].

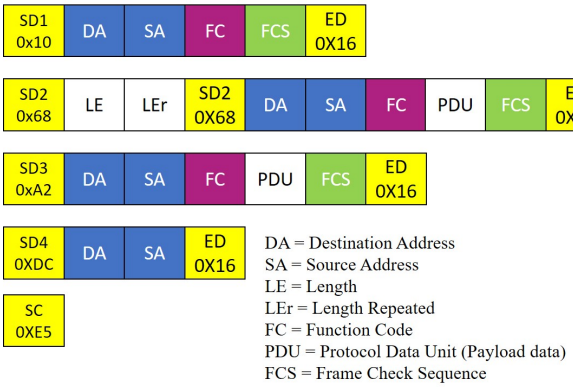

Fig. 3. Telegram formats [2]

V. Expert System Proposal

The Expert System is based on an important observation: Serial-based network traffic is highly periodic [6], [41]; if multiple traffic samplings are regularly performed, it tends to converge in a pattern in the long term. The regular pattern helps establish a normal behavior, and any deviation from this pattern signals an anomaly. The anomaly can represent a cyber security event or an issue in the typical network behavior (a defect, for example).

Serial-based network traffic behavior reduces the anomaly detection complexity because it can sense the minimum difference from the expected behavior. In this situation, a low false-positive rate technique called Knowledge-based is indicated to detect many types of attacks, predominately those find out through rules [36]. The knowledge-based technique implemented to the Profibus-DP network can be called an expert system [14].

However, these techniques have difficulties detecting new types of attacks because they only evaluate specific rules [14], [36]. Furthermore, these techniques have problems detecting attacks in the message data (PDU), as the message data can vary, which unfeasible evaluations through directives.

Unsupervised techniques can circumvent this problem; they may detect new types of attacks in the message data (PDU) and eliminate the need for known directives. Another advantage of these technics is that they do not need labeled training data, simplifying the model refining [14]. However, they have high false-positive rates. Even an efficient classifier may not be sufficiently discriminative and generates false positives despite robust training [37].

The expert system is based on the framework shown in Figure 4, which proposes composing a knowledge-based with unsupervised methods in a voting process. The composition aims to increase the scope of anomaly detection and reduce the false-positive rates. A voting processor evaluates (1) an anomaly in metadata, (2) or anomaly in the message data detected by the unsupervised algorithm; however, in this case, variations in the message data statistics should also occur.

The framework divides the ADS into two phases: insight momentum (or training phase) when the models are created and refined and running momentum when the detection actions occur.

A. Insight Momentum

In the insight momentum, the pre-processed data collected by a sniffer is stored at the Training Dataset. The Training Module transforms the stored data into a list of function codes, addresses used during the multiple cycles, and a traffic behavior model, as shown in Figure 5. The insight phase needs a dataset large enough to see reliable behavior patterns.

The lists and the model are the base to detect anomalies in two aspects: data behavior and message behavior.

1) Data Behavior: During the data exchange phase, the master requests slaves' information about their monitored physical variables or sends updates to their actuators' behavior. Profibus-DP expresses regular cyclical behavior, and the data in a stable operation in the long term tends to converge to a particular pattern that can be the reference of normal behavior. The network cycle arrangement expresses two data behavior clusters, as the network cycle has two primary cycles – Master Class 1 data exchange and configuration/parametrization phases (Figure 2).

The SD2 and SD3 telegrams are the data sources of this model. During the pre-processing, the PDUs are extracted and stored in the Training Database. The data extracted from PDUs can be clustered in different groups that represent a specific data behavior. The cluster method groups object into meaningful subclasses so that the members from the same cluster are similar, and the members from different groups are quite different from each other [38].

K-means is an unsupervised learning clustering technique that has shown promising anomaly detection techniques [38], [39]. The K-means algorithm creates two clusters representing the Profibus-DP primary cycles; the clusters are extracted with their centroids (centers of the clusters) referenced for Euclidean distance assessments in the running momentum.

2) Message Behavior: The message behavior focuses on the metadata. The metadata is extracted from the messages exchanged during all cycles. There is three message behavior evaluation:

- Traffic behavior: Refers to traffic statistics collected during the cycles. The statistics are master length data average, slave length data average, and message size average. The traffic behavior evaluates all cycles (defined by the SD4 telegram) and all telegrams.
- Function Codes: Collect all function codes exchanged between masters and slaves. The function code occurs in SD1, SD2, and SD3 telegrams.
- Address list: Collect all active addresses in the network. The tuple source target is also collected and stored. The addresses are collected in SD1, SD2, SD3, and SD4 telegrams.

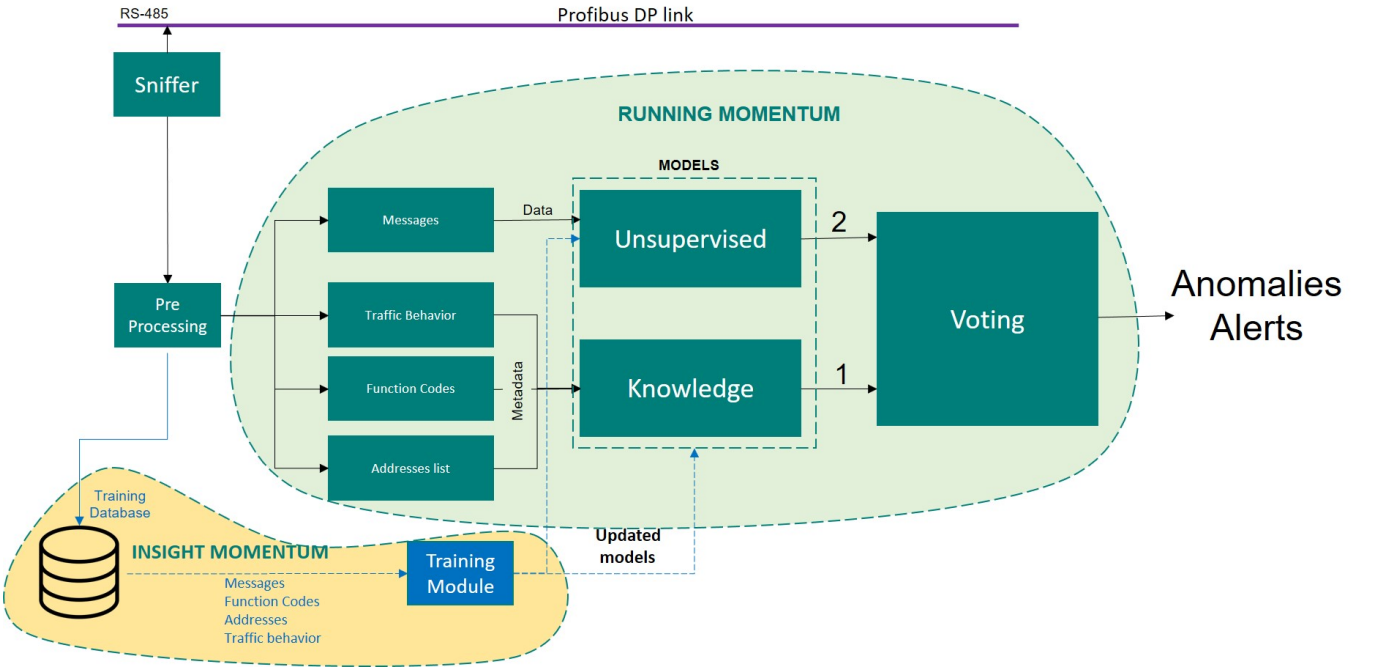

Fig. 4. Proposed Framework

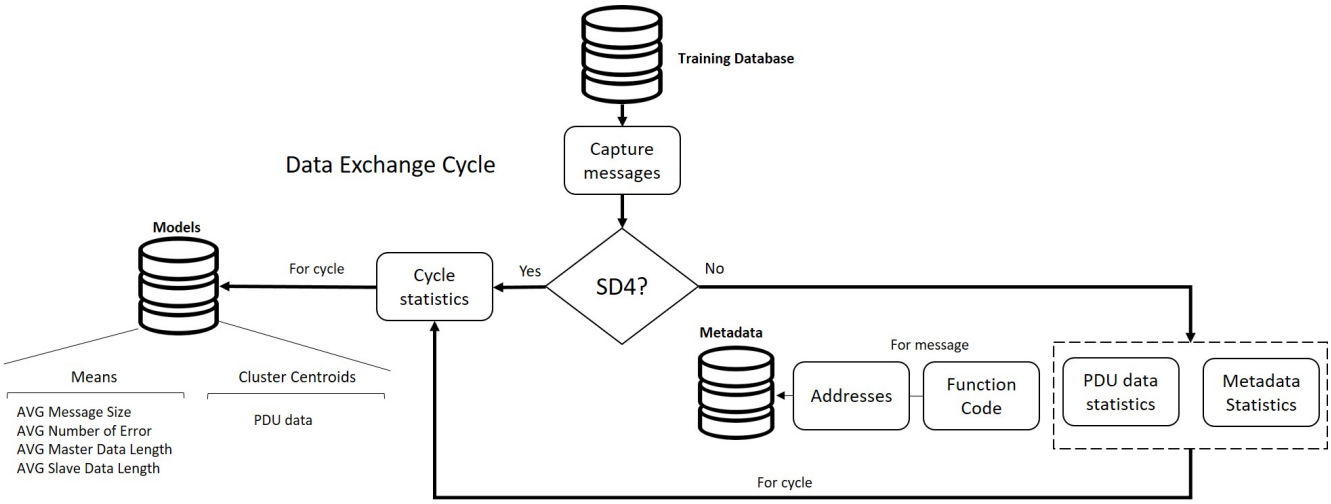

Fig. 5. Insight momentum flow.

B. Running Momentum

In the running momentum, the real-time data collected by the sniffer (Figure 4) is pre-processed to detach PDU data and metadata from the messages. Figure 6 shown the running momentum flow. The SD4 message delimit the network cycle [33]. Some evaluations are done for each message and others for each cycle. As such, PDU data statistics are calculated for every cycle, and the Euclidean distance and metadata are evaluated for every message.

1) Voting Method: Voting is a method that combines multiple detection techniques with the aim of reach better results. Different techniques can detect anomalies with their particular false-positive and false-negative rates. When they are combined, it is possible to reach better rates [17]. The proposed framework applies a voting method to get a more accurate result than a single technique [17]. Expert systems based on knowledge have low false-positive rates but can have high false-negative rates when exposed to new cyber-attacks [14]. On the other hand, the unsupervised technique is suitable to detect new attacks, but it has high false-positive rates [37]. Combining these two techniques in a voting process can improve anomaly detection reducing false-positives and false-negative rates. Algorithm 1 shows the background logic in the voting process.

Table I shows the anomaly detection techniques applied in the framework. For PDU Data, the average of master data,

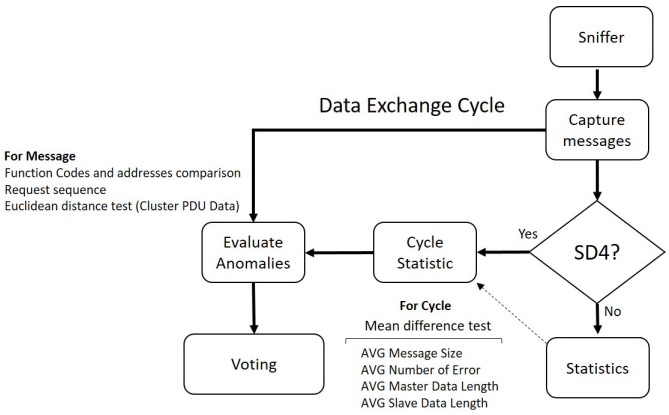

Fig. 6. Running momentum flow.

Algorithm 1: Voting algorithm for evaluate a cycle anomaly alert

function Voting $(alerts)$;
Input : An ordered alert array of the 9 (Table I) different anomalies $alert$
Output: Boolean$Anomaly$
if $countPDUDataAnomaly(alerts) \geq 2$ *or* *evaluateMetaDataAnomaly(alerts)* **then**
    return $True$*;*
    {*If there is more than one alert related to the PDU data or if there is at minimum one metadata alert.*}
else
    return $False$*;*

slave data, number of errors, and the message size is calculated in every cycle. This information is compared with the pattern through a difference in means (confidence level of 0.95), but no alert is issued unless another data deviation is detected. Additionally, the Euclidean distance evaluates data deviation comparing the distance between each message PDU and the cluster centroids (0.80 of percentile). Still, no alert is issued unless another data deviation is detected.

Figure 7 shown the voting process; each technique has the same probability of detecting an anomaly. Still, the alert is only issued if two or more techniques detect an abnormal behavior simultaneously. The techniques are described in Table I.

For message metadata, the knowledge model compares, in real-time, the function codes and the addresses with the pattern stored in the metadata database. Additionally, a tuple source-target is also compared to evaluate new communication flows. For message metadata, any deviation issue an alert due to is an apparent deviation of abnormal behavior. Any modification in the network should trigger a model retraining.

Another traffic behavior assessment is the class 1 master request message sequence (Table 2 – 1.4). There is a sequence of requests between masters and slaves (starting from 1 to 127) [33]. If the request series is interrupted, an anomaly alert

TABLE I
ANOMALY DETECTION TECHNIQUES

Method	Technique	Anomaly
1. Knowledge	1.1 Mean Difference	1.1.1 Master average PDU data deviation
1. Knowledge	1.1 Mean Difference	1.1.2 Slave average PDU data deviation
1. Knowledge	1.1 Mean Difference	1.1.3 PDU data average size deviation
1. Knowledge	1.2 Comparison	1.2.1 New Function Code (metadata)
1. Knowledge	1.3 Comparison	1.3.1 New Addresses (metadata)
1. Knowledge	1.3 Comparison	1.3.2 New Tuple (metadata)
1. Knowledge	1.4 Rule	1.4.1 Master requests sequence out of order (metadata)
1. Knowledge	1.5 Rule	1.5.1 Number of messages per cycle (metadata)
2. Unsupervised	2.1 Euclidean distance	2.1.1 PDU Data deviation

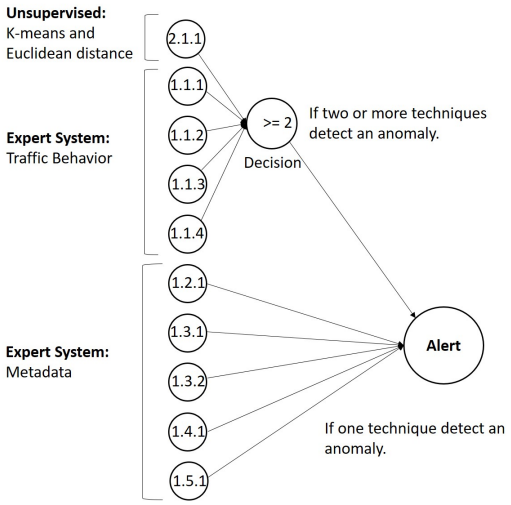

Fig. 7. Voting Process.

is issued.

VI. EXPERIMENT SETUP

The algorithms and techniques were implemented using the python language. The phyton code in insight momentum collects statistical data and rule-based data by storing it in a local database. At running momentum, the code counts the anomalous states identifying them for future analysis. The experiments were performed with data collected from a simulation software called "Profibus Network Simulator" [42]. The simulation software very accurately reproduces the normal functioning of a Profibus-DP network.

A real Profibus-DP network with a class 1 master and five slaves was the basis for experimenting using the Profibus-DP simulator.

In addition, one hundred thousand data records with normal behavior were collected, and these were used for the insight

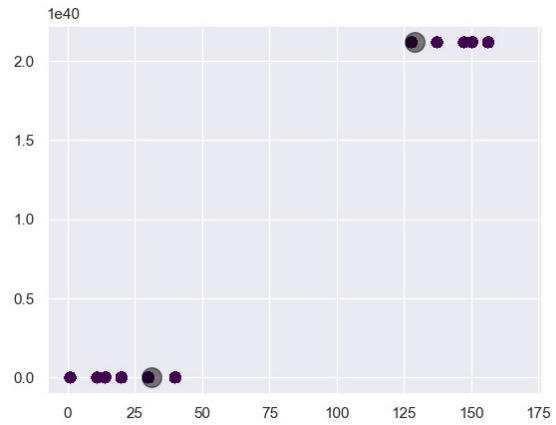

Fig. 8. Clusters centroids.

momentum for training. After the startup phase, the communication establishes a very repeatable routine. Since these networks are very repeatable [6], a smaller amount of data would be enough to model normal behavior. Table II shows the statistics calculated during the training phase.

TABLE II
STATISTICS

Statistic	Value
Master data length average (bytes)	6.99
Master data length standard deviation (bytes)	0.24
Slave data length average (bytes)	31.34
Slave data length standard deviation (bytes)	7.55
Message size average (bytes)	27.23
Message size standard deviation (bytes)	12.45
Message cycle count average (bytes)	25.98
Centroid X cluster 1 (Data Deviation)	30.999
Centroid Y cluster 1 (Data Deviation)	-9.769e40
Centroid X cluster 2 (Data Deviation)	129
Centroid Y cluster 2 (Data Deviation)	2.118e40

These statistics are used at running momentum as factors to compare traffic behavior—the Figure 8 shown the cluster centroids represented by gray circles and the observations by purple circles.

VII. Experimental Results

A second base was collected with 20,000 records from the network simulator for the running momentum. To simulate attacks or anomalies, new slaves were added via software; slaves were removed from the network, the master address was modified, repeated messages were inserted out of sequence. In addition, messages with errors or data different from the normal data were included in different moments.A total of 1398 abnormal situations were included to be detected by the algorithms. Figure 9 shown Profibus-DP telegram fields that were changed (in red) to simulate anomalies.

The abnormal states were identified, counted to enable the metrics calculations. An overview of the performance of the algorithms on the dataset is provided in Table III.

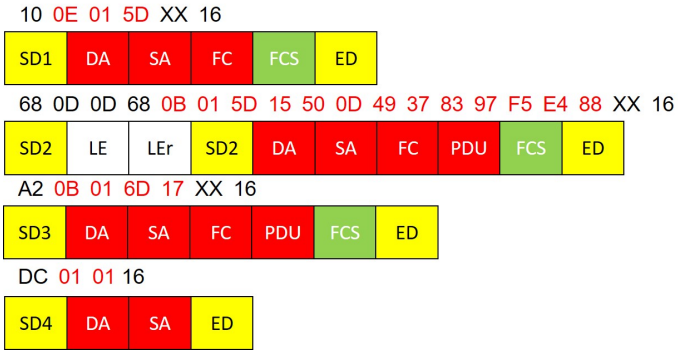

Fig. 9. Profibus-DP anomalies.

The overall accuracy and F1-Score were 99.90% and 99.28%, respectively. The high rates demonstrate that highly cyclic and repeatable networks make anomaly detection simpler.

VIII. Evaluation

Accuracy (A) is a ratio of correctly predicted observations to total observations. Total observations are the addition of True positives (TP), True Negatives (TN), False positives (FP), and False Negatives (FN), as shown in Equation 1. As shown in Table III, the voting process has a smaller accuracy which implies higher false-negative rates if compared with individual unsupervised techniques. Precision (P) (Equation 4) is the ratio of correctly predicted positive observations to the total predicted positive observations. The precision after the voting process is higher than the individual techniques, which implies fewer false positives. The recall (Equation 3) is the ratio of correctly predicted positive observations to all observations. Recall is the ability to find all the positive samples. F1 Score (F1)(Equation 4) is the weighted average of Precision and Recall (R) [25]. This score takes both false positives and false negatives into account. It is observed that after the voting process, a better F1 Score is obtained than the other individual detections.

$$A = \frac{TP + TN}{TP + FN + FP + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{4}$$

Analyzing the results, it is clear that, individually, the number of false positives is higher than after the voting process, which somewhat reduces the problem of unsupervised algorithms. However, this may imply an increase in the rate of false negatives, as it takes two detections to confirm an alert. As the accuracy of rule-based algorithms is high, this, in a

TABLE III
PERFORMANCE

Method	TP	FN	FP	TN	Accuracy	Precision	Recall	F1 Score
1.1.1 Master average data deviation	95	5	3	19897	0.9996	0.9694	0.9500	0.9596
1.1.2 Slave average data deviation	95	5	2	19898	**0.9997**	0.9794	0.9500	0.9645
1.1.3 Message average size deviation	197	3	3	19797	**0.9997**	0.9608	0.9800	0.9703
1.2.1 New Function Code	190	0	0	19810	1.000	1.000	1.000	1.000
1.3.1 New Addresses	120	0	0	19880	1.000	1.000	1.000	1.000
1.3.2 New Tuple	120	0	0	19880	1.000	1.000	1.000	1.000
1.4.1 Master request sequence out of order	160	0	0	19840	1.000	1.000	1.000	1.000
1.5.1 Number of messages per cycle	168	0	0	19832	1.000	1.000	1.000	1.000
2.2.1 Data deviation	130	10	5	19855	0.9993	0.9630	0.9286	0.9455
Voting (overall)	1373	15	5	18597	0.9990	**0.9964**	**0.9892**	**0.9928**

way, creates a balance that allows for both high accuracy and an added ability to detect new attacks.

Rules-based detection is 100% accurate because all deviations were detected. It was an expected behavior since rule-based algorithms do not rely on statistical inference to detect anomalies. However, it can be a problem in networks where constant changes happen. In these cases, new training must be carried out to update the model at every change in the network. Therefore, this type of detection only makes sense in stable networks, a situation expected in industrial networks.

Another important point to consider is the need to interrupt detection during network maintenance, diagnostics, configuration, and parameterization procedures. During these periods, the network behavior will be different, which can cause undue alerts. As these actions are always controlled, it will not be challenging to interrupt detection during these periods.

The experiment carried out with simulation software is an intermediate test built to attest to using ADS in industrial serial networks. The results were auspicious, which now allows us to proceed to tests in real networks.

IX. CONCLUSION

The framework application and experiment were made in a Profibus-DP network; however, there is no limitation in implementing it in any other serial-based industrial network. The framework is generic, and the same assumptions applied in Profibus-DP can be adapted to other networks taking into account the specificities of each network. The advantage of using anomaly detection techniques is that it is possible to detect cyber security events, other anomalies such as device defects or communication bus problems.

Applying two anomaly detection techniques in the context of the industrial network reduces the number of false positives. In addition, it increases the evaluation scope, allowing anomaly detection both in the context of messages (metadata) and data.

This work analyzed datasets captured from a Profibus-DP simulated tool to evaluate the voting framework proposed. The results showed 99.90% accuracy and 99.28% of F1-Score, which implies a promising approach and shows the advantages of combining the techniques.

The voting process reduces the number of false positives, but it increases the possibility of false negatives. However, part of knowledge-based algorithms does not participate in the voting process, creating a balance that turns the framework robust enough to practical implementation. Thus, it is attested that the feasibility of applying ADS in serial industrial networks.

A limitation of this work is that it uses data from the simulation. Therefore, the conclusions are still initial, done only to test the framework's viability. Future work may be carried out on real data collected in industrial networks. Real-time tests are also needed to verify the performance and algorithms response time and the possibility of implementation on dedicated hardware.

REFERENCES

[1] Moura, R. L.; Gonzalez, A.; Franqueira, V. N. L.; Neto, A. L. M.; Pessin, G. Geographically Dispersed Supply Chains: A Strategy to Manage Cybersecurity in *Industrial Networks Integration*. Chapter 6. In K. Daimi; C. Peoples (eds) Advances in Cybersecurity Management, Springer Nature, Switzerland, 2021.

[2] Moura, R. L.; Ceotto, L. L. F.; Gonzalez, A. Industrial IoT and Advanced Analytics Framework: an approach for the Mining Industry.*In 2017 International Conference on Computational Science and Computational Intelligence*. pp.1308-1314, 2017. DOI 10.1109/CSCI.2017.228.

[3] E. D. Knapp, *Industrial Network Security*. Syngress, 2011.

[4] L. C. Branquinho, Marcelo A.and Moraes, J. Seidl, J. A. Junior, and B. Branquinho, Thiago, *Segurança de Automação Industrial e SCADA*. 1ed., Campus, 2014.

[5] S. Liyakkathali, F. Furtado, G. Sugumar, and A. Mathur, Validating anomaly detection mechanisms in industrial control systems,in Proceedings of *TMCE 2020. Organizing Committee of TMCE 2020*, 2020, Conference Proceedings, pp. 89–102.

[6] A. B. Lugli and M. M. D. Santos, *Redes industriais para automação industrial - As-I, Profibus e Profinet*, 2nd ed. Erica, 2018.

[7] K.-H. Cho, B.-H. Kim, and K.-S. Park, Case study on rate-based traffic control of industrial networks employing LonWorks, *International Journal of Systems Science*, vol. 33, no. 3, pp. 161–164, 2002.

[8] Chandola, V. Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Computing Surveys*, v (9), pp.1-72, 2009. DOI:10.1145/1541880.1541882.

[9] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, Conference Proceedings, pp. 63–68.

[10] H. Kim, "Security and vulnerability of SCADA systems over IP-based wireless sensor networks." *International Journal of Distributed Sensor Networks*, vol. 8, no. 11, p. 268478, 2012.

[11] Rubio, J. E.; Alcaraz, C.; Roman, R.; Lopez, J. Analysis of Intrusion Detection Systems in Industrial Ecosystems. In Proceedings of the *14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 116-128, 2017. DOI: 10.5220/0006426301160128.

[12] D. Yang, A. Usynin, and J. W. Hines, Anomaly-based intrusion detection for SCADA systems, in *5th intl. topical meeting on nuclear plant instrumentation, control, and human-machine interface technologies*, 2006, Conference Proceedings, pp. 12–16.

[13] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, Network intrusion detection, *IEEE Network*, vol. 8, no. 3, pp. 26–41, 1994.

[14] Chandola, V.; Bannerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Computing Surveys*, v (9), pp.1-72, 2009.

[15] Hu, Y. Li, H.; Sun, Y.; Sun, L. A survey of intrusion detection on industrial control systems. *International Journal of Distributed Sensor Networks*, v (14), n (8), pp. 1-14, 2018. DOI: 10.1177/1550147718794615.

[16] Mitchell, R.; Chen, I. A Survey of Intrusion Detection Techniques for Cyber-Physical Systems. *ACM Comput. Surv.* v (46), n(4), pp. 1-29, 2014. DOI: http://dx.doi.org/10.1145/2542049.

[17] M. Raihan-Al-Masud and H. A. Mustafa, Network Intrusion Detection System Using Voting Ensemble Machine Learning, *2019 IEEE International Conference on Telecommunications and Photonics (ICTP)*, 2019, pp. 1-4, DOI: 10.1109/ICTP48844.2019.9041736.

[18] Morris, T.; Vaughn, R. Dandass, Y. A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems. In: *2012 45th Hawaii International Conference on System Sciences*, 2012. p. 2338-2345.

[19] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, An industrial network intrusion detection algorithm based on multi-feature data clustering optimization model, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[20] Khan, I. A.; PI, D.; Khan, Z. U.; Hussain, Y.; Nawaz, A. HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems. *IEEE Access*, vol.7, pp. 89507-89521, 2019. DOI: 10.1109/ACCESS.2019.2925838.

[21] Morris. T. H., J. B. A. and Vaughn R. B., and D. Y. S., Deterministic intrusion detection rules for Modbus protocols. inIn: *2013 46th Hawaii International Conference on System Sciences*., 2013, Conference Proceedings, pp. 1773–1781.

[22] Tomlin Jr., L.; Farnam, M. R. A clustering approach to industrial network intrusion detection. In: *Proceedings of the 2016 Information Security Research and Education (INSuRE) Conference (INSuRECon-16)* . 2016.

[23] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, Machine learning-based network vulnerability analysis of industrial internet of things, *IEEE Internet of Things Journal*, vol. 6, no.4, pp.6822–6834, 2019.

[24] Mehmod and H. B. M. Rais,Ant Colony Optimization and Feature Selection for Intrusion Detection. Lecture Notes in *Electrical Engineering*, 2016, book section Chapter 27, pp. 305–312.

[25] S. D. Anton, L. Ahrens, D. Fraunholz, and H. D. Schotten, Time is of the essence: Machine learning-based intrusion detection in industrialtime series data, in *2018 IEEE International Conference on DataMining Workshops (ICDMW)*. IEEE, 2018, Conference Proceedings, pp.1–6.

[26] A. Hijazi, A. El Safadi, and J.-M. Flaus, A deep learning approach for intrusion detection system in industry network, in *BDCS Intell*, 2018, Conference Proceedings, pp. 55–62.

[27] Javaid, Q. Niyaz, W. Sun, and M. Alam, A deep learning approach for network intrusion detection system, in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, Conference Proceedings, pp. 21–26.

[28] Martinez, C. V.; Sollfrank, M.; Vogel-Heuser, B. A Multi-Agent Approach for Hybrid Intrusion Detection in Industrial Networks: Design and Implementation. In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN).* IEEE, 2019. p. 351-357.

[29] H. Alvaro and C. Emilio, *Multiagent systems for network intrusion detection: A review*. Springer, 2009, pp. 143–154.

[30] C. Shen, C. Liu, H. Tan, Z. Wang, D. Xu, and X. Su, Hybrid-augmented device fingerprinting for intrusion detection in industrial control system networks, *IEEE Wireless Communications*, vol. 25, no. 6, pp. 26–31, 2018.

[31] Ullah, I.; Mahmoud, Q. H. A Hybrid Model for Anomaly-based Intrusion Detection in SCADA Networks. In: 2017 IEEE International Conference on Big Data (BIGDATA), 2017.

[32] Anton, S. D. D.; Sinha, S.; Schotten, H. D. Anomaly-based intrusion detection in industrial data with SVM and random forest.In: *27th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, IEEE, 2019.

[33] Felser, M. *PROFIBUS Manual*. Ed.1.2.2, ePubli GmbH: Berlin, 2012.

[34] PI. *Profibus System Description*. Profibus & Profine International, 2016.

[35] Acromag. *Introduction to Profibus DP Technical Reference*, Acromag: Wixom, USA, 2002.

[36] E. J. M. Colbert and S. Hutchinson, *Intrusion Detection in Industrial Control Systems, ser. Advances in Information Security*. Springer, 2016, book section Chapter 11, pp. 209–237.

[37] Balntas, V.; Tang, L.; Mikolajczyk, K. Improving object tracking with voting from false-positive detections. In *22nd International Conference on Pattern Recognition*, pp. 1928-1933, 2014.

[38] Jianliang, M.; Haikun, S.; Ling, B. The application on Intrusion Detection Based on K-means cluster algorithm. In *2009 International Forum on Information Technology and Applications*. pp.150-152, 2009.

[39] Feng, C., Li, T., & Chana, D. (2017, June). Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 261-272, 2017.

[40] S. Liyakkathali, F. Furtado, G. Sugumar, and A. Mathur, Validating anomaly detection mechanisms in industrial control systems, in P*roceedings of TMCE 2020. Organizing Committee of TMCE 2020*, 2020, Conference Proceedings, pp. 89–102.

[41] Goldenberg, N., & Wool, A. (2013). Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *International journal of critical infrastructure protection*, 6(2), 63-75.

[42] Veiga, R.; Brandão, Dennis. *Simulador Redes Profibus*. Escola de Engenharia de São Carlos, Universidade de São Paulo, Dissertação de Mestrado, 2013.