Edge Multi-agent Intrusion Detection System Architecture for IoT Devices with Cloud Continuum

Gustavo Funchal*, Tiago Pedrosa*†, Fernando de la Prieta‡, Paulo Leitao*†
* Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança,
Campus de Santa Apolónia, 5300-253 Bragança, Portugal,
Email: {gustavofunchal, pedrosa, pleitao}@ipb.pt
† Laboratório para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança,
Campus de Santa Apolónia, 5300-253 Bragança, Portugal
‡ BISITE Digital Innovation hub, University of Salamanca, Edificio I+D+i, C/ Espejos s/n, 37007, Salamanca, Spain,
Email: fer@usal.es

2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS) | 979-8-3503-6301-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICPS59941.2024.10639952

***Abstract*—The Industry 4.0 has brought significant changes in production processes and business models worldwide. Advanced technologies, e.g., Collaborative Robotics, Artificial Intelligence, Cloud Computing, and Internet of Things (IoT) are playing a crucial role in improving efficiency and productivity. However, the adoption of these technologies, particularly IoT, introduces security vulnerabilities and potential attacks due to inadequate security measures. This paper addresses the need for dedicated cybersecurity mechanisms and secure device design in IoT networks, particularly emphasizing the challenges faced in implementing Intrusion Detection Systems (IDS) on resource-constrained IoT edge devices, limiting the use of traditional machine learning based detection methods. Moreover, the limited computational resources of IoT devices require lightweight techniques that have low power requirements but can accurately detect anomalies in the network. To tackle these challenges, a novel multi-agent based architecture is proposed, considering the distribution of nodes along the edge-cloud continuum, and enabling the collaboration among different processes to detect anomalies during attacks. The proposed architecture is evaluated at the edge level using the CICIoT2023 dataset. The results demonstrate the feasibility of using multi-agent systems for a collaborative detection of IoT attacks, contributing to enhance the security of IoT-based systems against cyber threats in Industry 4.0 environments by leveraging lightweight techniques.**

***Index Terms*—Intrusion Detection Systems, Multi-agent Systems, Internet of Things, Machine Learning.**

I. Introduction

The fourth industrial revolution, also known as Industry 4.0 [1], is changing the production and business models around the world, improving processes and considerably increasing productivity, through the use of advanced technologies such as Artificial Intelligence (AI), Cloud Computing and Internet of Things (IoT). Among these technologies, the IoT consists in the interconnection between objects through the Internet to share data, i.e. objects have the ability to communicate and interact, and can be monitored and controlled remotely.

The use of these technologies to digitize the production processes (sensing, acquiring, transmitting and processing data) aims to optimize the processes, but open doors to vulnerabilities and attacks in these networks of devices. These heterogeneous networks are prone to a variety of attacks, including data leakage, spoofing, denial of service (DoS/DDoS) and energy bleeding, among others, due to the poor security measures and absence of specialized anomaly detection systems for them [2].

According to [3], 53% of organizations abandoned new business projects due to an inability to address cybersecurity risks, 74% faced a situation where there was a lack of an appropriate security solution, and 52% are worried about collecting Big data from IoT devices due to the risk of sabotage and espionage, but 64% of businesses already maintain or use IoT solutions. The same study refers that 43% of surveyed businesses indicated that at least one type of IoT was not protected, highlighting a clear need for dedicated cybersecurity tools and secure device design.

The Intrusion Detection System (IDS), which supports cybersecurity solutions for information systems, including Industrial Control Systems (ICS) and critical infrastructures in converged IT and OT environments [4], is a commonly used method for security monitoring. The IDS enables the adoption of preventive measures against intrusion risks and seeks to detect anomalous activity through the behavior analysis [4]. However, creating a lightweight IDS for IoT is difficult, mainly due to the following problems: i) most Machine Learning (ML) based detection methods cannot be implemented on IoT edge devices due to resource constraints, and ii) lack of data processing and model learning capabilities in edge resources.

In this context, knowing that IoT end devices are based on highly limited computational resources, it is necessary to search for alternatives that use lightweight techniques that do not require great computational power, but instead have a high accuracy and are able to detect abnormalities in the network where these devices are connected, in order to bring more security in the applicability of these devices.

Having this in mind, this work presents a Multi-agent Systems (MAS) based architecture that distributes the detection of anomalies, when the system is under attack, by different nodes along the edge-cloud continuum, which collaborate for this objective. For this purpose, agents located at the edge, which have low computational power, can run less complex intelligent algorithms that are based on internal process parameters and network traffic data, offering lower latency, faster response times, and reduced dependency on

centralized cloud resources. On the other hand, agents located in the cloud can take advantage of greater computational power to run complex algorithms and support other agents, e.g., updating their learning models. In this paper, the specification focuses on the edge layer, which allows to address unique challenges regarding the limited computational resources in IoT end devices. The proposed approach was tested using the CICIoT2023 dataset [5], allowing to verify its accuracy and efficiency to properly detect attacks on the system.

The remaining paper is organized as follows: Section II presents the IDS related work, particularly highlighting the gaps and challenges in this field. Section III presents the proposed MAS-based IDS architecture covering the edge-cloud continuum and Section IV describes its specification focusing the edge level, including the behavior and interaction between agents. Section V presents the experimental implementation in a case study. Finally, Section VI rounds up the paper with the conclusions and points out some future work.

II. Cybersecurity in Constrained Devices

Industry 4.0, driven by applying emergent digital technologies, revolutionizes the industrial production, offering enhanced value and services [6]. This paradigm shift promises an increased flexibility, mass customization, and efficiency, using smart and interconnected machines and systems that enable the real-time communication and monitoring, fostering the collaboration and service-oriented business models [7], [8]. IoT is one of the main pillars for the implementation of the Industry 4.0 concept, however, security is a major issue in the operation of IoT nodes that affects how well-functioning they are over time, especially in terms of data collecting, processing and transfer. In many applications, IoT nodes are used to collect sensitive/private/personal data, or control some CPS or actuators, and an attack directed to an end device can compromise the security of the network to which it is connected, which can bring risks to the IT infrastructure. In such situation, the security assumes a crucial role to ensure the integrity, confidentiality, and availability of the data and the overall system.

Implementing conventional security measures directly on IoT networks is challenging due to the limited resources nature of IoT devices. Any data analysis carried out via IoT requires the creation of novel approaches to operate within the constrained computing budget [9]. To meet the security requirements in IoT applications, it is more effective to adopt lightweight security solutions that find a balance between cryptographic techniques and optimized resource utilization, e.g., memory and power [10]. However, most of the current IoT security research focuses on computationally intensive design, making impractical for devices with limited resources.

One type of data analysis that looks for unusual states within the system is anomaly detection, in which the algorithms act as checkpoints for incoming traffic at various stages, from the IoT network level to the data center [9]. The work proposed by [11] suggests a lightweight, host-based DoS anomaly detection and defense mechanism for IoT devices with low computing power, using a tool that combines packets as accepted, discarded, altered or marked. The proposed defense mechanism automatically matches packets based on the output of the detection algorithm. However, the rules for combining packets is done manually, making the mechanism not scalable. On the other hand, [12] presented a B-Stacking intrusion detection model that uses optimized ML algorithms to effectively detect cyber attacks on an IoT network. This was achieved by removing multicollinearity, scaling data, reducing dimensionality and sampling, enabling the model to have lower computational cost and complexity, as well as faster training.

While certain mechanisms demonstrate excellent performance, they often exhibit limitations, whether in terms of scalability or ability to identify specific types of attacks. Furthermore, every algorithm, no matter how efficient, inherently possesses vulnerabilities due to its lightweight nature. To address this, a proposed solution leverages the strengths of both cloud and edge computing levels and adopts a decentralized structure to integrate various algorithms. This approach enables the detection of a diverse array of attack types by capitalizing on the unique strengths of each technique, thereby mitigating the weaknesses associated with individual algorithms.

III. Multi-agent IDS Architecture

The design principles for the proposed architecture rely on having low complexity in data analysis and quick response in detecting attacks, since most of the analysis are performed directly on the devices located at the edge, in addition to having a distributed data analysis in the system. For this purpose, the proposed architecture, represented in Fig. 1, is composed of two layers, edge and cloud, where the detection nodes are distributed by using the MAS technology. Incorporating MAS enhances security by fostering the collaboration and distributed decision-making among edge and cloud agents. The agent autonomy facilitates adaptive responses to emerging threats, enabling the real-time attack detection and mitigation, and the capability to distribute intelligence contributes to the system scalability, accommodating the dynamic nature of IoT environments. This strategical choice is aligned with the Industry 4.0's demand for a sophisticated, resource-efficient framework, ensuring the necessary security and responsiveness.

A. Agents in the Edge-Cloud Continuum Architecture

This architecture considers several agents distributed in the edge and cloud layers, being the edge agents responsible for analyzing the parameters of a certain process or system to ensure its correct operation, and identify anomaly due to external attacks. These agents can establish cooperation with other agents when they can not perform alone their detection or diagnosis tasks. On the other hand, cloud agents are responsible for observing the global system, aiming to cooperate with edge agents to help them to meet their local goals. Cloud agents have greater computational power than edge agents, and therefore are able to run more complex ML algorithms for identifying patterns of system operation, and can also bring better inference to the parameters used in the

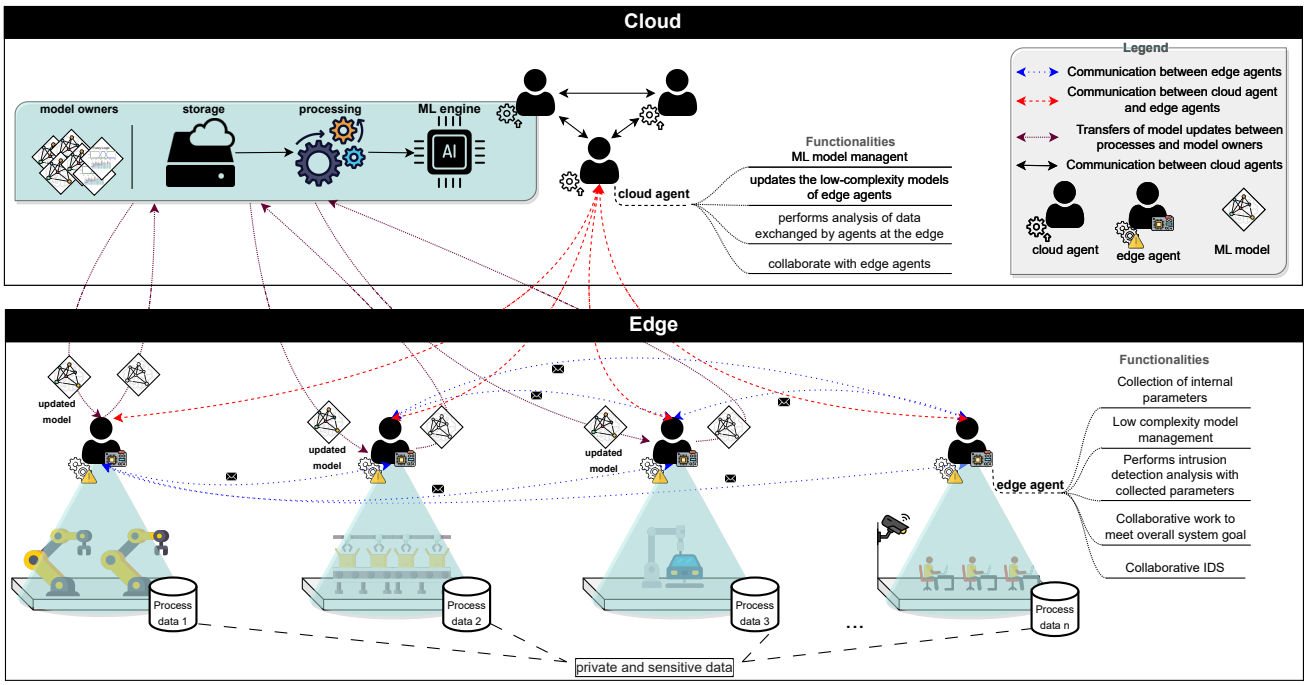

Fig. 1. Proposed MAS-based IDS architecture covering the edge-cloud layers.

models of the agents at the edge. Thus, the agents in the cloud are able to inform the edge agents to update the (ML based) models they are using, to improve their performance.

B. Communication between Agents

The primary role of the agents is to correctly identify attacks, and, in order to do this, in certain cases it is necessary to communicate with each other to detect a local or even global failure. Since this is a distributed environment, an agent can identify an attack that has been performed on another process/agent, and can also identify an attack on itself. Several different types of communication can be identified, aiming distinct objectives:

- Communication between agents on the edge: aims to collectively improve the anomaly detection and diagnostic processes through the collaboration between the various agents at the edge, sharing experiences and knowledge gained from local data analysis. By harnessing the collective intelligence of neighboring agents, the network at the edge becomes adept at identifying and understanding potential threats or attacks.
- Communication between agents on the cloud: cloud agents aggregate and bring together diverse data sets to enable comprehensive analysis and model refinements. This collaborative synergy actively influences the decision-making of the edge agents. The cloud acts as a centralized hub, validating and approving updates suggested by edge agents. Through consensus-building mechanisms, cloud agents collectively assess the potential impact of proposed modifications, ensuring alignment with global optimization objectives.
- Communication between agents at the cloud and at the edge: characterized by the cloud agents being monitoring and assessing the performance of edge agents in real-time, and performing a proactive optimization analysis when a degradation of performance or an improvement opportunity is identified, suggesting enhancements or adjustments to the edge agents' models. This bidirectional communication fosters a synergistic relationship where the computational capabilities of the cloud are harnessed to optimize the efficiency and effectiveness of edge agents, ultimately elevating the overall anomaly detection and diagnostic capabilities of the system.

Some characteristics must be taken into account when designing these collaboration patterns:

- There are some variations and modifications that can occur in the system processes, which can be homogeneous or heterogeneous. In the case of having homogeneous processes, similar or distinct algorithms can be used, and the update of ML or detection models can be performed in only one or more agents. In the case of having heterogeneous processes, this directly implies that each process has its own model.
- The agents can be in a collaborative environment, where a process can depend on information or even resources from another process, but they can also be in a competitive environment, e.g., two similar processes compete against each other for the production of a certain product. In this case, there could be negotiation between the agents to make the choice between similar processes that compete with each other.

- The information exchanged between cloud and edge agents can be limited. Agents in the edge may in some cases be handling sensitive and private data, which cannot be shared to the cloud for security reasons. As such, this data exchange may be restricted, but can be shared with other agents at the edge for collaboration in analysis.
- Agents have intelligent modules and intelligent mechanisms. The intelligent modules consist of AI algorithms, e.g., Neural Networks, Decision Trees, Support Vector Machine and Fuzzy Logic. The intelligent mechanisms consist of message filters and direct analysis of the messages exchanged by the various agents, allowing to understand various critical situations, in which the agents are able to identify them and act directly when detecting these situations.

C. Management of Data Analysis Models

The structure of the edge-cloud connection for updating ML models follows the basis/concept of Federated Learning (FL) [13] approach, but differs in some aspects. In terms of resource distribution, in the FL approach, the clients are allocated on the edge and the models are embedded in the devices, while the server is allocated in the cloud, and the information of the model parameters is exchanged from the edge to the cloud for model improvement. In the proposed MAS-based approach, agents can be scattered between edge and cloud, depending on the type of service that will be offered and the required resource, but an agent at the edge can collaborate with another one that is also at the edge by assisting it in the analysis procedure.

Regarding adequacy, FL is more suitable when there are the same or very similar processes, which will be trained with local data and the parameters of the trained model will contribute to the improvement of a global model that will make updates to the edge models. On the other hand, the proposed MAS-based approach is suitable for similar or identical processes, but also for complementary processes, where the collaboration of different processes aims at a global objective, where the knowledge from one process is transferred to another through agents.

In respect of general use, FL is useful when there are privacy or security restrictions as it allows data to remain locally on devices or servers, mainly when dealing with sensitive data that can not be transferred to a central server. For the proposed MAS-based approach, it is best suited when agents can share information about the model itself and when collaboration on a broader level is desired. If there is diversity in the models of each agent or if different agents have complementary knowledge, collaborative learning can be more effective in improving the performance of the models.

In terms of computational resource, FL requires less communication between devices and servers, since only the model weights are shared, while for the proposed MAS-based approach, besides the model parameters, the information about the system operation is also shared. In general, constraints on privacy, security, computational resources, and system goals should be considered. In summary, the main difference between FL and MAS-based approach is the nature of the collaboration, as FL uses only model weights, and MAS has a richer collaboration, it also uses information about the model itself.

In general, the proposed architecture aims to cover confidentiality, integrity and availability of information, restricting information to authorized entities and processes. Certain private and sensitive data is only shared (when possible) at the edge and not forwarded to the cloud. There is control over which agent can access information from certain processes, and the data is not altered, it is only used by the models, maintaining its integrity.

IV. Specification of the Edge Level

In a distributed system, what dictates how the system works is how the agents interact with each other. In this sense, considering that the focus of this work will be on the edge level, the specification described in this paper focuses the behavior of the edge agents and the interaction between them for making collaborative decisions about detecting attacks.

A. Behavior of Edge Agents

The agent's behavior for detecting threats is modeled using the Petri nets formalism, as shown in Fig. 2. Briefly, this behavior consists of periodically monitoring certain acquired information to perform a threat analysis (*t7*), in which according to its expertise, it can reach a conclusion (*t8*) or request the collaboration (*t9* and *t10*) to other agents to perform a deeper analysis aiming a more accurate detection and diagnosis (*t12*). In addition, the edge agent can receive messages from other agents requesting collaboration for their data analysis (*t4*), which requires to run its local model (*t5*) with the other agent's data; the achieved results are sent back to initiators (*t6*). After reaching a final conclusion on the detection of an anomaly, the process continues to the prevention phase if an attack has been detected (not covered in this work).

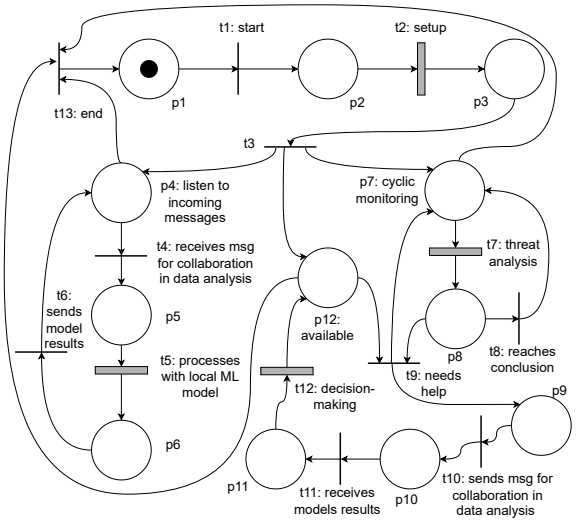

Fig. 2. Generic behavior of an edge agent.

B. Interaction Between Edge Agents

The interaction between agents in a MAS system is crucial for the orchestration and optimal functioning of the system. The interaction of the agents in the edge is shown in Fig. 3, and follows the objectives identified in the previous section. As mentioned earlier during the description of the agent's behavior, they send messages requesting collaboration in the data analysis, sending their local data, incorporated in the messages, to be analyzed by the other agents. This interaction always takes place in such a way as to have a well-defined and coordinated operation between the various agents present in the system.

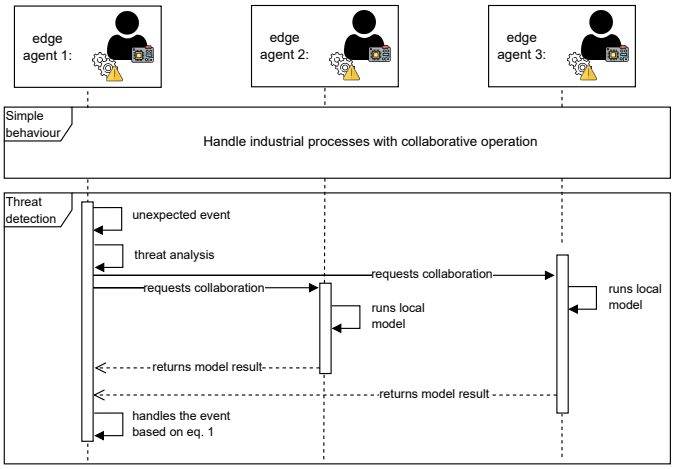

Fig. 3. Interaction between agents at the edge level.

In short, whenever an unexpected event is detected by the agent and it is unable to reach a conclusion with its local threat analysis, it requests the support from the other agents, sending the data relating to the unexpected event, so that it can be analyzed by the other agents. The agents that receive the request for collaboration will analyze the data with their local ML techniques and send a response to the agent that requested the help, which consists of its classification (whether it is an attack or not) and the metrics of its local model (i.e. accuracy, precision, recall, fscore, etc), based on past events, to inform the confidence of the analysis carried out. The agent that requested the collaboration will compile the responses from the other agents and will be able to perform a more accurate decision-making, as described in the next sub-section.

C. Decision-making by Edge Agents

After receiving the responses as a result of the collaboration with the other agents in analyzing the data to classify if the identified data is a threat/attack or normal data, the edge agent performs a decision-making by using the following equation:

$$FD = \begin{cases} 1, & \text{if } \left(\frac{\sum_{i=1}^{n}(W_i \cdot S_i \cdot \mathbb{I}(M_i \geq \beta))}{\sum_{i=1}^{n} W_i}\right) \geq \alpha \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where M_i is the performance metric associated with the technique i, W_i is the weight associated with the technique i based on its performance (e.g., M_i, which can be accuracy, precision, recall, f1-score, etc.) and S_i is the prediction of attack provided by the technique i (0 or 1). The $\mathbb{I}$ (condition) is an indicator function that returns 1 if the condition is *true* and 0 otherwise, n is the total number of techniques used, α is the threshold for the final evaluation, being a value between 0.5 and 1 (ensuring that it is only considered an attack if at least 50% of the techniques have predicted this value), and β is the minimum value of the performance metric that will be accepted in the collaboration, ensuring that the weight W_i is only taken into account in the weighted average if the performance metric M_i is above the desired minimum value (β). The indicator function $\mathbb{I}(M_i \geq \beta)$ has a value of 1 if the condition is true and 0 otherwise.

Briefly, after receiving the analysis of all agents, a weighted average will be calculated with the classification of each agent (whether it is an attack or benign), where the weights are the confidence of these classifications (e.g. accuracy). However, a condition has been added to improve the assertiveness of the decision, inserting the indicator function so that only decisions with a minimum acceptable confidence are considered.

V. Experimental Case Study

The proposed approach was tested using the CICIoT2023 dataset [5], which contains data from 105 devices and 33 different types of attacks in IoT environments. Considering the proposed approach, each edge agent will embed a ML model that will be trained with data from the dataset using a different ML technique, and the interaction between the agents follow the specification provided in the previous section.

The MAS system was implemented with the JADE framework, with the behaviors following the specification described in Fig. 2. For the threat analysis phase and local model processing (transitions *t7* and *t5* in Fig. 2, respectively), a python sub-process was implemented internally in the behavior to deal with the ML models in each agent, which involves using the *Runtime* class in Java to initiate a separate process that executes a specified Python script (ML algorithm). The communication between the JADE agent and the Python process is established through standard input/output streams. The integration of JADE and Python through this method opens avenues for collaborative intelligence where the strengths of both platforms are harnessed. This approach is particularly beneficial when intricate ML tasks can be offloaded to Python, maintaining flexibility within the JADE environment.

Three agents were implemented, all with the behavior described above, but with different data analysis techniques. All agents had their models trained in isolation with data from the CICIoT2023 dataset, in which the data was divided into 80% for training and 20% for testing. The data consists of network traffic data, in which attacks are classified into seven categories, namely DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai. The interaction between the agents resulted in a collaborative decision using equation 1 to classify the data, which was evaluated with the same 20% of the data reserved for testing. The values considered

were 0.6 for α, 0.9 for β and the values of the f1-score metric were chosen for M_i and W_i. The techniques considered for each one of the three different agents were the Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR), implemented with the scikit-learn library. The techniques and the collaborative decision were evaluated in terms of accuracy, precision, recall and f1-score. The achieved results are shown in Table I.

TABLE I
RESULTS OF APPLYING THE ML TECHNIQUES ALONE AND WITH THE PROPOSED INTERACTION BETWEEN THE EDGE AGENTS.

Technique	Accuracy	Precision	Recall	F1-score
LR	0.9890	0.8631	0.8904	0.8762
RF	0.9968	0.9672	0.9645	0.9658
SVM	0.9924	0.9465	0.9010	0.9225
Edge agents decision	0.9938	0.9806	0.9039	0.9388

The achieved results, only considering the operation of the edge level, show that by combining various techniques there has been a significant increase in the precision, which means a reduction in false positives, which is very positive from the perspective of control systems, preventing normal network traffic data from being blocked by the system. Furthermore, on average, the collaborative model performs better than most of the isolated techniques, bringing distinct strengths with the diversity of techniques and allowing a more holistic analysis of the data. This shows that the cooperation between agents can boost the detection of attacks in the system, and above all, without overloading a central node, since all the analysis is done in a distributed manner. It is worth mentioning that depending on the values set for α, β, the metric M_i, and the addition of more modules to the system, even better results can be achieved. In addition, considering that the models can and should be trained with different data, collaboration could achieve very significant improvements, as they may have different local knowledge and witness different attacks. Also, other models can be used, such as neural networks.

VI. CONCLUSIONS AND FUTURE WORK

This paper discusses the importance of security issues in constrained IoT devices, primarily aiming to ensure the system availability and protection. Recognizing the limitations of traditional cybersecurity solutions, especially in the context of resource-constrained IoT edge devices, this work proposes an edge-cloud continuum MAS-based IDS architecture. This innovative approach leverages the collaboration between edge and cloud agents, capitalizing on the strengths of each. Agents deployed at the edge, equipped with limited computational power, employ less complex intelligent modules based on internal process parameters and network traffic data. Meanwhile, cloud-based agents leverage substantial computational resources to execute complex algorithms and support edge agents by, for example, updating their models.

The proposed approach was tested by implementing the interaction between edge agents, and validated by using the CICIoT2023 dataset to simulate the network traffic from different processes and devices on the edge. The dataset comprehensively addresses 33 different attacks on an IoT topology that includes 105 different devices, providing a realistic environment for evaluating and validating security analysis applications in real IoT operations. The interaction between agents at the edge level in a multi-agent framework proves to be a suitable solution for addressing the challenges of the lightweight intrusion detection in IoT environments, showing promising results in terms of accuracy and efficiency in detecting system attacks.

Future work will be devoted to specify the operation of agents placed at the cloud level (behavior and interaction), as well as the collaboration between edge and cloud agents regarding, e.g., the update of the models deployed in the edge agents.

ACKNOWLEDGMENTS

This work has been supported by the Foundation for Science and Technology (FCT, Portugal) through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021). The author Gustavo Funchal thanks the FCT for the PhD Grant 2022.13712.BD.

REFERENCES

[1] H. Kagermann, W. Wahlster, and J. Helbig, "Securing the Future of German Manufacturing Industry: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0," ACATECH, Tech. Rep., 2013.

[2] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[3] Kaspersky, "Pushing the limits: How to address specific cybersecurity demands and protect IoT," https://www.kaspersky.com/blog/iot-report-2022/, January 2022.

[4] D. Hamouda, M. A. Ferrag, N. Benhamida, and H. Seridi, "Intrusion detection systems for industrial internet of things: A survey," in *2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS)*, 2021, pp. 1–8.

[5] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, 2023.

[6] A. Khan and K. Turowski, "A survey of current challenges in manufacturing industry and preparation for industry 4.0," 2016.

[7] B. P. Santos, F. Charrua-Santos, and T. Lima, "Industry 4.0: an overview," in *Proceedings of the World Congress on engineering*, vol. 2. IAEN London, UK, 2018, pp. 4–6.

[8] H. Cheng, P. Zeng, L. Xue, Z. Shi, P. Wang, and H. Yu, "Manufacturing ontology development based on industry 4.0 demonstration production line," *2016 Third International Conference on Trustworthy Systems and their Applications (TSA)*, pp. 42–47, 2016.

[9] A. Chatterjee and B. S. Ahmed, "Iot anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, p. 100568, 2022.

[10] R. Nath N and H. V Nath, "Critical analysis of the layered and systematic approaches for understanding iot security threats and challenges," *Computers and Electrical Engineering*, vol. 100, p. 107997, 2022.

[11] J. J. Kponyo, J. O. Agyemang, G. S. Klogo, and J. O. Boateng, "Lightweight and host-based denial of service (dos) detection and defense mechanism for resource-constrained iot devices," *Internet of Things*, vol. 12, p. 100319, 2020.

[12] S. Roy, J. Li, B.-J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for iot networks," *Future Generation Computer Systems*, vol. 127, pp. 276–285, 2022.

[13] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.