Non-IP Industrial Networks: An Agnostic Anomaly Detection System

Ralf Luis de Moura*, Virginia N. L. Franqueira and Gustavo Pessin*****

*Vale S.A., Vitória-ES, Brazil (e-mail: ralf.moura@vale.com)
**University of Kent, School of Computing, Canterbury, UK (e-mail: v.franqueira@kent.ac.uk)
*** Instituto Tecnológico Vale, Ouro Preto-MG, Brazil, (e-mail: gustavo.pessin@itv.org)

Abstract: This paper describes a system to detect anomalies in non-IP (Internet Protocol) industrial networks on Industrial Control Systems (ICS). Non-IP industrial networks are widely applied in ICS to connect sensors and actuators to control systems or business networks. They were designed to be in an air-gapped security environment and therefore contain almost no cyber security features and are vulnerable to various attacks. Even though they are part of the communication layers, a few external cyber security controls are applied in this crucial tier. As an extension of the work by De Moura et al. (2021), this study proposes and tests the proof-of-concept of an agnostic anomaly detection system (AADS) to detect anomalies on any non-IP industrial network (e.g., DeviceNet, CANBus) as an additional cyber security measure working at the physical network layer. The proof-of-concept is comprised of three modules, including hardware and software components: data gathering (sniffer), parser, and detection. Testing the proof-of-concept in an industrial lab network (i.e., a Profibus-DP lab network) showed the proposal's feasibility with a detection rate above 99% (overall accuracy: 99.59%; F1-Score: 99.18%).

Resumo: Este artigo descreve um sistema para detectar anomalias em redes industriais não IP (*Internet Protocol)* em Sistemas de Controle Industrial (ICS). As redes industriais não IP são amplamente aplicadas em ICS para conectar sensores e atuadores a sistemas de controle ou a redes de negócios. Eles foram projetados para estar em um ambiente de segurança isolado e, portanto, quase não contêm recursos de segurança cibernética e são vulneráveis a vários ataques. Embora façam parte das camadas de comunicação, poucos controles externos de segurança cibernética são aplicados nessa camada crucial. Como extensão do trabalho de De Moura et al. (2021), este estudo propõe e testa a prova de conceito de um sistema agnóstico de detecção de anomalias (AADS) para detectar anomalias em qualquer rede industrial não IP (por exemplo, DeviceNet, CANBus) como uma medida adicional de segurança cibernética trabalhando no camada física de rede. A prova de conceito é composta por três módulos, incluindo componentes de hardware e software: coletor de dados (sniffer), analisador (dados) e detecção. O teste da prova de conceito em uma rede industrial de laboratório (ou seja, uma rede de laboratório Profibus-DP) mostrou a viabilidade da proposta com uma taxa de detecção acima de 99% (precisão geral: 99,59%; F1-Score: 99,18%).

Keywords: Anomaly Detection Systems, Non-IP Industrial Networks, Cyber Security, Industrial Control Systems

Palavras-chaves: Sistemas de Detecção de Anomalias, Redes Industriais Não-IP, Segurança Cibernética, Redes Industriais

1. INTRODUCTION

Increasingly, data from the factory floor is used in critical infrastructure environments. This need derives from the escalated use of advanced data analytics at Information Technology (IT) levels boosted by the advent of Industry 4.0. The data is typically generated on Industrial Control Systems (ICS) hosted on automation levels supported by industrial networks and protected from business networks (De Moura et al., 2021).

Industrial networks are the way how IT systems can consume data from ICS. The traditional approach connects network layers through gateways, DMZs (Demilitarized Zones), firewalls among business security zones, and automation layers that allow real-time data exchange. Gateways may transfer and translate network variables and messages until they reach business networks (Fiedler et al., 2000; Cheminod et al., 2012).

Industrial networks apply to many industries and are extensively used in critical infrastructures that usually pose risks for assets that control, e.g., nuclear, electric, and chemical facilities, which, when attacked, may endanger people's lives. This fact corroborates a growing concern with the cyber security risks of these environments (De Moura et al., 2020).

In the past, ICSs were isolated from other environments. However, today, in the highly interconnected era, there is a substantial increase in cyber security risks centered on threats operating in different modes in the industry (Cazorla et al., 2016; Rubio et al., 2017). Recent cyber security events, such as Stuxnet, Duqu, Shamoon, and Gass (Branquinho et al., 2014; Gao and Morris, 2014; Hemsley and Fisher, 2018), demonstrated that industrial malware could potentially impact ICS.

Non-IP industrial networks (NIPIN) are a type of industrial network still very used for ICS and have poor cyber security mechanisms (Hijazi et al., 2018). They were created with the assumption of isolation (Ullah and Mahmoud, 2017) when the main desirable characteristics were determinism and performance. NIPIN are over 20 years old (Templeton, 2010) and typically cannot undergo security updates due to limited equipment resources and low transmission capacity (Luigi and Santos, 2018).

Thus, since NIPIN are being integrated and do not have adequate security mechanisms, a relevant cyber security gap needs to be addressed by strengthening cyber security defenses. Due to limited capabilities, a viable non-intrusive option would be to concentrate on detecting cyber-attacks rather than on prevention. A significant effort focuses on anomaly detection solutions (Yang et al., 2006; Krotofil and Gollmann, 2013; Zhou and Gu,2018; Yang et al., 2019; Martinez et al., 2019); however, previous solutions rarely address non-IP network cases.

In this work, we focus on an agnostic anomaly detection approach. We attempt to detect abnormal network behavior at the physical level in a non-intrusive manner. Our proposal is based on the premise that attacks at the physical network level have the ultimate objective of altering equipment behavior on the shop floor. Anomaly detection can facilitate early detection of cyber-attacks and faulty equipment, which ultimately represent potentially exploitable vulnerabilities.

Therefore, the contribution of this paper is a proof-of-concept for an agnostic anomaly detection system that detects anomalies related to cyber security at the physical level in any non-IP network. Tests in a lab network (NIPIN) have shown the feasibility of the proposed system.

2. NON-IP INDUSTRIAL NETWORKS AND CYBER SECURITY ISSUES

Technological evolution led mainly by the advent of digital electronics and later digital communication allowed the emergence and proliferation of digital industrial networks. Industrial networks result from the digitization of automation equipment initially supported through pneumatic and analog controllers (Galloway and Hancke, 2012).

Networks began to be applied due to the evolution of information technology, communication networks in companies, and the growing need for integration between production and corporate systems (De Moura et al., 2021; Knapp, 2015). However, industrial networks have not reached de-facto worldwide standards, unlike business networks, due to the numerous automation equipment manufacturers and the market competition. There are dozens, maybe hundreds, of different industrial networks.

Industrial networks can be defined as mission-critical distributed systems applied in industrial areas to enable communication between intelligent devices present in ICS (Knapp, 2015). Industrial networks are typically used to integrate devices in a particular automation subsystem responsible for part of the production process in industries. The primary objective is to increase efficiency and reduce production costs (Knapp, 2015; Galloway and Hancke, 2012).

The production process goes through several steps performed by different elements present in the industrial environment. The trend in the industrial environment is to have several subsystems with a certain autonomy, with each one being responsible for parts of the production process. These subsystems are autonomous, i.e., horizontally independent, but vertically integrated into communication silos. Vertical integration happens through integration between industrial networks and systems that allow, for example, data collected on the factory floor to be forwarded to corporate systems (De Moura et al., 2021a; Galloway and Hancke, 2012). In general, there is no standard for the network hierarchy, meaning that the subsystems can be supported through the composition of a diversity of industrial networks.

Compared to business networks, the primary difference is that industrial networks are connected to equipment and control and monitor physical actions and conditions. In general, some other differences are: architecture – industrial networks have a much deeper architecture with different protocols; failure severity – failure of the system has a much more significant impact such as production loss, human safety, and damage to equipment; real-time – the response time should be less than the sample time of data gathering; determinism – the transmission must also be done in a predictable or determinist fashion; and periodic and aperiodic traffic - industrial networks require the transmission of both periodically sampled data and aperiodic events such as change of state or alarm conditions (Galloway and Hancke, 2012).

NIPIN use the serial bus and point-to-point communication links that interconnect devices designed for remote command and control (Knapp, 2015), such as Profibus DP, Modbus/RTU, and DNP3. This kind of network is usually cyclic, deterministic, and repeatable with regular cycles and predictable traffic (Luigi and Santos, 2018).

The cyber security community is broadly unaware of the prevalent use of non-IP-based communication; consequently, most cyber security mechanisms are based on TCP/IP networks (Templeton, 2010). NIPINs were built assuming that all entities operating in the network were legitimately installed, performed the intended logic, and followed the protocol's rules (Goldenberg and Wool, 2013). In general, NIPIN do not implement robust cyber security defense mechanisms; they do not enforce cryptography; therefore, all the data is transferred over the network in plaintext

(Goldenberg and Wool, 2013) because of hardware and network bandwidth limitations (Song et al., 2016).

NIPIN's cyber security limitations derive from these networks being created when ICS was isolated. There was no concern about cyber security, as an attacker would have to be physically connected to the network to carry out an attack (Knapp, 2015; De Moura et al., 2021). Network components do not verify the identity and permissions of other associated components, authentication and authorization are not usual, and the control is limited to IP address checks (Goldenberg and Wool, 2013; De Moura et al., 2021).

The focus on cybersecurity-related modernization tends to overlook these systems, particularly security monitoring and attack detection (Templeton, 2010). Updates in NIPIN devices' hardware and firmware are often non-viable; some manufacturers do not even support upgrades of existing cyber security features (Kim, 2012). Furthermore, these networks remain intact for many years because they have planned operational lives exceeding 20 years, and the existing technology works well. Given the size of many critical infrastructure operations, exchanging or upgrading technology is not feasible until a strong need arises (Templeton, 2010).

The integration and modernization processes expose ICS to untrusted networks, which increases vulnerabilities; even so, the danger of threats is still underestimated (Gollmann, 2011). Recent reports of intrusion in industrial networks show that relying solely on perimeter protection and network segmentation is poor (ICS-CERT, 2022), leading to the urgent need to create mechanisms capable of increasing the security level of these networks.

3. ANOMALY DETECTION SYSTEMS

Anomaly detection systems (ADS) detect and track anomalous activities in computing and network resources (Yang et al., 2006). ADS are based on the hypothesis that an attacker's behavior will be noticeably different from that of a legitimate user (Mukherjee et al., 1994). Anomaly detection methods assume that everything abnormal is suspicious by tracking behavior and learning from continuous monitoring and data collection (Yang et al., 2006). Thus, anomalies are patterns in data that do not conform to a notion of normal behavior (Chandola et al., 2009).

ADS have requirements that should be delivered to ensure their correct and reliable operation (Martinez et al., 2019):

- Data trustworthiness: The data they collect must be trustworthy and protected against tampering.
- Interoperability: They must interoperate in a non-intrusive way with other components (industrial equipment, for example).
- Flexibility and scalability: They must be adaptable and extensible as needed.
- Robustness: They are also susceptible to attacks, so they must be resilient.
- Completeness: They must detect all types of anomalies (i.e., exploits, misuses, and intrusions).
- Up-to-date: They must be updatable to detect new threats.
- Configurability: They must allow adjustments according to specific environmental settings.
- Real-time: They must promptly detect and respond to anomalies.

ADS act when something out of the ordinary happens. In industrial networks, variations in behavior should be minimal due to their repeatable and predictable characteristics (De Moura et al., 2021; Branquinho et al., 2014). Clearly defining what is "normal" is critical to implementing successful ADS. The definition may be derived from rules and policies to establish baselines of normal behavior. These policies may address various behaviors; therefore, exceptions to these rules would show suspicious activities. The level of policies could encompass network traffic patterns, user access, and operation control (Chandola et al., 2009).

For NIPIN, an anomaly detection method could be built on a deterministic model and applied to analyze execution procedures of protocols, the pattern of communications, and states of operations to detect causes that result in deviations (Zhou and Guo, 2018). A model can be established for these networks based on the following behavior indicators (Knapp, 2015; Tomlin et al., 2016).

- Network traffic: Set of unique devices' IP addresses, traffic volume, and flow duration.
- Process/control behavior: Set of unique function codes, setpoints, or configuration changes.
- Event/incident activity: Set of expected events by criticality.
- Devices on the network: New physical (MAC) addresses appear in the network.
- Sensors and actuators: Operational process variations, abnormal function, or unexpected network operations.

The literature addresses several methodologies to detect anomalies. They are mainly based on the knowledge of the dynamics of data exchange and the protocol functions for attack detection. The described anomaly detection methods are, in some cases, quite effective. Regular communications can be mapped, and events such as out-of-sequence messages can be detected, in which cases the detection rate is around 99% (Goldenberg and Wool, 2013).

In general, detection systems use detection algorithms classified as signature-based, statistical-based, knowledge-based, anomaly-based, and machine-learning-based. Knowledge-based and signature-based techniques perform well over highly on periodic and predictable network behavior (Krotofil and Gollmann, 2013). However, they are limited to unknown attacks based on known standards (Gao and Morris, 2014). Statistical or anomaly-based techniques increase the unknown attack detection rate (Colbert and Hutchinson, 2016). Supervised learning techniques are also commonly used for anomaly detection in IP networks and have good acceptance among scholars (Ullah and Mahmoud, 2017; Yang et al., 2019; Hijazi et al., 2018; Javaid et al., 2016, Anton et al., 2018).

4. PROPOSED AGNOSTIC ANOMALY DETECTION SYSTEM

We propose to detect anomalies in the physical network, the last communication layer between the control systems and the final equipment. Anomaly detection applied to the physical network layer allows the detection of different attacks since the attacker has the ultimate objective of acting directly on the equipment behavior connected by the network. Communication will always occur over the physical network. Even if malware invades a programmable logic controller (e.g., Stuxnet, Black Energy, Crashoverrride), the supervisory system (e.g., Shamoon, Notpetya, triton) or a direct compromise of any network components, it will be possible to observe abnormal behavior at the physical layer.

The agnostic anomaly detection system (AADS) proposed defines hardware and software applicable in non-IP networks, provided the necessary adaptations are enforced. They will be required at the physical layer level, using the appropriate transceiver, parser modifications, and training of the provided models.

AADS is comprised of three modules: *Data Gathering*, *Parser*, and *Detection*. Each module is responsible for a part of the process, which is implemented in hardware and software. Figure 1 shows a high-level block diagram with the interaction among modules.

The Data Gathering module connects with the serial-based industrial network operating as a sniffer without interfering with the data traffic. Non-interference is particularly important because industrial networks are sensible, and any data deviation can disrupt communication. The function of the Data Gathering module is to transform electrical signals into a data stream that specific algorithms can process.

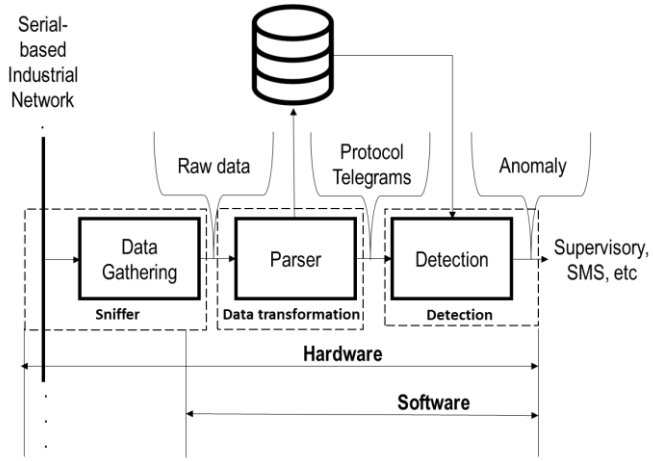

Fig. 1 – Proposed Agnostic Anomaly Detection System

The Parser module transforms the streaming into protocol telegrams and separates the data set into frames according to the protocol specification. The telegram is the data unit processed to evaluate the anomaly detection process. The Parser module is also responsible for collecting historical data used by the AADS to train and improve the models.

The Detection module has two different execution times. Training time uses historical data to create/improve its internal models. Moreover, it analyses protocol telegrams in real-time, searching for abnormal behaviors representing a cyber-attack or a network fault. After detection, an event is created to inform the network operators as soon as an anomaly is detected. The Detection module is also responsible for informing the anomaly using, for example, REST API or another type of message.

4.1 AADS Hardware

The block diagram in Figure 2 presents the minimum components that compose the hardware solution.

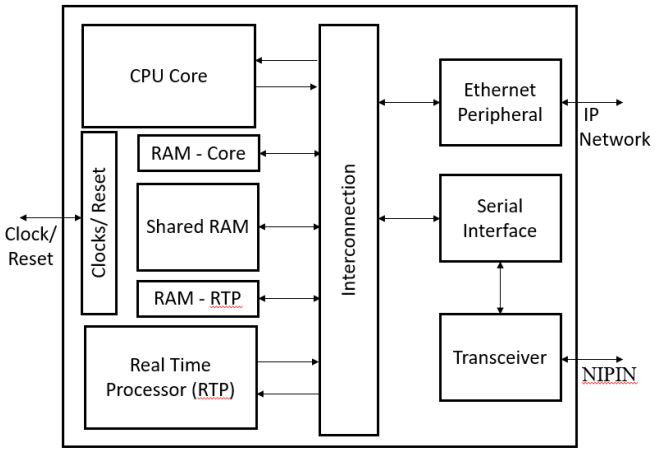

Fig. 2 – Hardware high-level block diagram of the Agnostic Anomaly Detection System

Each hardware component has an essential and indispensable function for the anomaly detection system:

- The transceiver connects to the NIPIN links.
- The Ethernet peripheral communicates with the IP network (i.e., business network).
- The real-time processor (RTP) analyses the data stream and telegrams.
- The CPU (Central Processing Unit) core processes the telegrams and dispatches events.
- The shared memory enables communication between the real-time processor and the CPU core.
- The interconnection bus integrates all components.

The components are applied in parts of the proposed AADS process with a specific function detailed in the following sections.

4.2 Data Gathering Module (*Hardware and Software*)

The Data Gathering module has three components: *Transceiver*, *Serial Interface*, and *Firmware*, as shown in Figure 3. The function of this Module is to convert electrical signals into a data stream. The Transceiver module is part of the hardware interfaces that should be specified according to the physical layer of the industrial network. Some industrial network physical layer standards are applied in many industrial networks, such as RS-485, RS-232, and IEC 1158-2 (Lugli and Santos, 2018), and can share the same

Transceiver module. However, it should be defined according to the electrical signals and voltage levels. It should be galvanically isolated and adapted for asynchronous data transmissions. The transceiver protects the electronic circuit from transients and static discharges during device or equipment handling with low-energy but high-voltage transients (Texas Instruments, 2015).

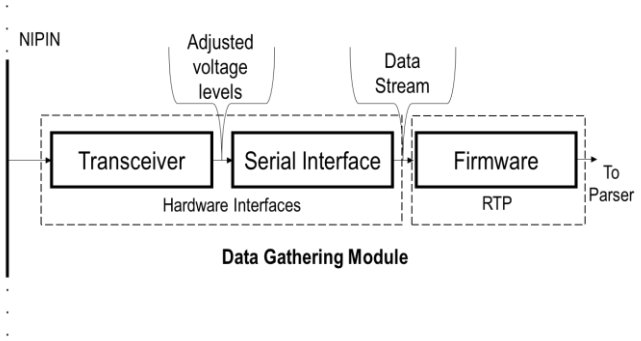

Fig. 3 –Data Gathering Module

The transceiver uses the serial interface component to establish connections, and serial ports can communicate with the RTP, which receives a data stream and sends telegrams to the Parser module.

The firmware software component executes into the RTP. It is developed in low-level languages like Assembly or C and is responsible for converting electrical signals into data streams. The firmware must run in a fast processor to have adequate performance.

4.3 Parser Module (*Software*)

The Parser module is responsible for converting a data stream into protocol frames (telegrams). It should be designed to identify the start and end message characters and separate them according to the protocol rules. Figure 4 shows that each telegram is sent to the detection module and stored in the local database.

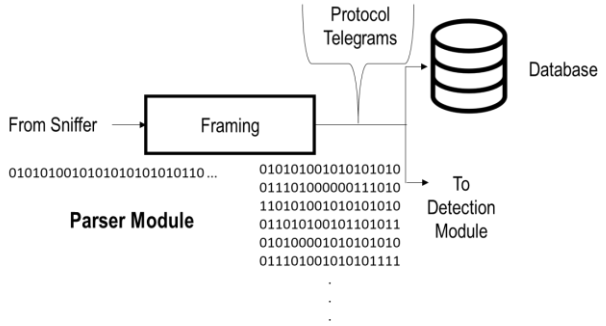

Fig. 4 –Parser Module

The local database is a repository of historical data that can be used to refine and improve the AADS model continuously.

4.4 Detection Module (*Software*)

The anomaly detection module is responsible for detecting anomalies in the industrial network physical layer. Its algorithms run over the CPU core and can be developed in high-level languages like Java and Python. The anomaly detection module is the solution's core and uses special algorithms to detect anomalies.

The NIPIN telegram comprises two main parts: Metadata and Message Data. The metadata includes source and targets IP addresses, commands, and protocol data control. The message data is the data collected from sensors or sent to the actuators. An anomaly detection system should consider the behavior of both parts of the telegram.

The predictable and repeatable nature of ICS traffic and relatively static network topology can be leveraged to detect anomalies, whereas known legitimate control sequences/codes and unsafe states make them suitable for several detection algorithms (Krotofil and Gollmann, 2013). This 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 detects many types of attacks, predominately those identified through rules (Colbert and Hutchinson, 2016; De Moura et al., 2021a).

However, knowledge-based techniques have difficulties detecting new types of attacks because they only evaluate specific rules (Colbert and Hutchinson, 2016). Furthermore, they have problems detecting attacks in the message data, as it can vary, making it unfeasible for evaluations through rules.

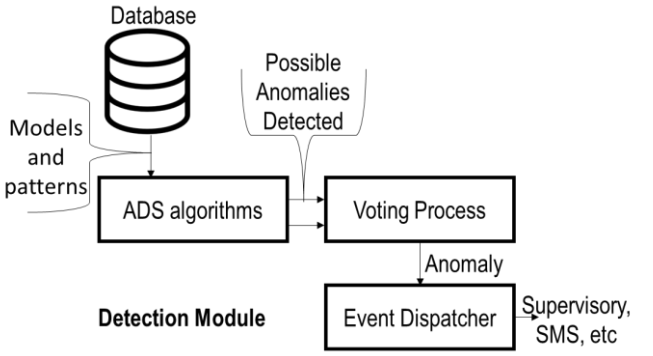

Fig. 5 –Detection Module (*Software*)

Unsupervised techniques can reduce this problem; they may detect new types of attacks in the message data and eliminate the need for rules. However, they have high false-positive rates. Even an efficient classifier may not be sufficiently discriminative and generate false positives despite robust training (De Moura et al., 2021).

The proposed detection module composes knowledge based on unsupervised techniques, as shown in Figure 5, in a voting process. The composition can increase the scope of anomaly detection and reduce the false-positive rate. A voting processor evaluates an anomaly in the metadata or an abnormality in the message data detected by the unsupervised algorithm; however, variations in the message data statistics should also occur in this case.

Knowledge-based Algorithm

The data collected by the Sniffer module is stored in the database. The detection algorithm transforms the stored data into a list of metadata such as function codes, IP addresses,

message data average size, and the number of messages per cycle. The metadata is used as a pattern compared during the NIPIN cycles. Any deviation from the pattern indicates an anomaly. Due to the regular cycles, anomaly detection should typically have a high rate.

Unsupervised technique

The data extracted from messages are clustered in different groups representing 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 dissimilar.
K-means is an unsupervised learning clustering technique that has shown promising results for anomaly detection (Jianliang et al., 2017; Feng et al., 2017; De Moura et al., 2021). The K-means algorithm creates n clusters representing NIPIN cycles; the clusters are extracted with their centroids (centers of the clusters) based on Euclidean distance assessments. The number of clusters depends on the message data behavior and should be evaluated during training. Figure 6 shows the complete software block diagram.

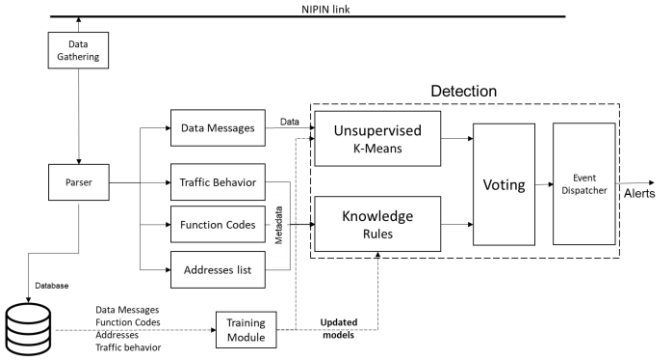

Fig. 6 – Software Block Diagram for the Agnostic Anomaly Detection System

5. OVERVIEW OF THE PROOF-OF-CONCEPT AND EVALUATION RESULTS

As a proof-of-concept, an experiment in a Profibus-DP network was set up in the laboratory using commercial components to create a similar environment. Figure 7 shows the proof-of-concept diagram. It uses a Beaglebone Black C and the transceiver ISO1176T. The Beaglebone Back C has two PRUs (Programmable Real-Time Units) and an ARM Processor AM3359 (Beaglebone, 2022).
The PRU runs the firmware code that collects the ISO1176T transceiver (Texas Instruments, 2015) data through the serial UAR (User Authorization Request) interface. The firmware is coded in C. The Parser module (coded in C++) runs over the Linux operating system and the ARM process that receives messages shared by the PRU. Communication between the PRU and ARM processors is handled by the R30 register from the PRU side and by the rpmsg_pru30 for PRU0 via Remote Procedure Message Framework (rpmsg driver), which uses a virtual I/O and virtual I/O device ring buffer construct. It is based on the Interrupt handling technique available in the BeagleBone Black (Beaglebone, 2022).

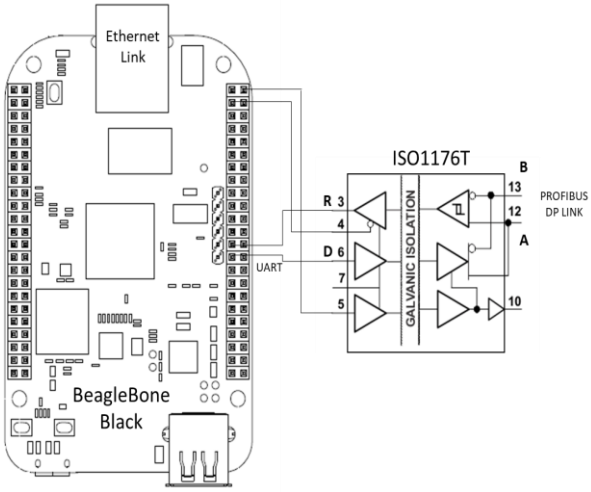

Fig. 7 – Proof-of-concept diagram for the Agnostic Anomaly Detection System

The ARM process receives the data stream and Parser module in individual telegrams stored in a local database and sends it to the anomaly detection module developed in Python. The detection module processes the information in real-time, generates an alert when detecting an anomaly, and sends it as a message through the Ethernet interface. The detection module has models previously trained with historical data (i.e., normal behavior) collected for a month.
This study extends the De Moura et al. (2021) study that tested these algorithms with software-simulated data. The proof-of-concept was inserted into a laboratory Profibus-DP network with one master and two device slaves, as shown in Figure 8. The data collected is real (not software-simulated) but comes from a laboratory network, not an operational plant network.

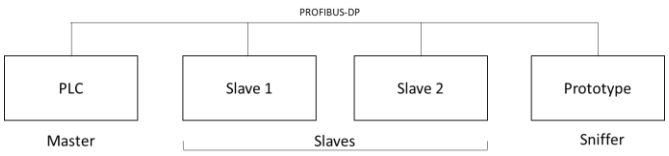

Fig. 8 – Laboratory Network

Data behavior is evaluated for data sent by masters, slaves, and all data (*Data, Master, and Slave average data deviation and Message average size deviation*). These data, over time, tend to converge to a particular pattern that can be referred to as normal behavior. The metadata summarizes the list of addresses used in the network, the list of functions, and the number of messages per cycle (*New Function Code, New Addresses, New Source-Target addresses, Master request sequence out of order, Number of messages per cycle*).
For testing purposes, four new nodes were physically added; nodes were removed from the network, and the master's address was modified. In addition, different commands were included at different moments to simulate various attacks. The testing window was 50000 messages. The abnormal states were identified and counted to enable the metrics calculations.
Metadata uses a rule-based detection that is 100% accurate because all deviations were detected. It was an expected

result since this algorithm does not rely on statistical inference but on strict rules.

Accuracy is a ratio of correctly predicted observations to total observations. The F1 Score is the weighted average of precision and recall; it considers both false positives and false negatives (De Moura et al., 2021). After the voting process (considering statistical inference – Data and Strict rules – Metadata), the results have shown a better F1 Score which denotes better performance when compared with each event individually with lower rates of false-positive and false-negative events.

The overall accuracy and F1-Score were 99.59% and 99.18%, as shown in Table 1 (more detail about the data analysis can be found in De Moura et al. (2021)), respectively revalidating the findings in De Moura et al. (2021) now with real network data.

Table 1. Proof-of-Concept results for AADS

Anomaly	Type	Accuracy	F1 Score
Master average data deviation	D	0.9980	0,9448
Slave average data deviation	D	0.9981	0,9690
Message average size deviation	D	0.9978	0,9780
New Function Code	M	1,0000	1,0000
New Addresses	M	1,0000	1,0000
New Source-Target addresses	M	1,0000	1,0000
Master request sequence out of order	M	1,0000	1,0000
Number of messages per cycle	M	1,0000	1,0000
Data deviation	D	0.9987	0,9451
Voting	**D/M**	**0.9959**	**0,9981**

D – Data; M – Metadata

The proof-of-concept response time measurements showed that anomalies were identified between 500 ms and 2 s after the occurrence, as shown in Figure 9.

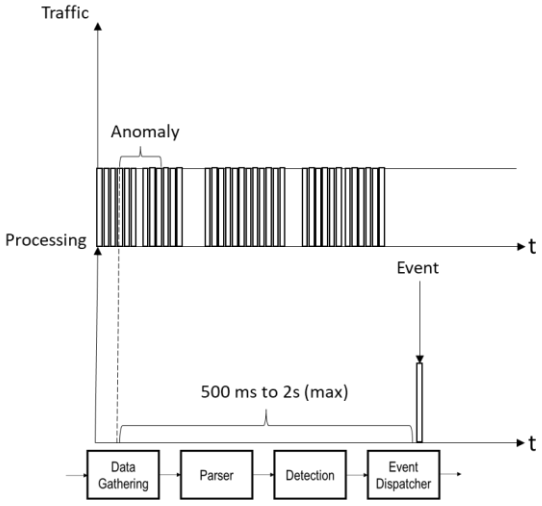

Fig. 9 – Response time for the Agnostic Anomaly Detection System

The results imply a promising approach and show the advantages of combining unsupervised and knowledge techniques with real data. The high rates demonstrate that highly cyclic and repeatable networks simplify anomaly detection, corroborating with the study by Goldenberg and Wool (2013).

6. CONCLUSIONS

This study proposed an agnostic solution to detect anomalies in non-IP industrial networks, aiming to promote the need to increase security protections they intrinsically lack. The proposed anomaly detection system needs to be agnostic because there are currently many non-IP industrial networks in the market, and a more comprehensive solution could have a greater reach. Although the proposal is agnostic, modifications must be made to adapt it to each industrial network; adaptations are required at the physical interfaces, for example, and the software layer, such as model retraining, according to the specificities of the network. The proof-of-concept shows the proposal's feasibility of achieving a high detection rate in cyclical and repeatable networks, as expected.

A limitation of this work is that it uses data from a laboratory. However, this work's objective was to achieve now that we have implemented an operational AADS that includes hardware and software and the possibility of using hardware to detect anomalies in non-IP networks.

Therefore, the initial conclusions were only to test the framework's feasibility. Future work may be carried out on data collected in the industrial networks from operational plants. Real-time tests are also needed to verify the performance and algorithms' response time and the possibility of implementation on dedicated hardware.

The proposed anomaly detection system works very well in static networks. If the topology, parameters, the number of nodes, or even the expected functions are changed, the behavior of the network will change, forcing it to retrain the models, which remains as future work. The AADS should also be turned off in a maintenance/parametrization period to avoid false alerts.

REFERENCES

Anton, S. D., Kanoor, S., Fraunholz, D., and Schotten, H. D. (2018). Evaluation of machine learning-based anomaly detection algorithms on an industrial Modbus/tcp data set., in: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. 1–9.

Beaglebone, (2022). *Beaglebone Black*, https://beagleboard.org/.

Branquinho, L. C., Moraes, M. A., J., and Seidl, J. A. J. B. B. (2014). *Segurança de Automação Industrial e SCADA*., 1st Edition, Campus.

Chandola, V., Banerjee, A., and Kuma, V. (2009). Anomaly detection: *A survey.*, *ACM computing surveys (CSUR)* 41 (3) 1–58.

Cheminod, M., Durante, L., and Valezano, A. (2012). Review of security issues in industrial networks, *IEEE transactions on industrial informatics*. 9 (1) 277–2935.

CISA, (2022). *ICS-CERT Advisories*, https://www.cisa.gov/uscert/ics/advisories

Colbert, E. J. M., and Hutchinson, S. (2016). Intrusion detection in industrial control systems., in: *Advances in Information Security*, 209–237.

De Moura, R. L., Gonzalez, A., Franqueira, V. N. L., and Neto, A. L. M. (2020). A cyber-security strategy for internationally-dispersed industrial networks, in: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, 62–68.

De Moura, R. L., Franqueira, V. N. L., and Pessin, G. (2021). Towards safer industrial serial networks: An expert system framework for anomaly detection, in: *IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021, 1197–1205.

De Moura, R. L., Gonzalez, A., Franqueira, V. N. L., and Neto, A. L. M., and Pessin, G. (2021a). Geographically dispersed supply chains: A strategy to manage cybersecurity in industrial networks integration., in: *Advances in Cybersecurity Management*, 97–116.

Feng, C., Li, T., Chana, D. (2017). Multi-level anomaly detection in industrial control systems via package signatures and lstm networks., in: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*., 261–272.

Fiedler, P., Bradac, Z., and Zezulka, F. (2000). New methods of interconnection of industrial fieldbuses, in*: IFAC Proceedings*, 145–147.

Galloway, B. and Hancke, G. P. (2012). Introduction to industrial control networks, *IEEE Communications surveys and tutorials* 15 (2) 860–880.

Gao, W., and Morris, T. (2014). On cyber-attacks and signature-based intrusion detection for modbus based industrial control system, *Journal of Digital Forensics, Security and Law*, 9 (1) 37–56.

Goldenberg, N., and Wool, A. (2013). Accurate modelling of modbus/tcp for intrusion detection in SCADA systems., *International Journal of Critical Infrastructure Protection* 6 (2) 63–75.

Gollmann, D. (2011). From insider threats to business processes that are secure-by-design., in: *INCoS*, 627. 17

Hemsley, K., Fisher, R. R. (2018). A history of cyber incidents and threats involving industrial control systems., in: *International Conference on Critical Infrastructure Protection*., 215–242.

Hijazi, A., El Safadi, A. and Flaus, J. (2018). A deep learning approach for intrusion detection system in industry network., in: *BDCSIntell*, 55–62.

Javaid, A., Niyaz, Q., Sun, W., and Alam, M. (2016). 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).*, 21–26.

Jianliang, M., S. Haikun, and Ling, B. (2009). The application on intrusion detection based on k-means cluster algorithm., in *International Forum on Information Technology and Applications*, 150–152.

Kim, H. (2012). Security and vulnerability of SCADA systems over IP-based wireless sensor networks., *International Journal of Distributed Sensor Networks* 8 (11) 268478.

Knapp, E. D. (20115). Industrial Network Security, 2nd Edition, Syngress.

Krotofil, M., and Gollmann, D. (2013). Industrial control systems security: What is happening?, in: 2013 11th *IEEE International Conference on Industrial Informatics (INDIN)*, 670–675.

Lugli, A. B., and Santo, M. M. D. (2018). *Redes industriais para automação industrial*, 2nd Edition, Erica.

Martinez, C. V., Sollfrank, M., and Vogel-Heuser, B. (2019). A multi-agent approach for hybrid intrusion detection in industrial networks: Design and implementation., in: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*., 351–357.

Mukherjee, B., Heberlein, L. T. and Levitt, K. N. (1994). Network intrusion detection., *IEEE Network* 8 (3) (1994) 26–41.

Rubio, J. E., Alcaraz, C., Roman, R., and Lopez, J. (2017). Analysis of intrusion detection systems in industrial ecosystems., in: *The 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017)*, 116–128.

Song, H. M., Kim, H. R., and Kim, K. (2016). 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)*, 2016, p. 63–68.

Templeton, S. (2020). Security monitoring and attack detection in non-ip based systems., in: *In International Conference on Cyber Warfare and Security*, 473.

Texas Instruments, (2015). *Iso1176 isolated RS-485 profibus transceiver*.

Tomlin, L., Farnam, M. R., and Pan, S. (2016). A clustering approach to industrial network intrusion detection., in: *Proceedings of the 2016 Information Security Research and Education (INSuRE)*.

Ullah, I., and Mahmoud, Q. H. (2017). A hybrid model for anomaly-based intrusion detection in SCADA networks, in: *2017 IEEE International Conference on Big Data (BIGDATA)*, 2160–2167.

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

Yang, H., Cheng, L., and Chuah, M. C. (2019). Deep-learning-based network intrusion detection for SCADA systems., in: *2019 IEEE Conference on Communications and Network Security (CNS)*, 1–7.

Zhou, L., Gu, H. (2018). Anomaly detection methods for IIoT networks., in: *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 214–219.