Blockchain and Deep Learning-Based IDS for Securing SDN-Enabled Industrial IoT Environments

Samira Kamali Poorazad, Chafika Benzaïd, and Tarik Taleb
Oulu University, Oulu, Finland
{samira.kamalipoorazad; chafika.benzaid; tarik.taleb}@oulu.fi

***Abstract*—The industrial Internet of Things (IIoT) involves the integration of Internet of Things (IoT) technologies into industrial settings. However, given the high sensitivity of the industry to the security of industrial control system networks and IIoT, the use of software-defined networking (SDN) technology can provide improved security and automation of communication processes. Despite this, the architecture of SDN can give rise to various security threats. Therefore, it is of paramount importance to consider the impact of these threats on SDN-based IIoT environments. Unlike previous research, which focused on security in IIoT and SDN architectures separately, we propose an integrated method including two components that work together seamlessly for better detecting and preventing security threats associated with SDN-based IIoT architectures. The two components consist in a convolutional neural network-based Intrusion Detection System (IDS) implemented as an SDN application and a Blockchain-based system (BS) to empower application layer and network layer security, respectively. A significant advantage of the proposed method lies in jointly minimizing the impact of attacks such as command injection and rule injection on SDN-based IIoT architecture layers. The proposed IDS exhibits superior classification accuracy in both binary and multiclass categories.**

***Index Terms*—Blockchain, Industrial IoT, SDN, Deep learning, Intrusion detection system, and Security.**

I. Introduction

The Internet of Things (IoT) has been increasingly adopted across various fields, including agriculture, manufacturing, and industry [1]. The Industrial Internet of Things (IIoT) is an application of IoT in manufacturing and industry that aims to automate industrial processes and achieve effective and appropriate products through data exchange and digitization [2]. IIoT offers advantages over traditional Supervisory Control and Data Acquisition (SCADA) systems, such as productivity, scalability, and data analysis [3], but the increase in connected devices and the lack of security design in older control systems make factories vulnerable to cyber-attacks. Industrial Control Systems (ICS), which include SCADA, Remote Terminal Unit (RTU), and Programmable Logic Controllers (PLCs), play a crucial role in various industrial infrastructures such as nuclear technology.

IIoT security research has increased due to the inadequacy of conventional methods such as access control mechanisms, firewalls, and encryption against multiple attacks such as denial of service (DoS) [4]. Machine learning methods have been used to detect intrusions, but they often fail to detect unknown and new security threats due to poor feature selection and classification [4], [5]. This approach is also not scalable for large volumes of data, and incorrect classification can have catastrophic consequences [6] such as a nuclear disaster in a nuclear power plant.

Ensuring accurate classification of attacks is crucial for timely analysis [6], [7]. Unbalanced datasets from ICS, which are usually in a stable and normal state, may affect machine learning algorithms [6]. To detect attacks quickly and accurately, Deep Learning (DL) methods [4], [6] have shown advantages over traditional machine learning methods, such as the ability to learn features from original data and manage high-dimensional data to extract valuable patterns. Leveraging the potential of DL, this study aims to eliminate the aforementioned limitations by implementing anomaly detection and attack classification using a DL-based IDS.

Besides security, current IIoT architectures suffer from issues such as scalability, monitoring, data management, and flexibility (since operators must manually configure devices whenever a device update request is received) [8]. Integration of SDN into IIoT architecture addresses these challenges [8]. In fact, SDN technology decouples the data plane and control plane, allowing for centrally and intelligently control of network behavior [9]. In addition, SDN programmability enables the integration of advanced services for managing the network, including its security. Driven by its benefits, this study uses SDN for two primary reasons. Firstly, it leverages its programmability feature to facilitate the timely detection and mitigation of attacks through the IDS; which is implemented as an SDN application. Secondly, it facilitates monitoring, data management, and flexibility in IIoT architectures.

Nevertheless, besides its advantages, SDN architecture is also prone to several vulnerabilities that can be exploited by attackers, such as man-in-the-middle (MITM) attacks [10]. Therefore, strengthening SDN security is crucial to reap its benefits in creating a reliable communication infrastructure for IIoT environments. Blockchain technology is a promising candidate to meet this goal, owing to its inherent features of transparency, immutability, traceability, and decentralization [11]. It operates as a distributed ledger, which is immutable and tamper-proof, and stores data on a peer-to-peer network.

Extensive research work has been engaged, investigating the security challenges in both SDN and IIoT technologies (e.g., [4], [6], [12], [13]) as well as leveraging SDN in IIoT for its perceived benefits (e.g., [8], [14]). However, none of existing contributions has comprehensively addressed security concerns in both technologies in an integrated way. Specifically, there is a lack of research that addresses security issues holistically, rather than separately, with the aim of mitigating the impact of

979-8-3503-1090-0/23/$31.00 © 2023 IEEE

attacks on the various layers of SDN-based IIoT architecture simultaneously.

It is important to clarify that our objective is not to propose a method that is superior to previous research in terms of IDS accuracy or Blockchain overhead. Rather, the primary differentiator of our approach from related work lies in the proposal of a novel and improved integrated method that aims to minimize the effects of IIoT security attacks and SDN security attacks on the various layers of an SDN-based IIoT architecture. This is achieved through the use of two security components – a DL-based IDS and a Blockchain-based system (BS) – that are integrated in a manner that allows them to work together and complement each other, resulting in a more comprehensive security approach. As a DL algorithm, a convolutional neural network (CNN) is used. Because the selected dataset (version 3) [15] is supervised and classification-oriented, CNN has been chosen for its capacity to automatically extract relevant features. The results of our evaluation also showed that CNN outperformed some machine learning algorithms in detecting and classifying attacks. This has been the motivation for pursuing the following:

- Introducing a SDN-based IIoT system that combines the benefits of both technologies to improve flexibility and scalability while also addressing security concerns.
- Proposing a novel method that utilizes two security components. Through this improved approach, the impact of security attacks across multiple layers of the SDN-based IIoT architecture will be minimized.
- Presenting and implementing a system model and attack model about SCADA network attacks and attacks specific to SDN networks.
- Evaluating the efficiency of the proposed solution against the attacks outlined in the attack scenario.

The rest of this paper is organized as follows. Section II categorizes existing articles related to the field. In Section III, the proposed method, system model, and intended attack model are presented. Sections IV and V present implementation details and evaluation results, respectively. Finally, Section VI concludes this paper and highlights some of our future work in this area.

II. Related Work

Three categories of related research are reviewed in this section, namely: IoT/IIoT Security and IDS-based solutions, SDN-based IoT/IIoT Security, and SDN Security and Blockchain-based solutions.

A. IoT/IIoT Security and IDS-based Solutions

Balil *et al.* [3] discussed classification and mitigation solutions for IIoT attacks. In [4], a two-step detection system is proposed: a machine learning-based anomaly detection module is used in the first step for binary classification. The output of the first step is used as the input for the CNN-Long short-term memory (LSTM) algorithm for multiclass classification in the second step. A 2D CNN algorithm is introduced in [6] to identify anomalies in industrial traffic. The idea is to convert one-dimensional traffic data into two-dimensional images representing specific traffic classes. As a result, CNN is capable of classifying these images effectively in order to detect anomalies. Rakas *et al.* [16] reviewed recent articles on SCADA security using IDS, and Alotaibi *et al.* [17] used stacked deep learning to detect malicious attacks targeting IoT devices in smart homes and smart grids. Gao [18] designed a SCADA anomaly-based IDS with different algorithms, such as Decision Tree (DT).

B. SDN-based IoT/IIoT Security

The use of SDN and blockchain in IIoT is proposed to enhance smart grid flexibility, energy optimization, and security against attacks in various articles, including [8], and [14]. In [8], SDN detects vulnerabilities and attacks and facilitates continuous IoT device monitoring. The authors leveraged Blockchain to secure Cloud data, counter Distributed DoS (DDoS) attack threats, and guard against distributed controller attacks. Machine learning-based models and clustering algorithms are also suggested to optimize resource consumption and prevent attacks. SDN technology is used in [19] and [20] to prevent MITM and DDoS attacks and enhance scalability and flexibility in IoT networks. Haseeb *et al.* [21] proposed a SDN-enabled security model using machine learning to improve network consumption and delivery of the Internet of Medical Things (IoMT) services on time. SDN clusters the nodes and optimizes routing performance using the unsupervised learning algorithm. Moreover, the intelligent centralized SDN controller protects data, minimizes power consumption, and manages critical infrastructure effectively, safeguarding against malicious users and unauthorized requests.

C. SDN Security and Blockchain-based Solutions

Liu *et al.* [10] reviewed different types of attacks on SDN. Scott *et al.* [22] investigated vulnerabilities in software-based networks and suggested security solutions. Derhab *et al.* [12] used a distributed controller and a blockchain between controllers to prevent flow rule injection attacks, and Boss *et al.* [13] prevent DDoS attacks using blockchain technology.

A number of articles investigated the issue of SDN security and IIoT networks, emphasizing the demand for an integrated solution to ensure network security and avoid additional costs. The work in [14] introduces a Random Subspace Learning (RSL)-based IDS method for IIoT attacks and a blockchain-based method for SDN attacks, but does not demonstrate how to implement detection based on the blockchain, and the role of SDN is not clearly defined. In contrast, our work outlines a novel approach for detecting SDN attacks based on the southern interface and emphasizes the role of SDN as a key component in detecting IIoT-related attacks through the use of an IDS. We make use of SDN's programmability feature to achieve this goal.

III. Methodology

This section describes the proposed approach, the attack model, and the system model of the desired architecture.

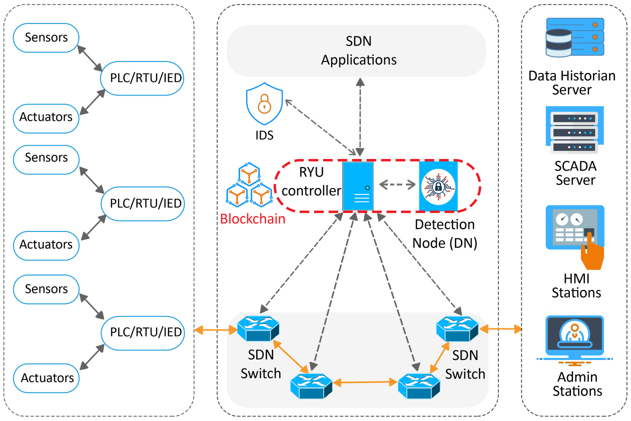

Fig. 1. Layered architecture of SDN-based IIoT networks.

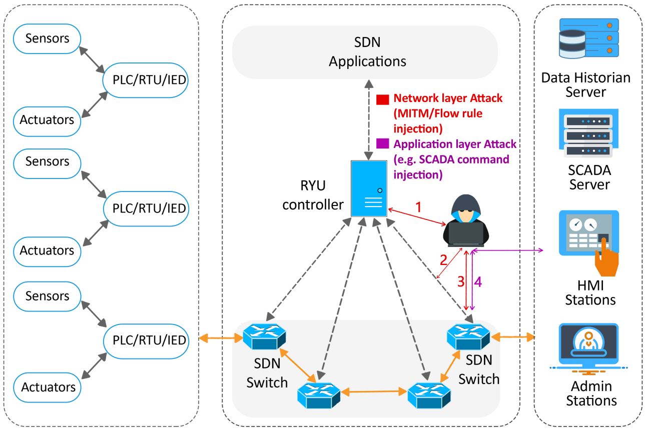

Fig. 2. Attack model.

A. System Model

The proposed system model, illustrated in Fig. 1, uses SDN technology at the network layer to transmit environmental information from sensors to industrial controllers and ultimately to the control room. The SDN-based network is responsible for transmitting information and making decisions based on data received from the sensors. Two security components, namely blockchain and IDS, are located within the system to enhance its security.

Fig. 1 shows the blockchain consisting of two nodes; an SDN controller/a block generator and a Detection Node (DN). The block generator has read and write access, while the DN can only read the block. In addition, the IDS is also placed as an application on the SDN controller - the IDS is trained by CNN algorithm. The blockchain plays a crucial role in our work by supporting the DN in more accurately detecting attacks.

B. Attack Model

Fig. 2 depicts a proposed attack model considering MITM attacks at the network layer, involving the injection of flow rules into switches' flow tables and command injection attacks at the application layer (the control room). Two types of attacks are considered: purple attacks modifying packet payload and red attacks changing packet header. Purple attacks alter actuator performance, while red attacks redirect packets to unintended destinations, causing chaos in the network. The control room's command could reach the wrong actuator and unintentionally perform an undesired action due to such attacks.

C. Method Suggested

In this method, both the packet payload and header are examined to ensure that neither command injection nor flow rule injection has taken place.

The first assumption is that a packet enters the switch. The switch performs an action on a packet if it is defined in its flow table, otherwise it sends it to the controller. Before considering any action, the controller forwards the packet to the IDS for detection of the malicious payload. If the IDS detects a malicious payload, the controller blocks the attacker, but if the packet payload is safe, the header is checked by the BS which includes a controller node and a DN. Thus, the BS is activated once the IDS has confirmed that the packet payload is safe.

Due to this, once the IDS has determined that the packet payload is normal, the controller decides on the packet by sending a flow rule to the switches in order to update their flow tables. Simultaneously, the blockchain is used to transmit the same flow rule to the DN. The DN stores the flow rules received from the controller via the blockchain in a file and requests the switch logs for analysis. Following that, DN saves the switch logs in another file. As a result, DN compares these two files to ensure that the flow rule has not been changed along the way (i.e., southern interface) due to a possible MITM attack. When the DN detects a change in the values sent by comparing these two files, it will generate an attack warning for the controller, otherwise, it will generate a warning for safe flow for the controller (the packet header is also safe). Consequently, by sending a malicious packet payload or a false flow rule, the attacker is not able to reach his target in this case. Therefore, neither the command sent from the controller room to change the performance of the actuators nor the packet headers to get the packet to the wrong destination have changed. A description of the proposed method is provided in Fig.3. In Fig.3, the arrows specified in the "Sending flow rules" section with the "*" sign occur simultaneously.

IV. Implementation

The proposed method is implemented using the Python programming language and the Numpy 1.18.5, Pandas 3.8.10, Keras1.1.2, and Sklearn 0.22.2 libraries to implement the IDS based on the CNN algorithm. Additionally, the MultiChain private blockchain is implemented using the Savoir library. Mininet 2.3.0, OpenFlow 1.3, and Ryu Controller are used as simulators, southbound interface, and SDN controller, respectively. In addition, two Ubuntu 20.04 virtual machines are used and the specifications of the systems and programs that are installed on them are listed in Table II.

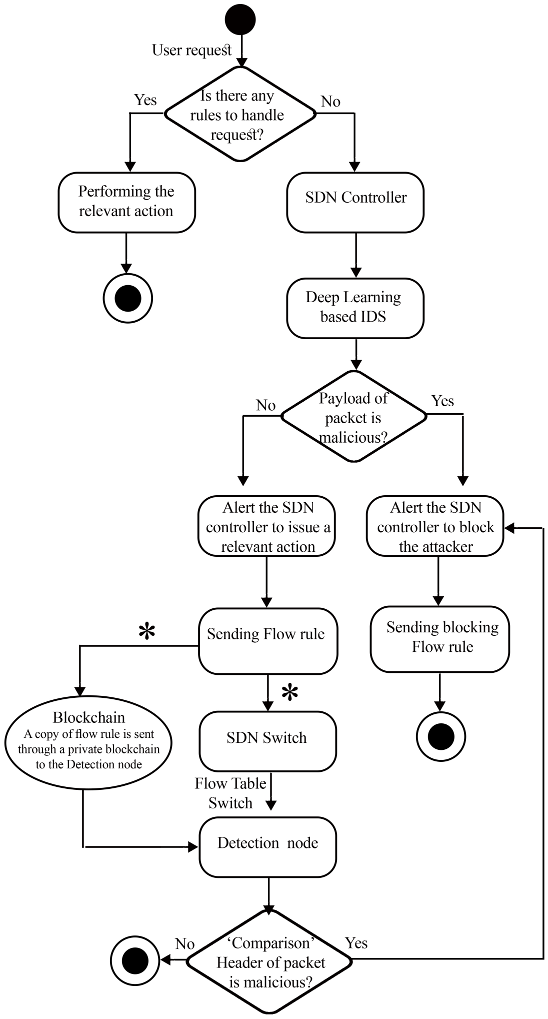
User request
Is there any rules to handle request?
Yes
No
Performing the relevant action
SDN Controller
Deep Learning based IDS
No
Payload of packet is malicious?
Yes
Alert the SDN controller to issue a relevant action
Alert the SDN controller to block the attacker
*
Sending Flow rule
Sending blocking Flow rule
*
Blockchain
A copy of flow rule is sent through a private blockchain to the Detection node
SDN Switch
Flow Table Switch
Detection node
No
'Comparison' Header of packet is malicious?
Yes

Fig. 3. Flow chart of the proposed method.

This study uses a dataset in the field of natural gas pipelines [15]. The dataset contains 27 features. It includes 8 classifications, one of which is designed for normal mode and seven for the attack. The specific classifications of the natural gas pipeline dataset are shown in Table I. More details on the description and classification of these features are in [23].

The CNN-based IDS is implemented after performing data pre-processing operations to clean the input data and improve accuracy. Features with only one value are removed, reducing the number of features from 27 to 18. Data grouping is used to balance the data in each of the eight classifications, resulting in four classifications: normal, injection attack, reconnaissance attack, and DoS attack. This is achieved by grouping five types of injection attacks into one attack group and defining one type of attack as an injection attack. It is worth mentioning that the DoS attack detected by the IDS in this study is a false state injection, not a flooding attack.

The dataset is split into three parts: 70% for training, 15% for validation, and 15% for testing. The data is then normalized using the MinMaxScalar, which places the values between 0 and 1.

TABLE I
ATTACK CLASSIFICATION IN THE DATA SET.

Abbreviation	Label	Label type
Normal	0	Normal Behaviour
NMRI	1	Naive Malicious Response Injection
CMRI	2	Complex Malicious Response Injection
MSCI	3	Malicious State Command Injection
MPCI	4	Malicious Parameter Command Injection
MFCI	5	Malicious Function Code Injection
DoS	6	Denial of Service
Recon	7	Reconnaissance

TABLE II
SPECIFICATIONS OF SYSTEMS USED IN RESEARCH.

System	Operating System	RAM	Installed Program
1	Ubuntu 20.04	8 GB	Mininet 2.3.0, Ryu 4.34
2	Ubuntu 20.04	4 GB	Multichain 2.2

Afterward, the CNN model is defined. It has 2 layers of 1D-convolution, 1 layer of Max-pooling, 1 layer of Average-pooling, and 2 Fully connected layers. The filter size in the convolution layer is 3, while in the max-pooling layer, it is 2. The Relu activation function is used in the convolution layer, sigmoid for binary classification, and softmax for multi-class classification in the last layer. The batch size is set at 100, with SGD as the optimization function. The learning rate and momentum are set as 0.01 and 0.8, respectively.

It is important to note that the selection of hyperparameters, including the number and type of layers, optimizer function, and learning rate, were determined through a process of trial and error. We tested various learning rates of 0.1 and 0.01 as well as momentum values of 0.8, 0.9, and 0.99. Ultimately, we found that the combination of a learning rate of 0.01 and momentum of 0.8 by SGD produced the most optimal results. However, due to space constraints, in this paper, we are only able to present the results of the model trained with SGD using a learning rate of 0.01 and momentum of 0.8.

Following the initial steps and defining the model using the aforementioned values, the model is implemented with 30 training epochs and then evaluated according to metrics. Section V presents the results of this evaluation based on the considered metrics. Following the implementation of IDS, each of the systems listed in Table II will be described.

A. *System 1 Operation in Implementation*

The system is simulated using Mininet and Ryu controller, with a network topology consisting of 2 switches, 3 hosts, and 1 controller with communication links. The presented topology uses a host called Client for legal communication, a host called Attacker for sending the false flow rules, and a host called Server for receiving requests from users. After implementing the topology in Mininet and its connection to the controller, it is possible to send packets with a specific payload. The trained CNN-based IDS model is saved in a file and called on the Ryu controller. An IDS Python application is created to receive packet payload, read its specifications, call the corresponding IDS, define flow rules sent by the controller, and perform blockchain operations.

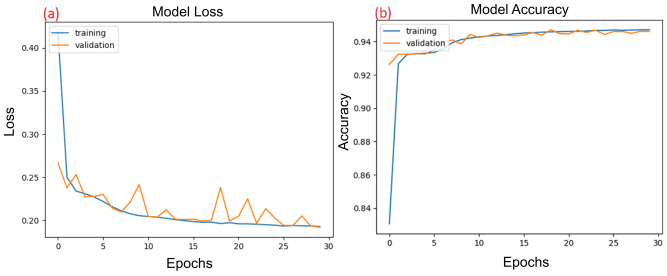

Fig. 4. Model's accuracy and loss at every epoch in binary-class classification.

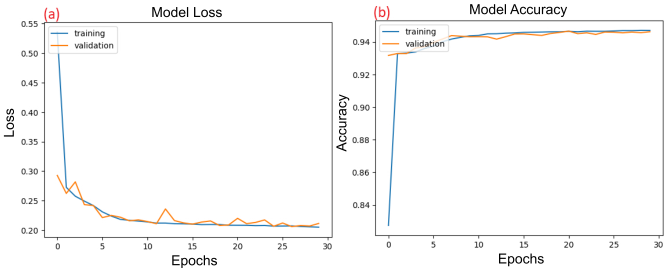

Fig. 5. Model's accuracy and loss at every epoch in multi-class classification.

B. *System 2 Operation in Implementation*

The controller sends a flow rule to the switch after the client hosts send a normal payload and the IDS verifies it. The DN (system 2) receives a copy of the same flow rule via Savoir, which is transmitted to the switch. The DN compares the switch flow table with the flow rule received from the blockchain. A MITM occurs when the number of flow table rows received from the switch does not match the number of flow table rows received from the controller. If the rows match in both files but the content does not, it suggests a modification in the flow rule at the switch. This is reflected in the ”Hard_age” feature of the switch flow table, which records the time since the entry was last modified. Hence, Hard_age can be used to detect flow table changes. Whenever a flow rule injection attack is detected, the DN notifies the controller that the packet is malicious, indicating that the flow rule sent from the SDN controller to the switch has been tampered with.

V. Evaluation

This section evaluates the IDS before calling in the Ryu controller, and in the next step, we evaluate the IDS after calling in the Ryu controller and the BS.

A. *IDS Evaluation Before Calling in Ryu*

The proposed IDS is evaluated using four performance metrics: accuracy, precision, recall, and F1-score.

The accuracy represents the overall performance of the classifier. Precision and recall ensure that the results were not distorted by too many normal samples (unbalanced dataset). Finally, F1-score acted as a reconciler between precision and recall.

Fig. 4, Fig. 5, Table III and Table IV present the evaluation results of the proposed algorithm based on the above-mentioned metrics. Fig. 4 as well as Fig. 5 illustrate the loss (a) and accuracy (b) of the binary-class and multi-class classification models during training, respectively. An evaluation of the model was done with a learning rate of 0.01 and momentum of 0.8 in 30 epochs.

TABLE III
Binary classification performance results.

Class	Precision	Recall	F1
Normal	94.17%	98.3%	96.6%
Attack	96.39%	89.65%	92.90%
Average results	95.28%	93.84%	94.48%

TABLE IV
Multi-class classification performance results.

Class	Precision	Recall	F1
Normal	94.11%	98.04%	96.3%
Injection	95.32%	88.29%	91.67%
DoS	98.67%	69.13%	81.30%
Reconnaissance	100%	100%	100%
Average results	97.02%	88.86%	92.31%

As previously mentioned, the accuracy of the model was tested using various hyperparameters and optimization functions. The accuracy of the model in the binary-class mode reached 93.50% with the learning rate of 0.1 and the momentum of 0.8. With the learning rate of 0.01, it reached 94.75%. In the multiclass mode, the model accuracy reached 93.30% with the learning rate of 0.1 and the momentum of 0.8. It reached 94.65% with the learning rate of 0.01. Model accuracy in the binary-class and multi-class modes reached 63% when the learning rate was set to 0.1 and the momentum was set to 0.99.

Based on the evaluations, it is demonstrated that the model trained using a learning rate of 0.01 and momentum of 0.8 by SGD outperforms other models. Table III shows the outcomes of binary classification, while Table IV demonstrates the results of multi-class classification.

To assess the effectiveness of the proposed algorithm, in comparison to the algorithms presented in [14] and [18], we utilize DT method and the RSL-K-Nearest Neighbor (KNN) approach to evaluate their accuracy performance in detecting SCADA attacks. Table V presents the outcomes of binary and multi-class classification based on the accuracy metric. We experimented RSL-KNN [14] with different K – the number of nearest neighbors – values of 5 and 10, but the best accuracy achieved was below 91.9% in both binary and multi-class classification. The accuracy of the DT method in both binary and multi-class classification was 92.3%, as well. We can observe that CNN outperforms RSL-KNN and DT under both classification tasks.

B. *IDS Evaluation in the Form of an SDN Application*

After ensuring that the controller is properly prepared, packets containing the desired payloads are sent through the server. The effectiveness of the IDS application in detecting both malicious and normal payloads is then assessed through careful evaluation. The attacker system sends a request with a malicious payload to the server, which is then sent to the controller for a decision as there is no appropriate action in the switch flow table. The controller forwards the packet to the

TABLE V
ALGORITHM COMPARISON RESULTS.

Algorithms	Accuracy in binary class classification	Accuracy in multi-class classification
CNN	94.75%	94.65%
RSL-KNN [14]	(K = 10) 90% (K = 5) 91.9%	(K = 10) 90.2% (K = 5) 91.9%
DT [18]	92.3%	92.3%

IDS application, which announces a value of 1 if the packet contains a detected malicious payload. The IDS application then sends a command to the controller to block the source system.

After that, the IDS application is tested by sending normal payloads to the server. The flow rule between the client and server is sent to the switches via the SDN controller, and a copy of the controller flow rule sent to the switch is collected. The collected flow tables are sent to the DN (system 2) using the Savoir module as a block. The DN receives the block and stores the results in a file. The DN then saves the content of the switch flow table in another file. The existence of different number of rows in flow tables in the two files when compared by DN indicates a MITM attack. The system also reports a modification when the switch flow table contains Hard_age and the number of rows is equal.

VI. CONCLUSION

This research utilizes CNN-based IDS and blockchain as complementary components to enhance the security of SDN-based IIoT architecture. SDN's programmability, centralized controller, and network-wide view make it a crucial factor in ensuring IIoT security. The study achieved 94.75% accuracy in binary classification and 94.65% in multi-class classification, along with satisfactory precision, recall, and F1-score metrics. The proposed method effectively detects malicious payloads and prevents their occurrence, as well as detects flow rule injection attacks. Future work could involve using balancing methods and alternative algorithms to improve classification metrics and multi-controllers can be implemented to prevent system bottlenecks.

ACKNOWLEDGMENT

This research work is partially supported by the Business Finland 6Bridge 6Core project under Grant No. 8410/31/2022, the Research Council of Finland (former Academy of Finland) IDEA-MILL project (Grant No. 352428), the Research Council of Finland 6G Flagship program (Grant No. 346208), and the European Union's Horizon Europe research and innovation programme HORIZON-JU-SNS-2022 under the RIGOUROUS project (Grant No. 101095933). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Y. Chen, T. Sun, B. Yang, and T. Taleb, "Joint caching and computing service placements for edge-enabled iot based on deep reinforcement learning," *in IEEE IoT Journal*, vol. 9, no. 19, pp. 19 501 – 19 514, Oct. 2022.

[2] Q. Guo, R. Gu, H. Yu, T. Taleb, and Y. Ji, "Probabilistic-assured resource provisioning with customizable hybrid isolation for vertical industrial slicing," *in IEEE TNSM*, vol. 20, no. 2, pp. 1660 – 1675, Jun. 2023.

[3] J. Sengupta, S. Ruj, and S. Das Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *in Journal of Network and Computer Applications*, vol. 149, p. 102481, Jan. 2020.

[4] M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-lstm network," *in Symmetry*, vol. 11, no. 4, p. 583, Apr. 2019.

[5] O. Abdul Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *in IEEE COMST*, vol. 23, no. 2, pp. 1342 – 1397, Secondquarter 2021.

[6] Y. Lai, J. Zhang, and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *in Security and Communication Networks*, vol. 2019, pp. 1–11, Sept. 2019.

[7] C. Benzaid, T. Taleb, and J. Song, "AI-based Autonomic & Scalable Security Management Architecture for Secure Network Slicing in B5G," *IEEE Network Magazine*, vol. 36, no. 6, pp. 165 – 174, Nov./Dec. 2022.

[8] M. J. Islam, A. Rahman, S. Kabir, M. R. Karim, U. K. Acharjee, M. K. Nasir, S. S. Band, M. Sookhak, and S. Wu, "Blockchain-sdn-based energy-aware and distributed secure architecture for iot in smart cities," *in IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3850–3864, Mar. 2021.

[9] O. Hireche, C. Benzaïd, and T. Taleb, "Deep data plane programming and ai for zero-trust self-driven networking in beyond 5g," *in Computer Networks*, vol. 203, p. 108668, Feb. 2022.

[10] Y. Liu, B. Zhao, P. Zhao, P. Fan, and H. Liu, "A survey: Typical security issues of software-defined networking," *in China Communications*, vol. 16, no. 7, pp. 13–31, Jul. 2019.

[11] M. L. Adjou, C. Benzaïd, and T. Taleb, "Topotrust: A blockchain-based trustless and secure topology discovery in sdns," in *in Proc. 2022 International Wireless Communications and Mobile Computing (IWCMC)*, Dubrovnik, Croatia, Jul. 2022.

[12] A. Derhab, M. Guerroumi, M. Belaoued, and O. Cheikhrouhou, "Bmc-sdn: blockchain-based multicontroller architecture for secure software-defined networks," *in Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–12, Apr. 2021.

[13] A. Bose, G. S. Aujla, M. Singh, N. Kumar, and H. Cao, "Blockchain as a service for software defined networks: A denial of service attack perspective," in *in Proc. 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress*, Fukuoka, Japan, Aug. 2019.

[14] A. Derhab, M. Guerroumi, A. Gumaei, L. Maglaras, M. A. Ferrag, M. Mukherjee, and F. A. Khan, "Blockchain and random subspace learning-based ids for sdn-enabled industrial iot security," *in Sensors*, vol. 19, no. 14, p. 3119, Jul. 2019.

[15] "Industrial control system (ics) cyber attack datasets," in *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*, 2022. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[16] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A review of research work on network-based scada intrusion detection systems," *in IEEE Access*, vol. 8, pp. 93 083–93 108, May 2020.

[17] B. Alotaibi and M. Alotaibi, "A stacked deep learning approach for iot cyberattack detection," *in Journal of Sensors*, vol. 2020, pp. 1–10, Sept. 2020.

[18] W. Gao, "Cyberthreats, attacks and intrusion detection in supervisory control and data acquisition networks," Mississippi, MS, USA, Dec. 2013.

[19] A. Al Hayajneh, M. Z. A. Bhuiyan, and I. McAndrew, "Improving internet of things (iot) security with software-defined networking (sdn)," *in Computers*, vol. 9, no. 1, p. 8, Feb. 2020.

[20] A. K. Nair and J. N. D. J. Jingle, "Distributed denial-of-service detection and mitigation using software-defined network and internet of things," vol. 11, no. 1, p. 10, Sept. 2019.

[21] K. Haseeb, I. Ahmad, I. I. Awan, J. Lloret, and I. Bosch, "A machine learning sdn-enabled big data model for iomt systems," *in Electronics*, vol. 10, no. 18, p. 2228, Sept. 2021.

[22] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *in IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[23] T. H. Morris, Z. Thornton, and I. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," *in Proc. 7th annual southeastern cyber security summit*, Alabama, USA, Jun. 2015.