applied sciences

MDPI

Article

A Secured Industrial Internet-of-Things Architecture Based on Blockchain Technology and Machine Learning for Sensor Access Control Systems in Smart Manufacturing

Hichem Mrabet [1], Adeeb Alhomoud [2], Abderrazek Jemai [3] and Damien Trentesaux [4,*]

[1] SERCOM Laboratory, Tunisia Polytechnic School, University of Carthage, B.P. 743, Tunis 2078, Tunisia; hichem.mrabet@gmail.com
[2] Department of Science, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh 11673, Saudi Arabia; a.alhomoud@seu.edu.sa
[3] INSAT, SERCOM Laboratory, Tunisia Polytechnic School, University of Carthage, B.P. 743, Tunis 1080, Tunisia; abderrazak.jemai@insat.rnu.tn
[4] LAMIH-UMR CNRS, Université Polytechnique Hauts-de-France, 59313 Valenciennes, France
* Correspondence: damien.trentesaux@uphf.fr

Featured Application: A potential application of the work concerns the development of secure IIoT architectures for smart manufacturing using blockchain technology and machine learning algorithms.

Abstract: In this paper, a layered architecture incorporating Blockchain technology (BCT) and Machine Learning (ML) is proposed in the context of the Industrial Internet-of-Things (IIoT) for smart manufacturing applications. The proposed architecture is composed of five layers covering sensing, network/protocol, transport enforced with BCT components, application and advanced services (i.e., BCT data, ML and cloud) layers. BCT enables gathering sensor access control information, while ML brings its effectivity in attack detection such as DoS (Denial of Service), DDoS (Distributed Denial of Service), injection, man in the middle (MitM), brute force, cross-site scripting (XSS) and scanning attacks by employing classifiers differentiating between normal and malicious activity. The design of our architecture is compared to similar ones in the literature to point out potential benefits. Experiments, based on the IIoT dataset, have been conducted to evaluate our contribution, using four metrics: Accuracy, Precision, Sensitivity and Matthews Correlation Coefficient (MCC). Artificial Neural Network (ANN), Decision Tree (DT), Random Forest, Naive Bayes, AdaBoost and Support Vector Machine (SVM) classifiers are evaluated regarding these four metrics. Even if more experiments are required, it is illustrated that the proposed architecture can reduce significantly the number of DDoS, injection, brute force and XSS attacks and threats within an advanced framework for sensor access control in IIoT networks based on a smart contract along with ML classifiers.

Keywords: Blockchain; industrial IoT; smart manufacturing; security threats; security solutions; machine learning; classifiers; privacy; smart contract; access control

Citation: Mrabet, H.; Alhomoud, A.; Jemai, A.; Trentesaux, D. A Secured Industrial Internet-of-Things Architecture Based on Blockchain Technology and Machine Learning for Sensor Access Control Systems in Smart Manufacturing. *Appl. Sci.* **2022**, *12*, 4641. https://doi.org/10.3390/app12094641

Academic Editors: Howon Kim and Thi-Thu-Huong Le

Received: 25 March 2022
Accepted: 2 May 2022
Published: 5 May 2022

1. Introduction

The smart manufacturing is a "form of production integrating manufacturing assets of today and tomorrow with sensors, computing platforms, communication technology, control, simulation, data intensive modelling and predictive engineering" [1]. The development of smart manufacturing is powered by technological advances relevant to the fourth industrial revolution (Industry 4.0) [2,3]. Among these technologies, the Industrial Internet-of-Things (IIoT) is receiving more attention from the researchers' community due to its ability to integrate new technologies such as sensors, radio frequency identification (RFID) and communication protocols. The supervisory control and data acquisition (SCADA)

system is one example of a system based on the IIoT including sensors and actuators controlled by programmable logic controllers (PLCs) [4]. However, the integration of these new digital technologies will lead to an increased number of vulnerabilities and attacks in the smart manufacturing context [5].

In that context, Machine Learning (ML) and Blockchain technology (BCT) can be viewed as a potential solution to reduce the new vulnerability introduced in IIoT fields for smart manufacturing. On the one hand, ML, which is a subfield of artificial intelligence dealing with the development of algorithms, can improve automatically through experience and by the use of data [6]. There already exists ML-based security solutions for IIoT [4,7,8]. ML is used in smart manufacturing and IIoT to improve decisions and has therefore become one key element of smart manufacturing [9].

On the other hand, the use of BCT for the smart manufacturing, which is a technology based on recording transactions among participants implemented via blocks inside databases, can be fruitful, leading to several benefits such as automation process, traceability, data integrity and sustainability. One challenge is the integration of IoT-related technology by considering the huge diversity of communication protocols, IoT device variety and the big data exchanged among various IoT applications. Moreover, there is no common model to support the IoT hardware and software diversity in the context of smart manufacturing [10]. Likewise, ML algorithms can be integrated in models based on BCT for data analysis and prediction to enforce security and for attacks detection as well.

The aim of this work is thus to propose a secure IIoT architecture including BCT enforced with ML algorithms for smart manufacturing applications. The proposed architecture is based on a previous work [11] developed in the specific context of IoT networks. Our architecture is compared to similar ones in the literature [12–14] in terms of number of layers, advantages, disadvantages and security issues. Through experimentations, it is demonstrated that our architecture can reduce the number of attacks and can mitigate threats thanks to the BCT inherent features for sensors' access control. Our experiments have been realized in two steps. Firstly, a scenario is carried out as a function of ML performance metrics. This scenario is led without considering BCT. It is based on a data-driven study and a classification of threats using various kinds of classifiers by considering a dataset for Industry 4.0 [15]. Secondly, an advanced scenario, considering a Blockchain (BC) data structure enforced with ML is used through the definition of a framework. The results are then compared to the ones of the first scenario to show the added value of the proposed architecture to mitigate the number of attacks in IIoT networks.

This paper is organized as follows. First, Section 1 presents state-of-the-art related works. Section 2 details the proposed architecture based on BCT and ML tools by considering the limitations in the literature. Then, in Section 3, a comparative study is led, and a discussion is suggested to position and evaluate the proposed architecture with relevant works in the state-of-the-art. A validation and data analysis with ML is investigated in Section 4 by considering two introduced scenarios. In Section 5, a discussion is made to highlight the main results and to point out the limitations of our work as a starting point to improve the system performance to tackle attacks over IIoT in the near future. Finally, a conclusion is drawn and prospects are identified.

2. State-of-the-Art: A Review

As introduced, BCT is based on recording transactions among participants implemented via blocks inside databases. Blocks are linked via a Hash function to preserve data integrity. The creation of a new block is enforced with proof of work (PoW) or proof of stake (PoS) mechanisms requiring the node to solve a cryptographic puzzle. The append-only ledger technology related to the BCT was initially proposed for the cryptocurrency systems, for instance Bitcoin. The concept of BCT was proposed in 2008 [16], and it includes three main types such as private, public and federated BC [17]. It has attracted much attention over the past years as an emerging peer-to-peer technology for distributed computing and traceable and decentralized data sharing. Due to the integration of cryptography

technology with a decentralized control and decentralized data storage, BCT can avoid attacks aiming to take control over the system. Later, in 2013, Ethereum, a transaction-based state-machine, was presented to implement the BCT [18]. Indeed, the latter is defined as a public and decentralized BC offering smart contract implementation. In addition, a smart contract is a program composed of code, data and rules that can be manipulated by a user account to send transactions over network. In this work, a smart contract data structure for sensor access control and a smart contract code for authentication process and rules are proposed in the context of smart manufacturing applications.

Due to its unique and interesting features, such as transactional privacy, security, the immutability of data, auditability, integrity, authorization, system transparency, and fault tolerance, BCT is an emergent technology applied in a constantly growing set of various fields. Typically, some of these fields are intelligent transportation [19], supply chain management [20], agriculture [21], Industry 4.0 [10,22], 3D printing [23], protecting museum-digital property rights [24] and Internet of Energy (IoE) [12,25,26]. Table 1 contains the state-of-the-art works based on BCT/ML for IIoT networks, considering smart manufacturing applications. The keywords used leading to Table 1 results were "blockchain", "Industry 4.0", "machine learning", "security", "smart contract" and "smart manufacturing". The results were gathered from Elsevier, IEEE, Springer, Google Scholar and MDPI databases during 2018–2022.

Table 1. Related works based on Blockchain/ML for Industrial IoT.

Years	Authors	Focus
2018	Gao et al. [12]	Authors have proposed a monitoring system based on smart contract to identify malicious usage of electrical power.
2018	Li et al. [22]	Authors have introduced the energy BC based on the consortium BCT and the Stackelberg game.
2018	Aitzhan et al. [26]	Authors have implemented a token-based private decentralized energy trading system.
2018	Lin et al. [27]	A four-layer framework based on smart contract BCT for fine-grained access control system is proposed for Industry 4.0.
2019	Dai et al. [13]	Authors have proposed a four-layer architecture for the concept of Blockchain of Things (BCoT) in industrial applications.
2019	Zhao et al. [14]	Authors have proposed a new architecture based on smart contract for client and resource registrations in IIoT.
2019	Liang et al. [25]	Authors have proposed a data protection framework based on distributed BC.
2019	Tanwar et al. [28]	Authors have proposed a hybrid technique based on BC and ML to detect attacks for energy-trading applications.
2020	Jameel et al. [29]	Authors have proposed a reinforcement learning technique to address the block time minimization and transaction throughput of blockchain-based IIoT networks.
2021	Shahbazi et al. [15]	Authors have proposed a new architecture based on smart contract, BC and ML for quality control in smart manufacturing application.
2021	Javaid et al. [30]	Authors have proposed BCT applications for Industry 4.0 such as manufacturing data protection, automotive, information and security.
2021	Rathee et al. [31]	An IIoT framework based on BC for sensor authentication is proposed by the authors.
2021	Shrivastava et al. [32]	The paper exposes the main enabling technologies for Industry 4.0 such as IoT, Artificial Intelligence, cloud computing and BC.
2021	Leng et al. [33]	Authors present the various metrics for the usage of BC in manufacturing.
2021	Faridi et al. [34]	Authors propose BC and IoT-based product traceability system for textile manufacturing applications.
2022	Chen et al. [35]	The review paper discusses the BC applications in Industry 4.0 such as authentication, asset tracking and smart contract exchange.

In 2018, GridMonitoring, a monitoring system, was introduced by Gao et al. [12], based on the Smart Grid. It ensures transparency, provenance, and immutability. In addition, the proposed system is based on smart contract to identify the malicious usage of electrical power by reporting any violations into a database in the context of a smart grid network.

Li et al. [22] introduced the energy BC for secure energy trading in Industrial IoT based on the consortium BCT and the Stackelberg game. Aitzhan et al. [26] implemented a token-based private decentralized energy trading system in decentralized smart grid energy, which can be applied to the IoE. Therefore, the IoE provides an innovative concept to increase the visibility of energy consumption in the Smart Grid. The authors in [27] propose a framework named BSeIn that is a BCT-based system for secure mutual authentication to enforce fine-grained access control polices. Furthermore, BSeIn ensures confidentiality, privacy and multifactor authentication.

Dai et al. [13] proposed a new architecture based on four layers for industrial applications. In addition, the authors define the concept of merging BC and IoT in one system named Blockchain of Things (BCoT). Likewise, a new architecture based on three layers for supplier chain applications among supplier, producer, factor and customer is suggested by Zhao et al. [14]. The latter architecture is based on a distributed ledger BC to ensure traceable transactions among untrusted peers. In modern power systems, Liang et al. [25] proposed a data protection framework based on distributed Blockchain, which can resist against data manipulation that is triggered by cyber attackers (e.g., false data injection attacks). To guarantee data accuracy, Liang et al. [25] created a framework that uses the consensus mechanism, which is automatically implemented by every node and has the representative characteristics. Additionally, the authors in [28] proposed a hybrid technique based on BC and ML to detect attacks for energy trading applications. In addition, the authors suggested in [28] to store datasets used by ML models into a BC network in order to reduce data errors such as duplication, missing data value and noise.

The authors in [29] suggested a reinforcement learning (RL) technique to address the block time minimization and transaction throughput of BC-based IIoT networks. The authors discussed how to obtain a low bias training of the agent in the RL and proposed some alternative solutions such as Q-learning, multi-armed bandit learning, actor–critic learning and deep policy optimization techniques. Applications of RL techniques in BC-based IIoT networks were discussed. They concluded that the Q-learning technique is appropriate for improving transaction throughput and minimizing forking events. Actor–critic learning and deep Q-learning techniques were discussed as a possible means to improve the energy efficiency of IoT devises. Q-learning and multi-armed bandit learning were also intended to minimize the time to finality and reduce the block time. Finally, deep Q-learning could be applied by adding an artificial noise in the network to protect broadcast/acknowledgement messages exchanged among IIoT-BC devices.

Shahbazi et al. [15] proposed a new architecture based on smart contract, BC and ML for quality control in smart manufacturing. In addition, the authors suggested a BC distributed ledger to store an information-related automation contract among the supplier, manufacturer and retailer. Likewise, an architectural diagram of the predictive analysis based on XGBoost (a flexible and highly efficient algorithm that can avoid overfitting issues) was proposed by the authors in [15] covering data pre-processing, feature selection and dividing the dataset into training and testing sets.

Javaid et al. [30] presented a review on BCT applications for Industry 4.0. Likewise, the authors exhibit the capabilities of BCT implementation in several industrial fields involving the healthcare domain, education services, logistics and transportation and government sectors.

The authors in [31] propose an IIoT framework based on BC for sensors authentication. Therefore, a performance analysis is performed by measuring the probability of attack success, a created hazard by the intruder and authentication accuracy metrics versus the number of IoT devices via an NS2 simulator.

Leng et al. [33] present the various metrics for the usage of BC in manufacturing including data-temper resisting, data-provenance, decentralized decision, collaborative optimization, system flexibility, cost saving, system sustainability, system resilience, network transparency and reputation improvement.

Finally, Chen et al. [35] discuss the BC applications in Industry 4.0 such as authentication, asset tracking and smart contract exchange. In addition, BC concerns are exhibited

by the authors in terms of higher energy consumption, large storage capacity and the vulnerability of the last block against attacks.

We provide hereinafter a more in-depth analysis of three of these contributions that focuses specifically on BCT applied to IIoT. Their potential benefits and limits to counteract cyber-attacks in the context of smart manufacturing are studied. The choice for these three contributions was motivated by the will to establish a fair comparison between our work and similar ones in terms of architectural design (i.e., number of layers and the functionality of each layer), smart contract data structure and application of smart contract in the manufacturing field.

The first one is the GridMonitoring system [12], which is based on four layers such as the user layer, data processing and monitoring layer, registration and authentication layer and energy and data center layer. The system integrates an implementation of a smart meter for transparency purposes between the customer and electricity company provider. The benefit of the proposed solution, besides the usage of BCT, is the employment of authentication mechanisms and data centers to save all information to report any violations. Nonetheless, the solution applies a classical smart contract database to report any violations that can be the object of an injection or/and XSS (cross-site scripting) attack. The solution can be enhanced with an ML-based solution to prevent and detect injection and XSS attacks.

The second one is the architecture proposed by Dai et al. [13], which is presented in Figure 1. As depicted, the architecture is composed of perception, communication, BC-composite and industrial applications layers.

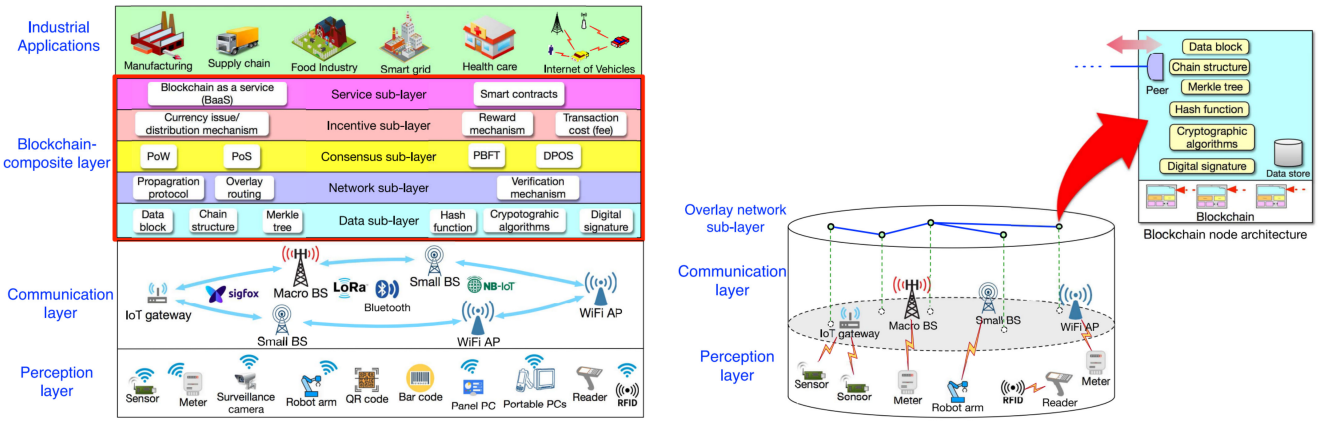

Figure 1. The architecture proposed by Dai et al. [13].

The BC composite layer covers five sub-layers including data, network, consensus, incentive and services sub-layers. The data sub-layer includes a data block, chain structure, Merkle tree, Hash function, digital signature and cryptographic algorithms. The consensus sub-layer contains the architecture-supported consensus algorithms such as PoW, PoS, and practical Byzantine fault tolerance (PBFT). This architecture presents the advantage of a well-structured composite BC sub-layer including all components of BC node architecture. However, merging data and network in one layer can lead to a serious privacy violations in case of man-in-the-middle (MitM) attacks. The potential benefits of this approach in the context of smart manufacturing can be applied in supply chain, food industry, smart city, healthcare and Internet-of-Vehicles (IoV) applications. Meanwhile, some limitations can be found concerning the fact that the data block, chain structure and network tools are merged in one BC composite layer.

The third studied architecture was proposed by Zhao et al. [14] and is presented in Figure 2, considering an application to the supply chain. As shown in Figure 2, the proposed architecture is based on three layers incorporating BC, data and governance layers. In the proposed model, all transactions including custom payment, producer registration, supplier registration, factor registration, contract term condition and refund are recorded

inside the BC data block structure. In addition, a Merkle tree is employed to save the hash function for every transaction.

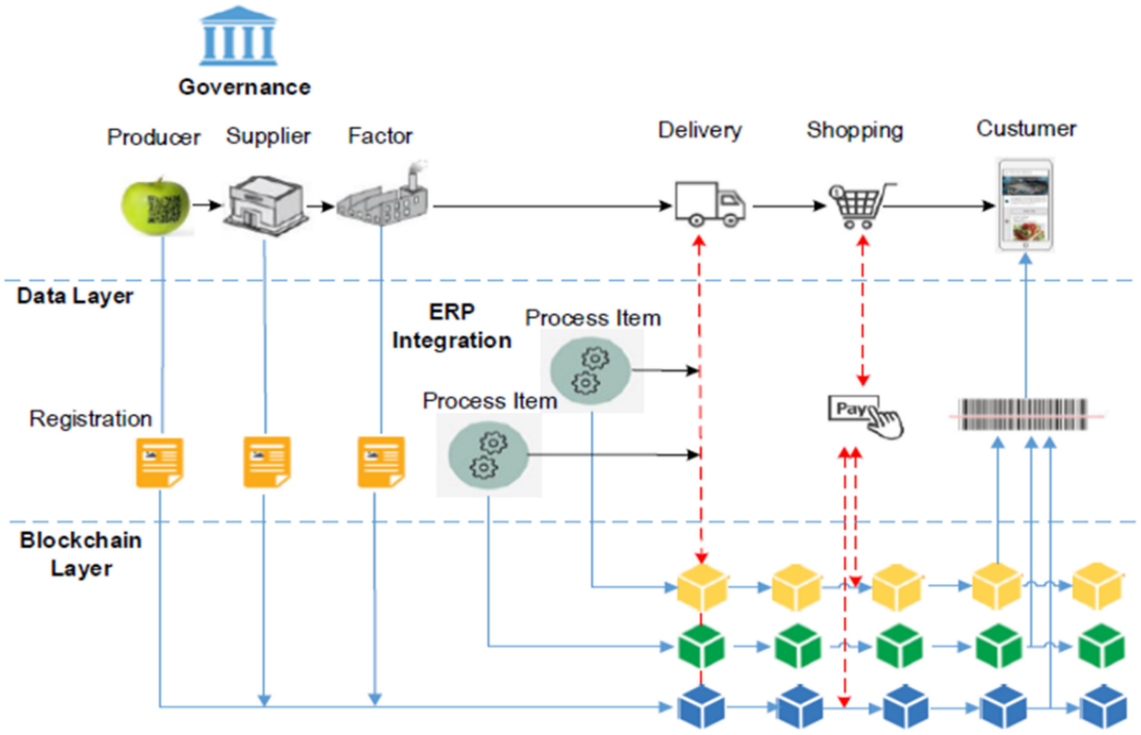

Figure 2. The architecture proposed by Zhao et al. [14].

The latter architecture presents a reduced number of layers (i.e., three), but a high number of registrations is observed in the BC data structure for the producer, supplier, factor, delivery, shopping and customer, respectively.

Our review led us to point out some limitations of the existing introduced state-of-the-art in the context of the smart manufacturing. These limitations concern mainly the lack of security mechanisms ensuring a sufficient protection level for sensing and control functions facing for example DoS (Denial of Service), DDoS (Distributed Denial of Service), injection, XSS and brute force attacks. For instance, the availability security metric is not preserved for [12] and the privacy security metric is not ensured for [12–14] as well.

Our aim is to propose a complete architecture based on a decentralized BC data structure along with ML for securing smart manufacturing sensing and control functions. To the best of our knowledge, this is the first time ML and BCT are suggested to reach this objective for an IoT sensor access control system in the context of smart manufacturing applications.

3. The Proposed Architecture

The contribution presented in this paper consists of the extension of a previous work [11], where an architecture to secure IoT applications was suggested, based on five layers such as physical sensing, network/protocol, transport, application and data and cloud services. This architecture exhibits the security threats and attacks against the IoT network and proposes security solutions based on the classification of the proposed layers.

The extended architecture considers the integration of BC components to ensure data integrity related to the sensors access control system. As shown Figure 3, this extended architecture contains five layers including a physical sensing layer, protocol network layer, BC tools and transport layer, application layer, ML, BC data structure and cloud service layer. The different layers are discussed hereinafter.

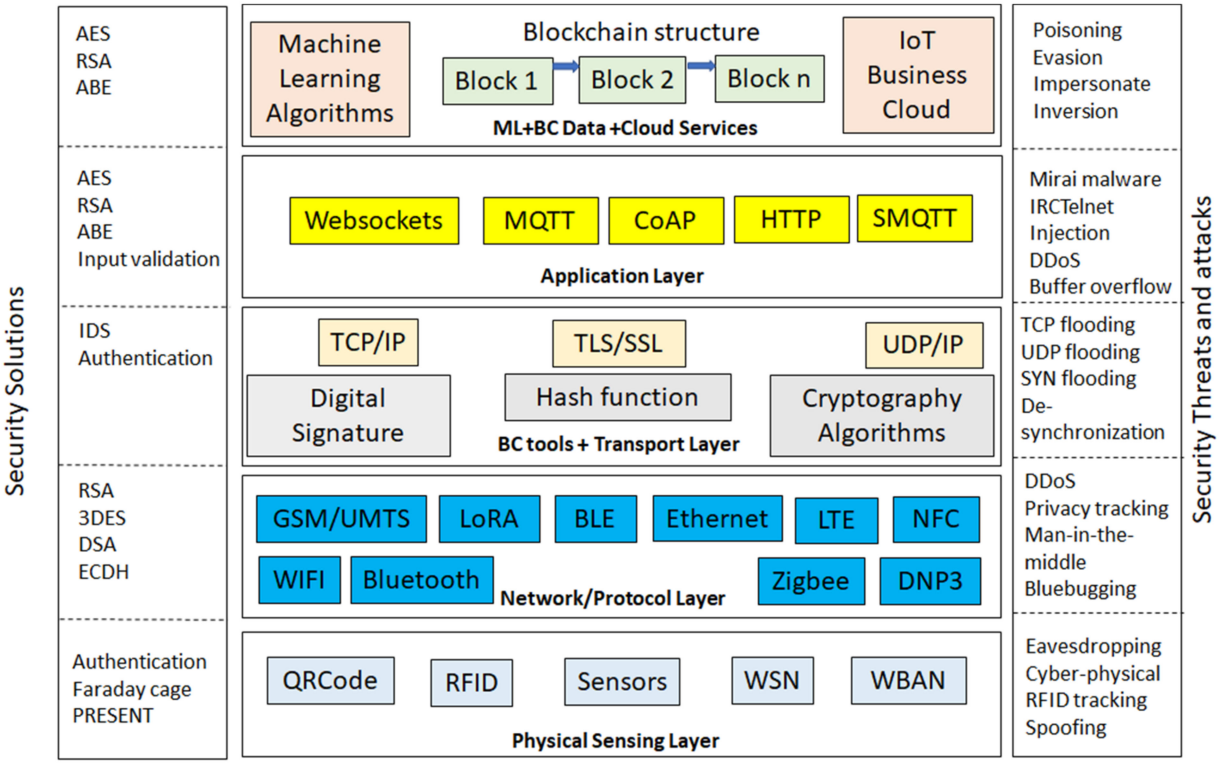

Figure 3. The proposed secured IIoT architecture, considering BCT and ML components.

3.1. Physical Sensing Layer

The physical sensing layer contains physical devices such as sensors, a wireless sensor network (WSN), a wireless body area network (WBAN), QRCode and RFID components. The security threats related to the physical layer are eavesdropping, cyber-physical attacks, jamming, RFID and spoofing. Authentication, Faraday cage and PRESENT are the effective security solution to preserve IIoT networks from physical sensing threats.

In the proposed architecture, we have classified the common attacks (in the right side) in the IIoT network according to the corresponding layer. For example, the spoofing attack could target a sensor in the physical layer. On the other hand, the security solutions are classified according to the corresponding layer. For instance, the authentication mechanism is a solution against a spoofing attack.

3.2. Network and Protocol Layer

The network and protocol layer supports most of the communication protocols used by IIoT networks such as Bluetooth Low-Energy (BLE), Ethernet, Long-Term Evolution (LTE), LoRA (Long Range), Near Field Communication (NFC), WiFi, Bluetooth, Zigbee, DNP3 ModBus, and GSM/UMTS. LoRa is an efficient energy radio communication system effective in IoT configurations. The security threats and attacks toward the network and protocol layer are DDoS, privacy tracking, MitM and bluebugging. Therefore, Rivest–Shamir–Adleman (RSA), triple data encryption standard (3DES), digital signature algorithm (DSA) and elliptic curve Diffie–Hellman (ECDH) are the most important security solutions that could be implemented to disable the security threats related to the network and protocol layer.

3.3. Blockchain Tools and Transport Layer

This layer integrates on the one hand the BC tools and the transport layer on the other hand. The latter incorporates common transport protocols such as TCP/IP and UDP/IP and TLS/SSL protocols. A lightweight BCT sub-layer equivalent to the Hyperledger Composer

component is added to verify and control the access to assets by the legitimate owner. Digital signature, Hash function, and cryptography algorithms are added to the BC tools and transport layer for enforcing an access control verification and validation process. In addition, in order to prove the identity of the sender, a digital signature (i.e., message digest or hash value) is generated by the sender private key and added to the sender authentication request to the desired sensor. Likewise, a digital signature is proposed at the level of the BC tools and transport layer to ensure data integrity and nonrepudiation.

3.4. Application Layer

The application layer covers the application protocols used in the IIoT context such as websockets, Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Hypertext Transfer Protocol (HTTP) and Secured Message Queuing Telemetry Transport (SMQTT). Mirai malware, IRCTelenet, Injection, DoS, DDoS and buffer overflow are the main threats regarding the application layer. Advanced encryption standard (AES), RSA, Attribute-Based Encryption (ABE) and input validation are the key security solutions to tackle the application layer threats.

3.5. Advanced Service Layer

Finally, the upper layer of our architecture, named the advanced service layer including ML, BC data and cloud services, is composed of the ML algorithms, the BC structure and the IIoT business clouds services. The BC structure is equivalent to Hyperledger Fabric and is used to save the access control information related to all assets in the IIoT networks for new creation. A new block is created for a new device or a change of the owner device or any changes regarding asset authorization by the legitimate owner.

Smart contract is defined as a finite number of statements with logical condition. Once the predefined conditions are fulfilled specified actions have to take place. In the literature, several works consider smart contract in an access control system for Internet service providing [36], IoT applications [37], and energy systems [38], respectively. Our system is based on a smart contract to ensure the users authentication process to sensors and to identify any malicious usage by reporting any violations into a database in the context of smart manufacturing. The advantage of the proposed solution, beside the usage of BCT, is the employment of an authentication mechanism that has a role to save all information and report any violations. Nonetheless, the solution applies a classical smart contract database and associated functions for authentication control (for example, canRead() and authenticated()) to report any violations that can be the object of a brute force or/and DDoS attack. The solution can be enhanced with an ML-based framework to prevent and detect a brute force attack and DDoS attack as well.

A sample of a smart contract data structure for sensor access control is presented in Figure 4 in which a sensor structure is composed of *SensorID* (defined through a Mac Address format), *OwnerID* (defined as the identification of the owner), *AuthenticationPwd* (i.e., the Hash of the user password is stored in the BC), *SensorStatus* (defined as the status of the sensor with two values: active or not active), *SensorValue* (used to save the last sensor value), *AuthorizedPerson* (used to save the authorized persons to manipulate the sensor with a limit of 10), and the *CommandLocation* structure used to detect a DDoS attack involving *CommandName*, *LocationForSameCommand* and *CommandTimeStamp*. A first rule stating that once the same *CommandName* is launched from three different locations (i.e., *LocationForSameCommand*) in a short period of time (according to *CommandTimeStamp*), then a DDoS attack is defined. A second rule statement is defined when three wrong consecutives *userpwd* in a short period of time specified in *CommandTimeStamp* happens, which is then considered as a brute force attack.

```
Contract SensorAccessControl {
Struct Sensor {
MacAdr SensorID;
Integer OwnerID;
String AuthenticationPwd;
Boolean SensorStatus;
String SensorValue;
Integer AuthorizedPerson[10];
Struct CommandLocation {
          String Command;
          String LocationForSameCommand[3]
          Integer TimeStamp;}
Integer Numberoftentativewithwrongpassword;
}
}
```

Figure 4. Smart contract sample data structure for sensor access control.

As depicted Figure 5, a smart contract code for the authentication process is exhibited to illustrate the condition for a successful authentication and a failed one.

```
Smart contract code
Function main() {
If (sensorID == SensorID and userID in the range of
(AuthorizedPerson[10]) and userpwd == AuthenticationPwd)
Then return "authentication successful"
Else if (userpwd != AuthenticationPwd)
return "authentication failed"
SensorID.Numberoftentativewithwrongpassword++
}
```

Figure 5. Smart contract code for authentication process.

In the current work, BC is applied to save the relevant information for sensor access control in the IIoT context. Once the sensor access control information is saved in the BC, we guarantee the data confidentiality, integrity and traceability thanks to the BCT inherent features. On the other hand, ML is used to detect any attack from a traffic packet crossing the IIoT network based on the information stored in the BC to prevent any sensor access control violations. In order to understand the architecture design of the paper, we suppose a coming packet where a user is asking for an authentication on a sensor in the IIoT network. We suppose that we have a client/server application to authorize or deny sensor access. Since all the sensors information related to the authentication process are saved on the BC, a client request is sent from the sensor (physical layer) to the BC (data and cloud layer) via a communication protocol (for instance, LoRA in network/protocol layer) passing through a TCP/IP connection (in the BC tools and transport layer, which is responsible for adding a digital signature to the client request) to a web application running under

HTTP protocol (in the application layer) containing the sensor identification (sensorID), user identification (userID) and user password (userpwd). Once the application server received the client request, it will check the correctness of the received information with the BC data structure (i.e., the sent sensorID should match the sensorID in the blockchain, the userID needs to match a value in the AuthorizedPerson table and userpwd should match AuthenticationPwd). In the case where all sent information are correct, the application server will authorize the client access to the sensor. According to Figure 5, in the case where the sent password (i.e., userpwd) is different from the stored password in the BC (i.e., AuthenticationPwd), the smart contract algorithm will increment the variable Numberoftentativewithwrongpassword related to the sensorID and the application server will deny the access to the sensor. The latter variable will serve as a feature in the ML process to detect a brute force attack. Moreover, CommandLocation in the BC structure will serve as a feature in the ML process to detect a DDoS attack.

By employing BCT components such as digital signature, Hash function and cryptography algorithms in the IIoT networks, various attacks such as eavesdropping, MitM, poisoning and evasion attacks can be de-activated by exploring inherited features provided by the BCT components such as the tractability and integrity. In addition, the Hash function is used to link between the BC nodes to ensure a high traceability when a new block is added at the level of the advanced service layer, so that injection attacks cannot be performed or may be reduced by intruders. Finally, a digital signature based on a message digest can provide authenticity, nonrepudiation and message integrity.

4. Comparison of Our Design Approach with the Existing Literature: Potential Benefits

In this section, a comparison study is performed among the architectures based on BCT and ML in the literature and the one proposed in this paper from a design perspective. Table 2 summarizes this comparison.

Table 2. Proposed architecture based on BCT/ML vs. literature: a comparison of design approaches.

Architecture	Description	Smart Contract Data Structure	Design Approach: Discussion
Latif et al. [8]	The authors propose an IIoT system architecture based on four layers such as physical, network, middleware and application layers.	Not applicable.	Authors suggest a solution based on ML to detect DDoS, malware, and advanced persistent threats in the edge and fog computing. The architecture can be enforced with BC to provide traceability and integrity.
Gao et al. [12]	The GridMonitoring system is based on four layers for smart grid network applications.	Smart contract is employed for the recording of the violations on the smart meter, data on the smart grid network and the state of the smart meter.	The benefit of the proposed solution beside the BCT is the usage of an authentication mechanism and data center to save all information to report any violations. The solution employed a classical smart contract database to report any violations that can be the object of an injection attack. The solution can be enhanced with an ML-based solution to prevent injection and XSS attacks.
Dai et al. [13]	The architecture is composed of perception, communication, Blockchain-composite and industrial applications layers.	The survey paper exhibits the life cycle of smart contracts including creation, deployment, execution and completion.	The architecture is highly structured. BC-composite layer includes data, consensus and network sub-layers. Merging data and network in one layer can lead to serious privacy violations in case of MitM attacks.

Table 2. *Cont.*

Architecture	Description	Smart Contract Data Structure	Design Approach: Discussion
Zhao et al. [14]	The proposed architecture is based on three layers incorporating BC, data and governance layers.	A smart contract is used for both client and resource registrations.	The number of layers is optimized. The solution lacks a cloud data storage to recover data from disaster situations.
Shahbazi et al. [15]	The suggested architecture is based on a smart contract between manufacturer and supplier for quality control applications in smart manufacturing.	The proposed BC data structure is used to store manufacturer and supplier data.	The proposed architecture is composed of three layers such as sensor layer, smart contract layer and distributed ledger layer.
Zaidi et al. [37]	Authors propose an access control contract to control the request sent by subjects to IoT objects.	Authors propose three types such as object attribute management contract, subject attribute management contract and policy management contract.	ML is not suggested in the proposed architecture.
Our architecture	The architecture is based on five layers including physical sensing, network/protocols, Transport-BC, application and advanced service layers.	A smart contract data structure and algorithm are proposed for sensor access control system, DDoS and brute force attacks prediction.	The BC components are dispatched into two layers such as BC tools and transport layer and advanced service layer. On the one hand, BC tools and transport layer provides data integrity. On the other hand, advanced service layer offers data traceability.

The conclusion of this comparison is that our design approach outperforms other architectures in terms of decentralized architecture and the capability to detect the most dangerous attacks in IIoT networks such as DDoS, injection, brute force and XSS attacks by proposing a smart contract data structure and algorithm for sensor access control system. In addition, in our architecture, the BC tools and transport layer provides data integrity by offering a digital signature. Furthermore, the advanced service layer provides traceability by adding pertinent authentication data inside the BC.

Table 3 presents the comparison between our system and similar ones in the literature in terms of security metrics such as integrity, availability, immutability, confidentiality and privacy.

Table 3. Comparison between our system and similar one in terms of security metrics.

Security Metric	GridMonitoring [12]	Zhao [14]	BSeIn [27]	ABAC [37]	Our Proposal
Integrity	✓	✓	×	✓	✓
Availability	✓	×	×	✓	✓
Immutability	✓	✓	✓	✓	✓
Confidentiality	✓	✓	✓	✓	✓
Privacy	×	×	✓	✓	✓

In our model, integrity is considered in the BC tools and transport layer through a digital signature generated by the sender private key and added to the sender authentication request to the desired sensor. Availability and immutability are granted thanks to the inherent features of BCT. Confidentiality and privacy is preserved by using a symmetric key encryption for all messages transferred across the IIoT network.

The next section describes a two-step experimental study that is aimed to illustrate these potential benefits.

5. Validation and Data Analysis with ML

5.1. Selected Attacks

To test and validate our contribution, we selected and tested several types of attacks on IIoT networks. For that purpose, we based our selection on the work of Ferag et al. [39] who presented the major attacks on a BC/ML-based application for IoT networks. Therefore, Table 4 exhibits the set of selected attacks, which are structured with respect to the concerned layers of our architecture.

Table 4. Layer-based attacks in IIoT networks.

Layer	Attacks	Description
ML, Data and cloud services	Poisoning	Attack against ML via injecting adversarial samples to the training data in order to distort the model prediction
	Evasion	Samples are changed at the inferring phase to evade detection
	Impersonate	Prefers to imitate data samples from victims, in particular for application scenarios related to image recognition
Application	Injection	Untrusted data that are sent to an interpreter or database
	Brute force	An attempt to guess a password via sending various passwords
	Buffer overflow	This attack is targeting SCADA system and tries to overwrite a buffer to disrupt controller activity
	XSS	A kind of injection attack sent via a browser script
Transport	Flooding	Repeating the request of a new connection until the IIoT system reaches maximum level
	De-synchronization	Disruption of an existing connection
Network/protocol	DoS	Attempt to stop or reduce activity of an IIoT
	DDoS	A distributed DoS attack from several location
	MitM	Violating data confidentiality or integrity during transfer
	HELLO flood	Uses HELLO packets as weapon to launch the attack on IIoT system
Physical sensing	Eavesdropping	Deducing data sent by IIoT devices across network
	RFID tracking	Modifying a content of a tag or trying to disable it
	Jamming	Creating radio interference and exhaustion on IIoT devices

The upper layer concerns various attacks such as poisoning, evasion, impersonate and inversion attacks. The application layer incorporates the most important attacks against IIoT applications such as injection, brute force, buffer overflow and XSS attacks. Flooding and desynchronization are examples of transport layer attacks. Attacks against the network/protocol layer are considered as the bulky portion of attacks by including DoS, DDoS, MitM, HELLO flood and Sybil attacks. Finally, eavesdropping and RFID tracking jamming belong to physical sensing layer attacks.

5.2. First Scenario: Experimentation without BCT

We experimented with ML classifiers in order to enforce the proposed architecture by detecting attacks as a provided functionality by the upper advanced service layer without considering BCT in the first scenario. The performance metrics that were used are: accuracy, precision, sensitivity and MCC performed under a WEKA data-mining environment.

WEKA is an open source environment used by researchers worldwide to manipulate various kinds of ML algorithms such as classification, regression and clustering. In this work, WEKA 3.8 under Windows 64 bits is used to perform the experiments.

The TON_IoT dataset is used for attack detection in the context of IIoT. It is composed of 35975 instances, 127 attributes and considers attacks such as DoS, DDoS, injection, MitM, brute force, XSS and scanning attacks [40], as presented in Table 5. In this experiment, the employed dataset is generated by a testbed based on a virtual machine to represent the IoT networks and a KALI offensive system to represent hacker attacks. To gather the packet traveling inside the network to identify whether it is a normal activity or an attack, the Netsniff-ng and Zeek (Bro) tools are used. The experiment results have been carried out on a hardware characterized by a processor Intel Core i7 2.8 GHz and a 16 Go RAM.

The choice of the TON-IoT dataset is motived by grouping two mains features such as representing an IoT network by containing packet information traffic and gathering the most common attacks targeting manufacturing environment by simulating a hacker attacker with a KALI offensive system.

The TON_IoT dataset attributes are grouped into six categories such as processor information, process information, packet information, memory information, logical disk information and type and label for attack detection attributes:

- The attributes related to processor information include Processor_pct_User Time, Processor_pct_Processor_Time, Processor_pct_Privileged_Time and so on.
- The process information attributes cover Process_IORead_Operations_sec, Process_IO _WriteOperations_sec, Process_IO_Write_Bytes_sec and so on.
- The packet information attributes involve Network_I(IntelR_82574L_GNC)Packets ReceivedUnknown, Network_I(IntelR_82574L_GNC, Packets Outbound Errors, Network_I(Intel R _82574L_GNC), PacketsSentUnicastsec and so on.
- Memory information attributes are (but are not limited to) MemoryAvailable Bytes, Memory Cache Bytes and MemoryPage Faultssec.
- Logical disk information attributes contain LogicalDisk(_Total)pct_DiskReadTime, LogicalDisk(_Total)DiskWritessec and LogicalDisk(_Total)CurrentDiskQueue Length,

In our approach, a supervised ML based on attribute type and label to classify the various attacks was defined, according to Table 5.

During the experimentation, the training and testing phases while using a training set were performed. To illustrate the whole life cycle performed to carry out the predictive ML metrics, Table 6 highlights the time to build a model (in second) and the time to test a model on the training data (in second) for the various classifiers to classify the different attacks in an IIoT network.

On the one hand, as provided in Table 6, the ANN classifier requires the highest time for the training phase to build the ML model. However, NB takes less time to build the model. On the other hand, DT classifier presents the minimum time to test the model and NB presents the maximum testing time.

Figure 6 represents the partition of the normal activity and attacks among DoS, DDoS, injection, MitM, brute force, XSS and scanning attacks during training and testing phases.

To assess the performance of the model, metrics have to be defined without considering the data provided by the BC structure in this scenario. Indeed, the evaluation of the performance of a model informs us about the effectiveness of the predictions of a dataset by the trained model. For that purpose, a confusion matrix of a binary classification assessment is used. As shown in Table 7, the confusion matrix is a table showing the number of instances belonging to each of four categories (represented by TP, FP, FN and TN). The ML performance metrics used in this study are as follows:

- Accuracy defined as:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \times 100 \quad (1)$$

- Precision defined as:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \times 100 \quad (2)$$

- Sensitivity defined as the proportion of actual positives which are predicted positive:

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \times 100 \quad (3)$$

- Matthews Correlation Coefficient (MCC): defines the correlation between the predicted value and the observed one [4]. MCC measures the quality of a classifier to perform a classification task.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt[2]{(\text{TP} + \text{TN}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{FP} + \text{FN})}} \times 100 \quad (4)$$

Table 5. Type and label attributes for attack detection.

Type	Label	Description
0	normal	Normal activity
1	ddos	Distributed denial of service attack
2	dos	Denial of service attack
3	injection	Injection attack
4	mitm	Man in the middle attack
5	password	Brute force attack
6	xss	Cross-site scripting attack
7	scanning	Port scanning attack

Table 6. Required time for training and testing phases for the studied classifiers.

	ANN	DT	SVM	RF	NB	AdaBoost
Time to build the model (s)	3232.14	30.59	2.31	13.6	0.55	3.81
Time to test model on training data (s)	2.72	0.12	0.22	0.42	3.06	0.55

The different required steps to perform the various ML metrics are depicted in Figure 7. Therefore, the life cycle of ML evaluation metrics involves six steps: dataset selection, pre-processing, classifier model selection, model training, test phase and ML evaluation metrics. In the pre-processing step, a correlation-based feature selection (CFS) is employed with the best first search approach. Indeed, the CFS algorithm starts with an empty node and can go up to five nodes. In addition, subsets are evaluated with 10-fold cross-validation on the training dataset.

For instance, the different steps needed for simulation experiments to carry out predictive accuracy for the ANN classifier are defined as follows:

1. Upload the TON_IoT dataset.
2. Data pre-processing (a CFS is selected for the attribute evaluator and best first is selected for the search method).
3. Select the classifier type and configure the classifier parameters.
4. Model training.

5. Test phase and select the test options (cross-validation = 10-fold).
6. Run the simulation to evaluate predictive accuracy.

The parameters set on the WEKA environment for the different used classifiers in the experiments are the following.

- The name of the ANN classifier in the WEKA environment is called multilayer perceptron: a classifier using the backpropagation technique to classify instances. In our experiments, the multilayer perceptron classifier parameters were set as follows: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a. Here, the LearningRate (the learning rate for weight updates) is equal to 0.3, Momentum = 0.2, TrainingTime = 500, Validation threshold = 20, HiddenLayer (to define the hidden layers of the neural network) equal to a (attributes + classes)
- Regarding the decision tree classifier under the WEKA environment, REPTree classifier was selected and defined as a fast decision tree learner with the following setting: REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0. Here, numFolds (the specified number of folds of data used for pruning the decision tree) equals 3, minNum (minimum number of instances per leaf) equals 2 and minVarianceProp (the minimum proportion of the variance on all the data that need to be present at a node) equals 0.001.
- In the WEKA environment, the SVM classifier is called SMO and it implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. For SMO, the following options are active: SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4". Here, epsilon (the epsilon for round-off error) is set to 1.0E-12, SVM kernel (i.e., polynomial kernel) is set to -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" and calibrator (the calibration method to use) is set to "weka.classifiers.functions. Logistic -R 1.0E-8 -M -1 -num-decimal-places 4".
- The Random Forest classifier under WEKA was selected, which is defined as a class for constructing a forest of random trees with the following setting: RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1, where bagSizePercent (size of each bag, as a percentage of the training set size) equals 100, numIterations (the number of trees in the random forest) equals 100 and maxDepth (the maximum depth of the tree, 0 for unlimited) is equal to 0.
- Naïve Bayes classifier, a statistical classifier, has been adopted. It assumes that the values of attributes in the classes are independent. This assumption is called class conditional independence and it is based on Bayes' theorem. Under WEKA, the NB classifier is called Naive Bayes using estimator classes. Therefore, the Naive Bayes parameters were set as follows: useKernelEstimator (use a kernel estimator for numeric attributes rather than a normal distribution) is set to false, numDecimalPlaces (the number of decimal places to be used for the output of numbers in the model) is set to 2, and batchSize (the preferred number of instances to process if batch prediction is being performed) is set to 100.
- Finally, AdaBoost classifier is used for comparison purposes. It is defined as an adaptive boosting algorithm based on minimizing the exponential loss function. Therefore, the AdaBoost parameters in WEKA were set as follows: AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump, where the base classifier to be used is DecisionStump and batchSize is set to 100.

Figure 8 represents the comparison of the Accuracy of studied classifiers performed under a WEKA data-mining environment.

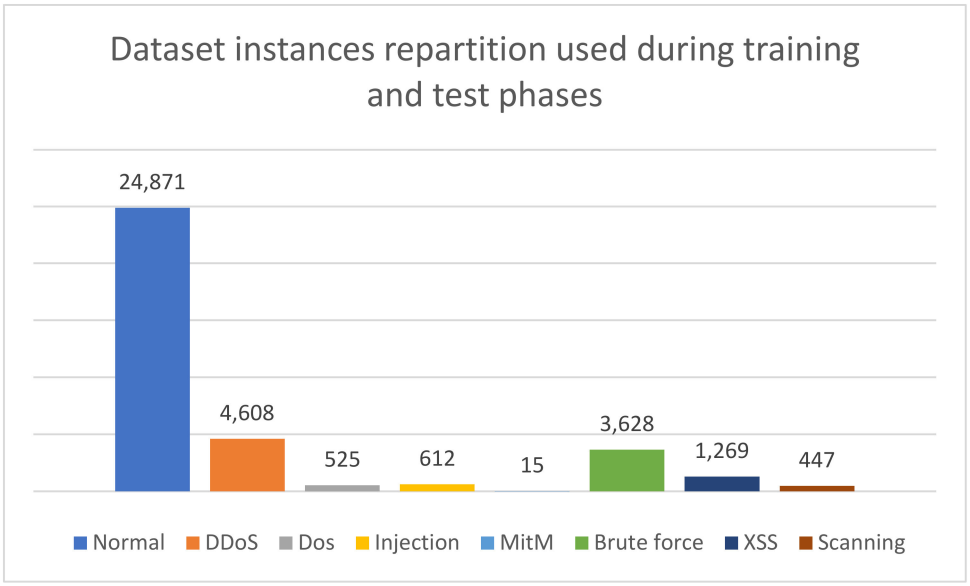

Figure 6. Dataset instances repartition used during training and test phases.

Table 7. Confusion matrix.

Confusion Matrix		Target	
		Normal	**Attack**
Model	Normal	True Positive (TP)	False Positive (FP)
	Attack	False Negative (FN)	True Negative (TN)

Figure 7. ML metrics evaluation life cycle.

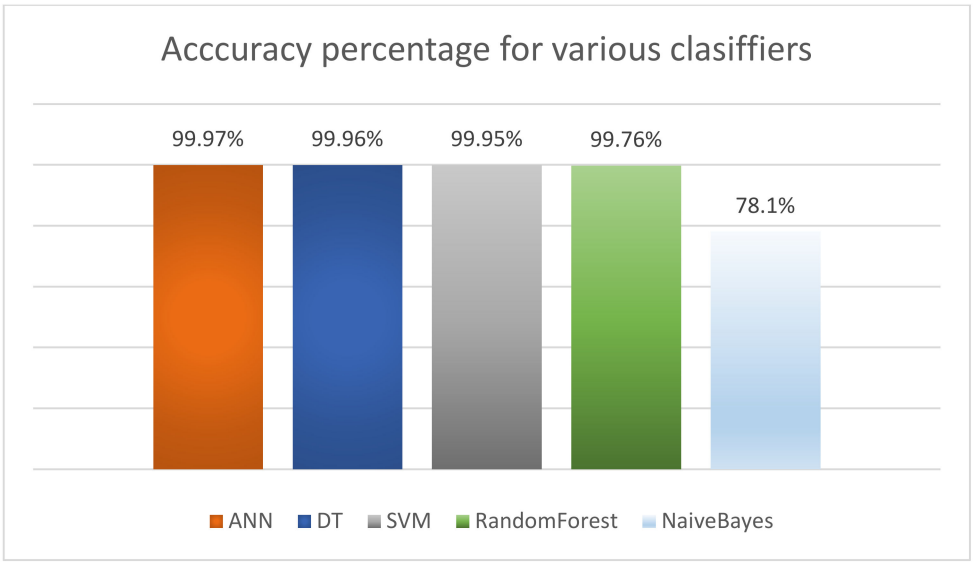

Figure 8. Accuracy percentage for studied classifiers.

As shown in Figure 8, the accuracy percentage is equal to 99.97%, 99.96%, 99.95%, 99.76% and 78.1% for artificial neural network (ANN), DT, SVM, RF and NB classifiers. ANN classifier outperforms Decision Tree, SVM, Random Forest and Naive Bayes classifiers in terms of accuracy percentage.

Figure 9 presents the results obtained for the second and third metrics (Precision and Sensitivity) for the studied classifiers, considering the number of TP and FP. One can note that ANN, DT and SVM present the best performance in terms of TP, precision and sensitivity metrics. NB classifier shows the worst performance in terms of FP, precision and sensitivity metrics.

Table 7 contains the fourth metrics (MCC) for the studied classifiers over the different classes. Tests shows that the ANN, DT and SVM classifiers provide the best quality among the studied classifiers regarding MCC.

According to Table 8, NB presents the worst MCC value to classify MitM and Scanning attacks with a percentage of 11.8% and 25.1%, respectively.

5.3. Second Scenario: Experimentation with BCT

Several research works consider both BC and ML-based solutions to protect their network against security attacks. For instance, the authors in [41] suggested classifying an attack on BC with the deep learning (DL) approach based on historical security data, SHA256 encryption to hash public key and a secret key to sign transaction. The authors in [42] proposed AES encryption to protect IoT-generated data and a BC-based solution to ensure the dataset and ML algorithms integrity for e-health applications. Furthermore, a DL approach was suggested by the authors in [43] to detect security attacks for Ethereum classic network-based BC. Likewise, the authors in [44] proposed the usage of BC for collaborated federated learning (CFL) in order to preserve the privacy of data edge devices. Indeed, a CFL is defined as a distributed implementation of centralized ML in which each edge device trains their local model and then sends their model parameters to the central controller in order to develop an aggregated ML model.

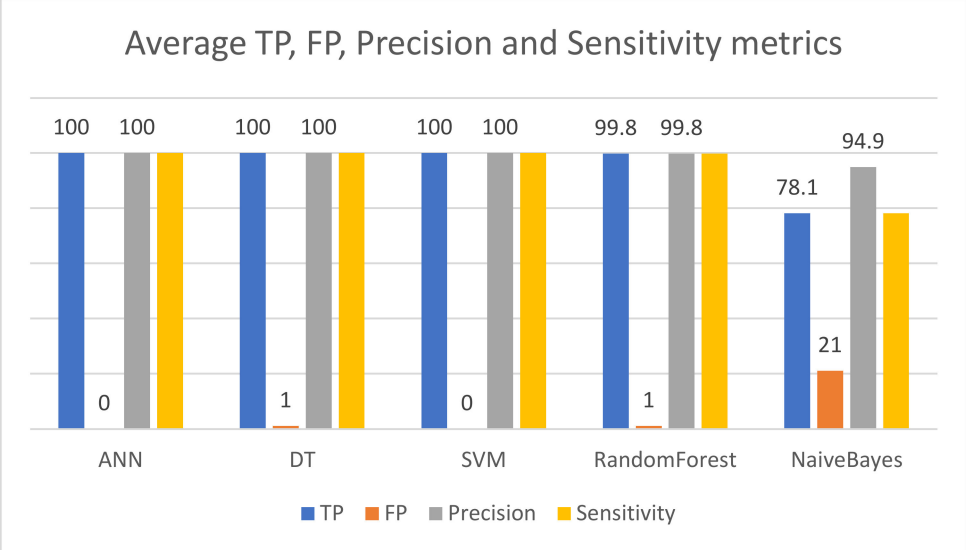

Figure 9. Average classifier measured metrics.

In this section, a framework based on a hybrid solution in which a BC structure based on smart contract is used to store sensor access control information and an ML classifier is employed to detect DDoS, DoS, injection, MitM, brute force, XSS and scanning attacks.

As depicted Figure 10, a BC access control data structure is now used to collect all assets needed for identity authentication, authorized persons to use sensors, location for the same command to detect DDoS attacks and number of tentative providing a wrong

password to detect brute forcing attacks. On the one hand, by considering a BC data structure and thanks to the inherent features of BCT such as immutability and transparency, the number of injection and XSS attacks could be reduced considerably. On the other hand, the data stored into the BC can be used as an input to create the dataset.

Table 8. MCC value per class for the studied classifiers related to the first scenario.

%	Normal Activity	DDoS	DoS	Injection	MitM	Brute Force	XSS	Scanning
ANN	100	100	99.3	100	100	100	100	99.7
DT	99.9	100	99.9	99.4	100	100	100	99.8
SVM	100	100	100	100	100	99.8	100	100
RF	99.5	99.7	99.3	99.6	57.7	99.8	99.4	99
NB	61.9	95.8	56.6	82.4	11.8	98.2	87.8	25.1

To validate our model in this context, a five-step-based scenario is proposed as follows:

1. Data sensing gathered from physical layer (including read/write of new sensor value and authentication request).
2. Data control and validation performed by the access control system in the transport layer level (i.e., the client process will prepare the sensorID, userID and userpwd to be sent to server process for an authentication request).
3. BC access control data structure will be used to save relevant data related to sensor authentication process (i.e., including authentication identity, authorized persons, location for the same command and the number of tentative with wrong password. All collected data are used to create a smart contract describing the sensor information asset performed by the application layer to provide sensor authentication service.
4. ML tools and optimization technique. This process is carried out by the advanced service layer based on the information provided by the BC access control data structure to detect DDoS and brute force attacks by using the gathered data (i.e., LocationForSameCommand and Numberoftentativewithwrongpassword will serve as features). The optimization technique is employed through the usage of the SMO classifier (a kind of SVM classifier) by implementing John Platt's sequential minimal optimization algorithm for training a support vector classifier.
5. Data analysis and performance metrics: this step is required to determine the performance evaluation of the proposed model in terms of the four introduced metrics (accuracy, precision, sensitivity and MCC).

In the second scenario, the same classification process as in the first scenario (as described in Figure 7) was selected, considering extra features (authenticationIdentity, AuthorizedPersons, Numberoftentativewithwrongpassword and CommandLocation) provided by the smart contract data structure.

Figure 11 describes the data instance repartition for the various studied attacks. The BC access control data structure is considered in terms of normal activity and the detected attacks in the IIoT networks. Based on the BC access control data structure, the results illustrate the ability of the system to detect brute force attacks by counting the number of times tentatively guessing a password that exceeds a number of times (this value has been set to five in our experiments). Several locations for the same command are considered as a DDoS attack against an IIoT network (i.e., greater or equal to two locations).

By considering BC for sensor access control, DDoS, injection, brute force and XSS attacks are reduced considerably: as depicted in Figure 11, the DDoS, Brute force, injection and XSS attacks are reduced by a factor of 34.98%, 25.93%, 37.90% and 46.57%, respectively. Brute force attack cannot be detected for 100% due to the wrong number of times tentatively entering a password performed by an authorized user. In addition, DDoS could not be detected due to the huge variant of this kind of attack.

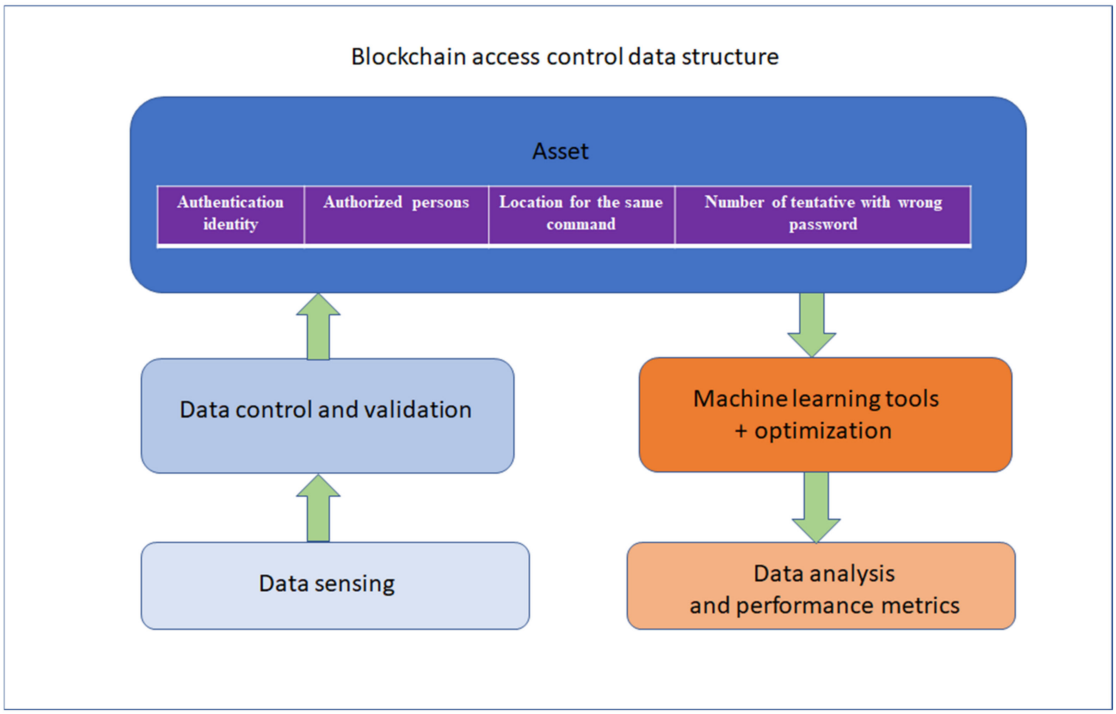

Figure 10. Advanced scenario by considering BC data structure and ML tools.

Table 9 contains the results regarding the MCC metric for the studied classifiers over the different classes based on a BC data structure.

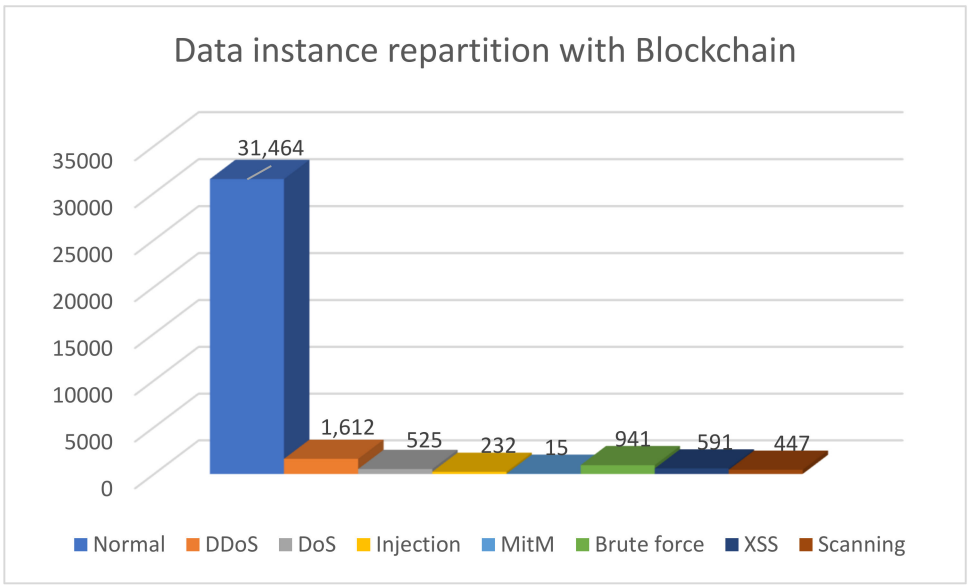

Figure 11. Data instance repartition with considering BC data structure.

According to Table 9, an improvement in terms of MCC performance metric to classify normal activity, DDoS, injection, brute force and XSS attacks for DT, SVM, RF and NB classifiers can be identified. Moreover, an improvement of 0.1% and 1.8% is observed regarding RF and NB to classify DDoS attacks. Likewise, an enhancement of 0.2%, 0.1% and 1.8% is observed regarding DT, RF and NB to classify injection attack. Furthermore, the MCC value is increased by 0.1%, 0.2% and 1.3% for SVM, RF and NB to classify brute

force attack. Finally, an improvement of 0.1% and 2.6% is observed regarding RF and NB to classify XSS attacks.

Table 9. MCC value per class for the studied classifiers related to the second scenario.

%	Normal Activity	DDoS	DoS	Injection	MitM	Brute Force	XSS	Scanning
ANN	100	100	99.3	100	100	100	100	99.7
DT	99.9	100	99.9	99.6	100	100	100	99.8
SVM	100	100	100	100	100	99.9	100	100
RF	99.6	99.8	99.3	99.7	57.7	100	99.7	99
NB	63.2	97.6	56.6	85.4	11.8	98.5	90.4	25.1

6. Discussion

In this section, we discuss the results obtained. Firstly, the first scenario revealed that the ANN classifier outperforms Decision Tree, SVM, Random Forest and Naive Bayes classifiers in terms of accuracy percentage. Additionally, ANN, DT and SVM presented the best performance in terms of TP, precision and sensitivity metrics. However, the NB classifier showed the worst performance in terms of FP, precision and sensitivity metrics.

Secondly, regarding the second scenario based on the provided BC access control data structure, the use of the proposed architecture led to an increase in the ML performance metrics in terms of MCC to detect DDoS, injection, brute force and XSS attacks thanks to extra features provided by the smart contract. The best results were given by the NB classifier to classify an XSS attack while the worst results were provided by RF and SVM classifiers to classify DDoS, injection and brute force attacks, respectively.

Table 10 presents an ML metrics comparison for the studied classifiers versus a similar study in the literature [15] in terms of time for training (s), time for prediction (s) and accuracy. On the one hand, ANN is compared to the K-nearest neighbors (KNN) classifier, NB is compared to our work and in [15]. On the other hand, two classifiers based on ensemble model techniques such as AdaBoost and XGBoost [15] are compared as well. The XGBoost classifier is based on a genetic algorithm that is used essentially to find the differentiable loss function concern.

Table 10. ML metrics comparison for various classifiers.

Metrics	ANN	KNN [15]	NB	NB [15]	AdaBoost	XGBoost [15]
Time for training (s)	3232.14	1.119	0.55	1.115	3.81	1.412
Time for prediction (s)	2.72	1.482	3.06	1.112	0.55	1.118
Accuracy	99.97	91.73	78.1	68.5	81.943	95.56

As shown from Table 10, ANN outperforms the KNN [15] in terms of accuracy. However, it requires more time for training and prediction. The NB classifier in [15] presents an accuracy of 68.5% compared to 78.1% in our study. In addition, XGBoost [15] outperforms the AdaBoost classifier in terms of accuracy due to its capability to include an arbitrary loss function optimization. Nonetheless, XGBoost requires more time for prediction compared to the AdaBoost classifier.

Some limitations of our work can be found. First of all, the number of experiments must be increased to enable future complete statistical studies; this paper contains only preliminary works that can only illustrate, at the writing of the paper, the potential benefits of the proposed architecture to mitigate various cyber-attacks in smart manufacturing applications.

Several prospects can be identified. First, a more complete experimental protocol must be led to statistically prove the performance of our architecture, which was only illustrated in this paper on a limited set of experiments. Another prospect concerns the

implementation of an optimization technique such as ant colony optimization, genetic algorithm, heuristic and particle swarm optimization to improve and refine the ML layout analysis. Optimization technique can be a potential solution to enhance the performance metrics provided by the ML tools. For instance, the number of neurons as well as the weight and bias in each neuron can play a crucial role in the performance evaluation of the ANN classifier. Another prospect is related to the improvement of our framework using an RL technique to detect new kinds of threats and attacks in the IIoT network. Finally, our architecture can be augmented with new emerging technologies such as the 5G/6G communication system in the network/protocol layer supporting cloud radio access network (CRAN) and fog radio access network (FRAN) architectures.

7. Conclusions

In this work, an IIoT architecture based on smart contract-based BCT and ML was proposed for smart manufacturing applications. The architecture is composed of five layers including sensing, network/protocol, transport enforced with BCT components, application and advanced services (ML and BCT data and cloud services) layers. A comparative study was carried out with contributions from the literature to position our approach in terms of architectural design, smart contract data structure and application of smart contract in the manufacturing field. To illustrate the potential benefits of our architecture, a data-driven study based on the TON_IoT dataset was investigated. The performance metrics of ML classifiers against common attacks in an industrial context were considered with two scenarios. A first scenario considered a data-driven analysis without BC by taking into account ML classifiers to classify the common attacks in IIoT in terms of accuracy, precision, sensitivity and MCC metrics. A second scenario used an advanced framework based on a BC smart contract data structure, smart contract code and rules for sensor access control along with ML classifiers in the context of IIoT to detect DoS, DDoS, injection, XSS, scanning and brute force attacks. It was demonstrated in our experiments that by implementing a framework based on the second scenario, the number of DDoS, injection, brute force and XSS attacks were reduced considerably. The DDoS, brute force, injection and XSS attacks are reduced by a factor of 34.98%, 25.93%, 37.90% and 46.57%, respectively. An improvement of 0.1% and 1.8% was also observed for the MCC metric regarding RF and NB classifiers to classify DDoS attacks. Finally, the MCC value is increased by 0.1%, 0.2% and 1.3% for SVM, RF and NB classifiers to classify brute force attack.

Author Contributions: Conceptualization, H.M. and A.J.; software, H.M.; validation, D.T.; formal analysis, A.A.; investigation, A.A.; resources, D.T.; data curation, H.M.; writing—original draft preparation, H.M.; writing—review and editing, H.M. and D.T.; visualization, A.A.; supervision, D.T.; project administration, A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kusiak, A. Smart manufacturing. *Int. J. Prod. Res.* **2017**, *56*, 508–517. [CrossRef]
2. Kagermann, C.H.; Helbig, J.; Hellinger, A.; Wahlster, W. Recommendations for Implementing the Strategic Initiative Industry 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. 2013. Available online: https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf (accessed on 1 February 2022).
3. Da Xu, L.; Xu, E.L.; Li, L. Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **2018**, *56*, 2941–2962.
4. Zolanvari, M.; Teixeira, M.A.; Gupta, L.; Khan, K.M.; Jain, R. Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 6822–6834. [CrossRef]

5. Tuptuk, N.; Hailes, S. Security of smart manufacturing systems. *J. Manuf. Syst.* **2018**, *47*, 93–106. [CrossRef]
6. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997; Volume 45. *ISBN 007042* 8077.
7. Aboelwafa, M.M.N.; Seddik, K.G.; Eldefrawy, M.H.; Gadallah, Y.; Gidlund, M. A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 8462–8471. [CrossRef]
8. Latif, S.; Zou, Z.; Idrees, Z.; Ahmad, J. A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network. *IEEE Access* **2020**, *8*, 89337–89350. [CrossRef]
9. Ambika, P. *Chapter Thirteen—Machine Learning and Deep Learning Algorithms on the Industrial Internet of Things (IIoT)*; Raj, P., Evangeline, P., Eds.; Advances in Computers 2020; Elsevier: Amsterdam, The Netherlands, 2020; Volume 117, pp. 321–338.
10. El Mamy, S.; Mrabet, H.; Gharbi, H.; Jemai, A.; Trentesaux, D. A Survey on the Usage of Blockchain Technology for Cyber-Threats in the Context of Industry 4.0. *Sustainability* **2020**, *12*, 9179. [CrossRef]
11. Mrabet, H.; Belguith, S.; Alhomoud, A.; Jemai, A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors* **2020**, *20*, 3625. [CrossRef]
12. Gao, J.; Asamoah, K.O.; Sifah, E.B.; Smahi, A.; Xia, Q.; Xia, H.; Zhang, X.; Dong, G. GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid. *IEEE Access* **2018**, *6*, 9917–9925. [CrossRef]
13. Dai, H.-N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A Survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
14. Zhao, S.; Li, S.; Yao, Y. Blockchain Enabled Industrial Internet of Things Technology. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1442–1453. [CrossRef]
15. Shahbazi, Z.; Byun, Y.-C. Integration of Blockchain, IoT and Machine Learning for Multistage Quality Control and Enhancing Security in Smart Manufacturing. *Sensors* **2021**, *21*, 1467. [CrossRef] [PubMed]
16. Maleh, Y.; Shojafar, M.; Alazab, M.; Romdhani, I. *Blockchain for Cybersecurity and Privacy Architectures, Challenges, and Applications*; CRC Press: Boca Raton, FL, USA, 2020.
17. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 25 March 2022).
18. Said, A.; Janjua, M.U.; Hassan, S.-U.; Muzammal, Z.; Saleem, T.; Thaipisutikul, T.; Tuarob, S.; Nawaz, R. Detailed analysis of Ethereum network on transaction behavior, community structure and link prediction. *PeerJ Comput. Sci.* **2021**, *7*, e815. [CrossRef]
19. Jović, M.; Tijan, E.; Žgaljić, D.; Aksentijević, S. Improving Maritime Transport Sustainability Using Blockchain-Based Information Exchange. *Sustainability* **2020**, *12*, 8866. [CrossRef]
20. Litke, A.; Anagnostopoulos, D.; Varvarigou, T. Blockchains for Supply Chain Management: Architectural Elements and Challenges towards a Global Scale Deployment. *Logistics* **2019**, *3*, 5. [CrossRef]
21. Prashar, D.; Jha, N.; Jha, S.; Lee, Y.; Joshi, G. Blockchain-Based Traceability and Visibility for Agricultural Products: A Decentralized Way of Ensuring Food Safety in India. *Sustainability* **2020**, *12*, 3497. [CrossRef]
22. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1. [CrossRef]
23. Klöckner, M.; Kurpjuweit, S.; Velu, C.; Wagner, S.M. Does Blockchain for 3D Printing Offer Opportunities for Business Model Innovation? *Res. Manag.* **2020**, *63*, 18–27. [CrossRef]
24. Wang, Y.-C.; Chen, C.-L.; Deng, Y.-Y. Authorization Mechanism Based on Blockchain Technology for Protecting Museum-Digital Property Rights. *Appl. Sci.* **2021**, *11*, 1085. [CrossRef]
25. Liang, G.; Weller, S.R.; Luo, F.; Zhao, J.; Dong, Z.Y. Distributed Blockchain-Based Data Protection Framework for Modern Power Systems Against Cyber Attacks. *IEEE Trans. Smart Grid* **2018**, *10*, 3162–3173. [CrossRef]
26. Aitzhan, N.Z.; Svetinovic, D. Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852. [CrossRef]
27. Lin, C.; He, D.; Huang, X.; Choo, K.-K.R.; Vasilakos, A.V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52. [CrossRef]
28. Tanwar, S.; Bhatia, Q.; Patel, P.; Kumari, A.; Singh, P.K.; Hong, W.-C. Machine Learning Adoption in Blockchain-Based Smart Applications: The Challenges, and a Way Forward. *IEEE Access* **2019**, *8*, 474–488. [CrossRef]
29. Jameel, F.; Javaid, U.; Khan, W.; Aman, M.; Pervaiz, H.; Jäntti, R. Reinforcement Learning in Blockchain-Enabled IIoT Networks: A Survey of Recent Advances and Open Challenges. *Sustainability* **2020**, *12*, 5161. [CrossRef]
30. Javaid, M.; Haleem, A.; Singh, R.P.; Khan, S.; Suman, R. Blockchain technology applications for Industry 4.0: A literature-based review. *Blockchain Res. Appl.* **2021**, *2*, 100027. [CrossRef]
31. Rathee, G.; Balasaraswathi, M.; Chandran, K.P.; Gupta, S.D.; Boopathi, C.S. A secure IoT sensors communication in industry 4.0 using blockchain technology. *J. Ambient Intell. Humaniz. Comput.* **2020**, *12*, 533–545. [CrossRef]
32. Shrivastava, A.; Krishna, K.M.; Rinawa, M.L.; Soni, M.; Ramkumar, G.; Jaiswal, S. Inclusion of IoT, ML, and Blockchain Technologies in Next Generation Industry 4.0 Environment. *Mater. Today Proc.* **2021**. [CrossRef]
33. Leng, J.; Ye, S.; Zhou, M.; Zhao, J.L.; Liu, Q.; Guo, W.; Cao, W.; Fu, L. Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 237–252. [CrossRef]
34. Faridi, M.S.; Ali, S.; Duan, G.; Wang, G. Blockchain and IoT Based Textile Manufacturing Traceability System in Industry 4.0. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage*; Wang, G., Chen, B., Li, W., Di Pietro, R., Yan, X., Han, H., Eds.; SpaCCS 2020. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12382.

35. Chen, Y.; Lu, Y.; Bulysheva, L.; Kataev, M.Y. Applications of Blockchain in Industry 4.0: A Review. *Inf. Syst. Front.* **2022**, 1–15. [CrossRef]
36. Ghaffari, F.; Bertin, E.; Crespi, N.; Behrad, S.; Hatin, J. A Novel Access Control Method Via Smart Contracts for Internet-Based Service Provisioning. *IEEE Access* **2021**, *9*, 81253–81273. [CrossRef]
37. Zaidi, S.Y.A.; Shah, M.A.; Khattak, H.A.; Maple, C.; Rauf, H.T.; El-Sherbeeny, A.M.; El-Meligy, M.A. An Attribute-Based Access Control for IoT Using Blockchain and Smart Contracts. *Sustainability* **2021**, *13*, 10556. [CrossRef]
38. Kirli, D.; Couraud, B.; Robu, V.; Salgado-Bravo, M.; Norbu, S.; Andoni, M.; Antonopoulos, I.; Negrete-Pincetic, M.; Flynn, D.; Kiprakis, A. Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. *Renew. Sustain. Energy Rev.* **2022**, *158*, 112013. [CrossRef]
39. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet Things J.* **2019**, *6*, 2188–2204. [CrossRef]
40. Moustafa, N. New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets. In Proceedings of the eResearch Australasia Conference 2019, Brisbane, Australia, 21–25 October 2019.
41. Taha, S.; Abdulrazzaq, F.; Safauldeen Omar, M.; Mustafa, A. Decentralized security and data integrity of blockchain using deep learning techniques. *Period. Eng. Nat. Sci.* **2020**, *8*, 1911–1923.
42. Gadekallu, T.R.; Kumar, N.; Hakak, S.; Bhattacharya, S. Blockchain-Based Attack Detection on Machine Learning Algorithms for IoT-Based e-Health Applications. *IEEE Internet Things Mag.* **2021**, *4*, 30–33. [CrossRef]
43. Scicchitano, F.; Liguori, A.; Guarascio, M.; Ritacco, E.; Manco, G. A Deep Learning Approach for Detecting Security Attacks on Blockchain. In Proceedings of the Fourth Italian Conference on Cyber Security (ITASEC), Ancona, Italy, 24–25 May 2020; pp. 212–222.
44. Afaq, A.; Ahmed, Z.; Haider, N.; Imran, M. Blockchain-based Collaborated Federated Learning for Improved Security, Privacy and Reliability. *arXiv* **2022**, arXiv:2201.08551.