Neural Computing and Applications (2020) 32:17361–17378
https://doi.org/10.1007/s00521-020-05189-8

S.I. : EMERGING APPLICATIONS OF DEEP LEARNING AND SPIKING ANN

Anomaly detection via blockchained deep learning smart contracts in industry 4.0

Konstantinos Demertzis[1] · Lazaros Iliadis[1] · Nikos Tziritas[2] · Panagiotis Kikiras[3]

Received: 23 December 2019 / Accepted: 11 July 2020 / Published online: 21 July 2020

Abstract
The complexity of threats in the ever-changing environment of modern industry is constantly increasing. At the same time, traditional security systems fail to detect serious threats of increasing depth and duration. Therefore, alternative, intelligent solutions should be used to detect anomalies in the operating parameters of the infrastructures concerned, while ensuring the anonymity and confidentiality of industrial information. *Blockchain* is an encrypted, distributed archiving system designed to allow for the creation of real-time log files that are unequivocally linked. This ensures the security and transparency of transactions. This research presents, for the first time in the literature, an innovative *Blockchain Security Architecture* that aims to ensure network communication between traded Industrial Internet of Things devices, following the Industry 4.0 standard and based on *Deep Learning Smart Contracts*. The proposed smart contracts are implementing (via computer programming) a bilateral traffic control agreement to detect anomalies based on a trained Deep Autoencoder Neural Network. This architecture enables the creation of a secure distributed platform that can control and complete associated transactions in critical infrastructure networks, without the intervention of a single central authority. It is a novel approach that fuses artificial intelligence in the Blockchain, not as a supportive framework that enhances the capabilities of the network, but as an active structural element, indispensable and necessary for its completion.

Keywords Industry 4.0 · Industrial IoT · Blockchain · Smart contracts · Anomaly detection · Advanced persistent threat

1 Introduction

Industry 4.0, commonly referred to as the fourth industrial revolution [1], is concerned with the trend of automation and data sharing in mass plant technologies. It includes technologies like Cyber-Physical Systems (CPS) [2], Internet of Things/Industrial Internet of Things (IoT/IIoT) [3], Cloud and Cognitive Computing (COC) [4].

CPS function within modular and structured-smart factories, where they monitor physical processes, they create virtual copies of the physical world, in order to make decentralized decisions [2]. Through IIoT, Cyber-Physical Systems communicate and collaborate in real time with each other and with people. This is performed internally and also by employing external organizational services, offered and used by participants in the production chain. This vision allows the manufacturing sector to make huge leaps in innovation, gain significant extroversion, and develop activities that were previously impossible [3].

✉ Lazaros Iliadis
liliadis@civil.duth.gr

Konstantinos Demertzis
kdemertz@fmenr.duth.gr

Nikos Tziritas
nikolaos@siat.ac.cn

Panagiotis Kikiras
kikirasp@uth.gr

[1] Department of Civil Engineering, School of Engineering, Faculty of Mathematics Programming and General Courses, Democritus University of Thrace, Kimmeria, Xanthi, Greece

[2] Research Center for Cloud Computing, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[3] Department of Computer Science, School of Science, University of Thessaly, Lamia, Greece

These new capabilities presuppose that industrial systems that control the process of production and operation of smart factories, have continuous access to the Internet, their industrial networks, as well as to the information and data of the company they belong to. Such access, digital communication and connectivity, improve the efficiency of their operation, but at the same time, they pose significant challenges to the ways in which these infrastructures are secured, in terms of their digital status and integrity [4].

Specifically, Internet connectivity and data sharing increase the risk of attacks, which may be aimed at stealing, altering and spying on data or how it is handled. This can lead to the loss of sensitive data, the slaughter of individual machines or even the shutdown of entire production. Another very important fact that makes things even worse, is that the machinery and equipment in modern industrial plants are not designed to be securely connected, which makes them particularly vulnerable to cyber-attacks. This is confirmed by the increasing number of similar attacks on industrial production facilities [5].

It is particularly important to consider the inherent challenges associated with the business environment of Industry 4.0, so that it can achieve its goals. Only then will further procedures be ensured and cyber-security problems will be resolved. The ultimate target is to secure the systems and to ensure their functional continuity.

1.1 Challenges of industry 4.0

The proposed solutions should seriously consider the specificities and in particular the Industry 4.0 related challenges [6], but most of all the information protection methods, in order to ensure industrial confidentiality. It should be emphasized that production facilities and industrial systems in general, require a different kind of protection from conventional networks, as conventional security solutions, such as virus scanners or conventional firewalls, do not meet industry standards and requirements. Similarly, digital security incident monitoring and industrial detection systems, receive huge amounts of time unit data from heterogeneous systems of specialized interconnected industrial equipment [7]. Because in most cases it is not possible or appropriate to store all of this historical data centrally, it is necessary to extract real-time knowledge and data streams, containing a small but recent percentage of observations of the general set.

The exploitation of these flows, the timely behavior prediction, and the optimal decision making in dynamic shifting and feedback environments, where usually the newest data is the most important, it is necessary to process and analyze them, by employing alternative and more effective active security approaches. Algorithms that are employed to solve data flow problems, need to be dynamically adapted to new patterns or data, and moreover they should be used when the data itself is produced as a function of time. Given that the available data is scaled in a sequential order for the calculation of the error at each iteration, it is clear that the goal of these algorithms is to minimize the cumulative error for all iterations.

Intelligent real-time data analytics systems, capable of displaying rational, empirical learning and best decision-making capabilities without human intervention, are considered the most suitable for use in industrial environments. Of course, a prerequisite for achieving this goal is their proper training by historical data sets that are representative of the problem they are trying to solve. However, even these intelligent algorithms, are challenged and controlled by a variety of potential factors mainly related to their reliability and classification accuracy. Some of the most important problems encountered during their operation are related to the high speed at which flow information arrives and to the natural tendency of data to evolve over time, resulting in the loss of the classifiers' efficiency, due to the constant change of data (concept drift). Also given the heterogeneity characterizing an IIoT network comprising of various devices (e.g., switches, encoders, potentiometers, power inductors, capacitors) as well as a variety of configurable services and terminal devices (such as smart phones, laptops) designing the architecture of a system for regulating IIoT is a complex and particularly difficult process, which should take into account the following constraints [7–9]:

a. Ensure functionality combined with minimal access. This restriction is particularly found in ambiguous areas of interest, where security policies conflict. For example, allowing a user to configure and print on a 3D printer, but at the same time forbidding its use.
b. No interaction. As IoT devices are intermittently connected, access should be provided in such a way as to ensure that the particular entity can access it whenever it wishes, without further interaction.
c. Attribution of rights with dynamic redeployment capabilities. Access should be provided on the basis of a dynamic template that will be reordered according to the needs or priorities.
d. Limited resources. Many devices in the IoT ecosystem have limited computing power, minimal storage capacity and very specific energy resources.

Practically addressing the above limitations requires the achievement of the following objectives [7–9]:

System's architecture should ensure that the expression of authorization relationships between domains and trust applications is implemented in an integrated manner to ensure the principle of functionality based on minimal access. It should also be taken into consideration that the

means of expressing and enforcing authorization are burdensome to the user, they are prone to errors, and they impede the development of applications in areas of trust.

The system should not be based on a single central point of trust, as this scheme is particularly vulnerable to failure. It should also protect against Distributed Denial of Service (DDoS) attacks, which are trying to exploit the limited resources of the IoT devices. Moreover, it should protect against leaks or intercepts, of sensitive or confidential business information and secrets, such as industrial patents, production standards, and copyrights.

1.2 Contribution—novelty of this research

This work presents for the first time in the literature, an innovative digital security architecture designed to ensure network communication between the traded IIoT devices under this model. This is done in order to address the challenges and individual issues that the Industry 4.0 business environment entails. This is a type of *Blockchain* (BLO) communication, where smart contracts programmatically implement a bilateral traffic control agreement, capable to detect anomalies based on a trained *Deep Autoencoder Neural Network* (DANN). This architecture enables the creation of a secure distributed platform that can control and complete associated transactions in critical infrastructure networks, without the intervention of a single central authority. This innovative research incorporates (for the first time in the literature) Artificial Intelligence (AI) into the *Blockchain* network, not to be used as a supportive framework enhancing its capabilities, but as an active, structural element, indispensable for its completion. More specifically, this novel research, introduces a *Blockchain Security Architecture* for IIoT, which is based on Deep *Learning Smart Contracts*. This approach can improve the security and functionality of industrial applications, by providing a decentralized, reliable, peer-to-peer network for device communication and therefore critical infrastructure management. In essence, this novel architecture is employed to fill a key gap in the way industry operates, in the context of converging heterogeneous infrastructure and mid-term financial investment.

2 Discussion of the involved algorithms

The proposed system is an anomaly detection [10] framework based on Deep Learning (DEL) network architecture [11].

Anomaly Detection (ADE) [10] is the process of pattern recognition from a dataset that exhibits behavior different from the one expected. The goal is to detect high levels of potential irregularities, while maintaining low rates of false alert. There are three categories of anomaly detection techniques. The first is the Supervised ADE, where the techniques applied receive the availability of both a set of training data that is classified as normal and one that is labeled as extreme. In such cases, prediction models for normal versus abnormal classes are usually applied. The second category is the semi-supervised abnormal detection, where it is assumed that a set of training data has been characterized only as normal. The usual approach in this case is to construct an overall model that responds to normal behavior and to apply it for anomaly identification in the testing data. Finally, the third ADE type is the one of non-supervised anomaly detection, where no training data is required. The techniques in this case, assume that the normal vectors are more than the extreme ones in the testing data. If this reasoning is not true, then there should be a high rate of error estimations.

2.1 Deep learning

Deep Learning [11] is a class of Machine Learning algorithms, which apply structural architectures based on many layers. These networks simulate complex functions, using abstractions, intermediate representations and feedback relationships, capturing multiple levels of operation and optimally matching input data to the desired network output responses. Each unit in a DEL network converts its input representation to a higher level. High-level features are more general and they basically remain unchanged, while low-level ones enable the classification of inputs. Artificial systems that simulate such functions are designed to learn how to create the necessary intermediate representations and how to successfully produce their final estimates. Multiple layers of nonlinear processing units are also used to extract and transform attributes into a slightly more abstract and complex representation. These attributes can be used as input to a corresponding 'deeper' level of operation, aiming to learn multiple levels of representations which correspond to different abstraction levels. This process can lead to a supervised or unsupervised learning outcome. It is important for a deep learning process to be able to find out what characteristics are best for each level, on its own. Of course, this does not completely rule out the need for manual adjustments (e.g., configurations for different number of layers, size of layers) which can provide different degrees of abstraction.

Deep learning systems [12] have a *Credit Assignment Path* (CAP) [13] on their depth, which describes the chain of input-to-output transformations and the potential causal links between input and output. For example, in a feed-forward neural network, the depth of CAPs is the number of its hidden levels plus one, as the output level is configured as well. For repetitive neural networks, in which a

signal can be transmitted over a layer more than once, the depth of the CAP is potentially unlimited. There is no generally agreed depth threshold separating shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth > 2. Depth CAP equal to 2 has been shown to correspond to a general approach, in the sense that it can mimic any function. The need to detect anomalies in critical infrastructure networks is an extremely important process, capable of ensuring their uninterrupted and safe operation. Likewise, DEL techniques, can lead to optimization and they can enhance the accuracy performance of intelligent systems. They are considered as a reliable solution for the timely and accurate detection of anomalies in industrial design and network architectures.

2.2 The blockchain ecosystem

In this spirit and in order to maximize the robustness of the IIoT ecosystem, a secure, reliable and completely unambiguous mediator is required for device users' transactions who wish to make use of ecosystem services such as *Blockchain* [14]. BLO is a transparent, verifiable, permanent transaction management system that operates in a distributed mode, across peer networks. It offers and maintains a robust consensus mechanism that unlike normal processes, does not base its reliability on an entity acting as a credible third party. In essence, this technology can provide a simple communication infrastructure for the transfer and direct exchange of a property segment between two devices, such as money or data, with a secure and reliable time-coded conventional handshake. Also, the lack of centralized control that characterizes this technology, ensures its scalability and robustness, while utilizing the resources of all participating nodes, eliminates many-to-one traffic flows. This reduces the delay and overcomes the problem of a single failure point.

Correspondingly, the intrinsic anonymity offered and the implementation of a secure network between unreliable parties and heterogeneous devices are important features of blockchain [15] that make it an attractive technology, capable to address the most important security and privacy challenges in IIoT, especially in the case of critical infrastructure [16]. The idea of combining IIT with BLO can lead to a verifiable, secure and permanent method of recording data processed by "smart" machines capable of communicating and operating automatically. This combination creates a permanent, unchanged record of the data moving across the supply chain, resolving the very core issue of oversight. Moreover, this activity can be monitored and analyzed in a transparent manner by anyone authorized to connect to the network. If something goes wrong with the integrity or quality of the process (e.g., data leaks), the blockchain makes it simple to identify the weak link and it takes corrective action.

The use of BLO encryption and distributed storage, also reinforces the belief that all stakeholders can trust the processes involved in the supply chain, since the same machines can safely record the details of transactions made between them, without human supervision. This creates a file that no human is able to replace or "infect" with inaccurate information, as the private keys that give write access to the BLO, are monitored by machines. Some of the most powerful available encryption standards enforce the overall security of the IIoT environment [16]. Due to the distributed nature of this approach and to its inherent ability to copy the log of its processes to hundreds or possibly thousands or millions of devices, the scaling and decomposition requirements that the IIoT has to solve are met.

In particular, the enormous number of devices integrated into an IIoT, using BLO, are detached from the existence of a centralized data storage and processing service. In fact, data can always be available when needed, as there is no single point of failure. The required time for data access or data transfer is reduced and the network scalability potential is maximized. The performance is further enhanced by the use of *Blockchain-enabled Smart Contracts* (BCeSC), which allow communication between machines under specific strictly defined and pre-agreed conditions [17]. These contracts allow the execution of the agreements, when specific conditions are met. Essentially, the result is the digital facilitation, verification or enforcement of the negotiation or execution of a contract. These contracts allow the execution of credible transactions, without third parties involved, whereas the monitored transactions are secured, and irreversible. The actual purpose is the achievement of security, (which is superior to contract law) and the reduction of the additional transaction costs associated with the award and execution of brokerage services. BCeSC are visible to all users of the network. This feature enhances the transparency and credibility of transactions in complex environments. This automates the adoption of a system for carrying out a pre-agreed procedure, where conditions clearly indicate that the pre-agreed services are provided from both sides. This approach, allows smart contracts that go beyond simple data or capital transfers, by incorporating code with more extensive instructions. Specifically, rules apply to a variety of possibilities and they offer functionality based on changes, on facts, on external data, on enforcement and proof [17].

These capabilities include monitoring and adjusting the status of the system over time. Finally, it is important to note that smart contracts are autonomous since the software developer who created them does not need to monitor them

during their function. Once executed, they can be self-sufficient, as they are programmed to commit the necessary resources, such as processing power and storage space. They guarantee the execution of the intended transaction when the required conditions are met and they can prove their solvency [18].

The *Blockchain* architecture, acts as a distributed database or global registry that holds logs for all transactions on a network [19]. A transaction is a time-stamped record that specifies the identity and mode of an operation, the function itself, and the users involved. Transactions are combined into blocks where each block is identified by a cryptographic hash. Each user forms an open public–private key pair (linked to his/her account) which is used to sign a transaction and to clearly identify the ownership of a function. To build a block, a hash function is applied with all the transaction information, and then the hash value is used to calculate its cryptographic form. Thus, transaction information becomes unchanged. This fact ensures the security, authenticity and robustness of the BLO data storage. If there are conflicting transactions in the network, only one of them is selected to become part of the block. The blocks are added linearly to the chain at regular intervals, where each one reports the hash of the previous. In this way, each node can execute and record all transactions made. The trading account is designed in a way that allows it to be distributed. This means that each user has access to the entire transaction log and can check the hash of each new block. Thus, it is possible to confirm the correctness of each transaction. Following this approach, it is possible to reach a common consensus when performing the functions. This built-in consent mechanism, allows the organization of all published information. Any protocols can be included at the top of the chain, during the management of the business processes [15, 19].

3 Architecture of the proposed system

The proposed system is a fully decentralized authorization system, based on the WAVE architecture [20] that operates on a global scale providing fine-grained permissions, non-interactive delegation, and proofs of permission that can be efficiently verified, while still supporting revocation. The architecture comprises of the Authorization, Syndication, and Overlay layers, which are described below [20]:

3.1 Authorization layer

This layer provides levels of access, by expressing security policies in the IoT ecosystem, using entities, namespaces, resources, and *Delegations of Trust* (DoTs). An Entity is a standalone control unit that can grant, reset or delete access rights. It acts as a global type of username or a role.

Anyone can develop an entity characterized by a pair of keys (public and private). For example, the admin is a pair $<A_{pk}, A_{vk}>$ where A_{pk} is the public key and A_{vk} the private one. A namespace is a sector containing a hierarchy of resources. The *Raw_Materials_Utility* (RMU) is a namespace and each included resource is related to a unique *Uniform Resource Identifier* (URI). The temperature of a thermostat in the RMU is described by the URI as follows: *BW://Raw_Materials_Utility/thermostat/stats/temp*. The term "stats" refers to data describing the operational status of the resource, while the term "cmd" includes control commands such (e.g., "restart", "lock"). Any other entity that interacts with resources within a namespace, must obtain permission from the namespace entity, directly or indirectly. This is the Delegations of Trust (DoT) property.

$$\text{E.g. DoT} = A_{pk}^{from}, A_{pk}^{to}, URI_{rsrc}, Permissions, Metadata$$

The metadata can use specific properties, characterizing the resources.

3.2 Syndication layer

This layer provides publish/subscribe functions to system resources and it is directly related to the Authorization layer. For example, a specific resource can be published in a namespace by an entity, receiving a specific URI.

The *subscribe permissions* determine if a *receiver_entity* can receive information, from the published resource. The *publish permissions* decide if an entity can publish–interact with the resources (e.g., to send restart commands, to control the thermostat operation temperature). The subscribe permissions are related to the "stats" of the URI, whereas the publish permissions with the "cmd" capabilities.

3.3 Overlay layer

This layer forms the communication network between the IoT devices, which is responsible for providing the available applications-services, and the *Blockchain* network. This layer essentially forms an overlay network over the existing physical network. Its nodes can be considered as connected to virtual or logical links, each corresponding to a path, to the underlying network.

3.4 Deep learning smart contracts

A *Smart Contract* (SC) [17, 18] is a piece of code located in the *Blockchain* network. It can be considered as a decentralized application available to all users on the

network, which is recognized by a single address. The smart contract code contains a set of executable functions and state variables, with functions executed when transactions are made. Transactions include input parameters required by the contract. When executing a function, the state variables change according to the logic applied each time. At the same time, the conditions under which the contract must be executed and the list of actions attributed to the submitted conditions, must be fulfilled. It should be emphasized that all conditions of a *smart contract* must have a mathematical description and a clear execution logic. Smart Contracts can be written in a variety of high-level languages, such as *Solidity* or *Python*. As a result, the conventional code description allows the application of complex algorithmic relationships. Once compiled, they are uploaded to the *Blockchain* network, which assigns a unique address for each new SC. After signing by the contracting parties, using their open private keys, the contract enters into force. Each user can activate the contract functions by sending a transaction written in the block chain. The conventional code is executed at each node participating in the network, as part of verifying new blocks. All *Blockchain* nodes execute the same commands thereby they provide redundant execution of SC. While this is not an effective approach, it is needed to maintain consensus on the network, as there is no central authority or a reliable third party that will validate the procedures. A corresponding environment is required that allows full automation, in order to ensure the automated execution of contractual obligations. This means that smart contracts can only exist within an environment that has unlimited access to the executable code of smart objects. Thus, a SC follows the defined conditions of reaching or violating the terms of the contract and makes independent decisions based on the planned conditions. In this way, the basic principle of a smart contract, which is the full automation and reliability of the performance of contractual obligations between the parties, is fully satisfied. Concluding, we can say that smart contracts act as agreements between the parties involved, allowing secure trading and hence the development of credible decentralized peer-to-peer transactions.

In this case, the proposed SC is based on Deep learning, to detect malicious web traffic and malfunctions in general, which may be an indication of malicious activity among the IIoT devices. Specifically, the contract includes the export and intelligent analysis with a trained *Autoencoder*-type Neural Network (NN) [21, 22], and the features of network traffic between the vending machines.

3.5 Autoencoders

An *Autoencoder* (AUE) is a NN [21, 22] that is separated into a pair of two connected networks, one having the role of the encoder and the other of the decoder. Autoencoders consists of 4 main parts:

1. Encoder: in which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.
2. Bottleneck: which is the layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data.
3. Decoder: in which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.
4. Reconstruction loss: this is the method that measures measure how well the decoder is performing and how close the output is to the original input.

The encoder network receives an input and converts it into a smaller, dense representation, which the second decoder network can use to convert it to the original input. AUEs work by compressing the input into a latency representation and then by reconstructing the output from it. In this way they learn to compress the original data from the input layer into an abstract form, which then decompresses this form, by converting it into something that fits perfectly with the original data. This forces AUE to get involved in the reduction of the dimensions of an initial problem and in learning how to ignore noise. Architecturally, the simplest form of an Autoencoder is a feed-forward, non-recurrent neural network similar to multilayer perceptron, which has an input layer, an output layer, and one or more hidden layers that connect them. The output layer must have the same number of nodes as the input layer, in order to reconstruct the inputs (instead of predicting the target value Y given inputs X). Autoencoders are considered unsupervised learning models as they do not need labeled data to be trained. The encoder and decoder can be defined as transitions x and $\hat{x}$ so that [22]:

$$x : X \rightarrow F \tag{1}$$

$$\hat{x} : F \rightarrow X \tag{2}$$

$$x, \hat{x} : \underset{x,\hat{x}}{\arg\min} \|X - (\hat{x} \circ x)X\|^2 \tag{3}$$

The training process is based on the optimization of a cost function, which measures the error between input x and its reconstructed output $\hat{x}$.

If the input to an Autoencoder is a vector $x \in \mathbb{R}^{D_x}$ then the encoder corresponds vector x to another vector $z \in \mathbb{R}^{D^{(1)}}$ as seen in function 4 below:

$$z = h^{(1)}\left(W^{(1)}x + b^{(1)}\right) \tag{4}$$

The notation (1) indicates the network level (in this case it is the first level of the network). Moreover, $h^{(1)} : \mathbb{R}^{D^{(1)}} \rightarrow \mathbb{R}^{D^{(1)}}$ is the transfer function of the encoder, $W^{(1)} \in \mathbb{R}^{D^{(1)}_{xD_x}}$ is a weight matrix, and $b^{(1)} \in \mathbb{R}^{D^{(1)}}$ is a bias vector. The encoder then maps the encoded representation z again to an estimate of the original input vector, x, as follows:

$$\hat{x} = h^{(2)}\left(W^{(2)}z + b^{(2)}\right) \tag{5}$$

The symbol (2) corresponds to the second layer of the NN. Specifically, $h^{(2)} : \mathbb{R}^{D_x} \rightarrow \mathbb{R}^{D_x}$ is the transfer function of the encoder, $W^{(1)} \in \mathbb{R}^{D_{x\times D^{(1)}}}$ is a weight matrix, and $b^{(2)} \in \mathbb{R}^{D_x}$ is a bias vector (Fig. 1).

The decoder function, maps the latent space at the bottleneck to the output. The output, in this case, is the same as the input function. Thus, we are basically trying to recreate the original input after some generalized nonlinear compression. The encoding network can be represented by the standard neural network function passed through an activation function. Similarly, the decoding network can be represented in the same fashion, but with different weight, bias, and potentially activation functions being used.

There are 4 hyper-parameters that have to be tuned before the training of the Autoencoder [21, 22]:

a. *Code size*

It refers to the exact number of nodes in the middle layer

b. *Number of layers*

The Autoencoder has no limit in the levels it can integrate and it can become as deep as it takes to solve a problem.

c. *Number of nodes per level*

The number of nodes per level decreases with each subsequent level of the encoder, while the opposite occurs with the decoder. Also, the decoder is usually symmetrical to the encoder in terms of the structure of the layers.

d. *Loss function*

The used indices are Root Mean Squared Error (RMSE) or Binary Cross-entropy.

The Autoencoder used in this work, has as input layer of 18 neurons, 6 fully connected layers with 14, 7, 3 (code size), 7, 14 neurons and 18 neurons (output). Tα 3 first levels are used by the encoder and the 3 last by the decoder. The decoder is simply the complementary function that creates a map from the (encoder's) latent space to another target space (what is it we want to decode from the latent space) (Fig. 2).

The Root Mean Squared Error (RMSE) loss function was used in this approach. RMSE is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. They can be positive or negative as the predicted value under or over estimates the actual value. The RMSE is used as a measure of the spread of the y_i values about the predicted $\hat{y}_i$ ones. The formula is:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{6}$$

Additionally, Lasso Regression (Least Absolute Shrinkage and Selection Operator) or L_1 regularization [23] was used during training. L_1 adds "absolute value of magnitude" of coefficient as penalty term to the loss function. The cost function described below:

$$L_1 = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{7}$$

If lambda is equal to zero then the Ordinary Least Squares (OLS) is performed, whereas a very large value

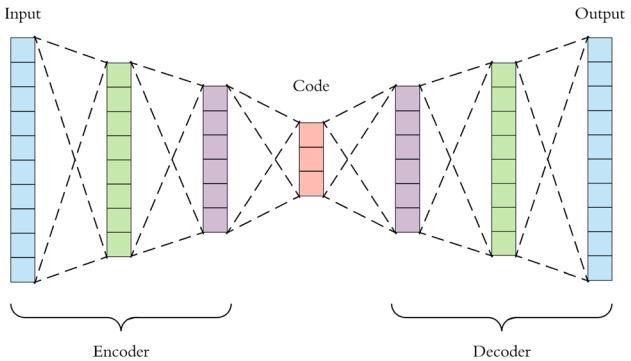

Fig. 1 A detailed visualization of an autoencoder

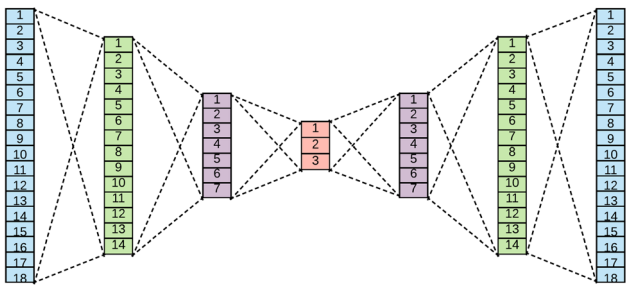

Fig. 2 Detailed visualization of the proposed deep autoencoder

makes coefficients equal to zero. This means that we have an under-fit case. Communication is allowed only if the results of the process do not exceed a limit, which is a clear indication of the smooth operation of the network traffic. The exact procedure provided by Deep Learning Smart Contract is described in detail below.

3.6 The deep learning smart contract

The following paragraph presents the relevant parameters used in drawing up the contract, which is used to analyze a relative example of device communication based on Deep Learning Smart Contract.

a. *MachineAccount*

This is the account that represents a device in the IoT ecosystem. Each machine must have its own account to sign the transactions it needs to execute

b. *MachineAddress*

It describes the unique address that each device has on the network.

c. *MachineInternals*

It concerns monitoring the internal parameters of the device, such as operating temperature, battery status, operating time.

d. *Publisher*

The user or application that generates the data on the network.

e. *Subscriber*

The user or application that uses the data on the network.

f. *Sender*

The end user who sends the data that the Publisher has generated to the network.

g. *Receiver*

The final user that receives the data produced by the Publisher in the network.

h. *MachineStatus*

It refers to the processes that check the status of the devices, such as they produce data (Publisher), when a process is requested (Sender).

i. *Session*

The time period in which a transaction takes place.

j. *SessionID*

A unique number characterizing a Session.

k. *DataStream*

A data flow that one device or application wishes to transfer to another.

l. *DataStreamID*

A unique number characterizing a DataStream.

m. *TrafficFlow*

A sequence of packages from a source device to a destination, which may be another device or group of devices.

n. *TrafficFlowID*

A unique number characterizing a TrafficFlow.

o. *FeaturesOfTrafficFlow*

The features extracted from the web traffic.

In the case of the problem under consideration, the network traffic features used to detect anomalies related to attacks are described below.

p. *IIoT rule*

It describes a process, which checks if a particular action can be implemented. For example it controls the necessary rights of those involved in the action, if the action causes an additional one (e.g., such as triggering a contract).

q. *SmartContract*

A smart contract that is triggered in specific trading situations. The Smart Contract receives a characteristic ID (in this case the described SC has an ID equal to 101).

The proposed process is described below:

The *IIoT_thing_thermostat* (Sender), wishes to interact by sending data (DataStream) in the *IIoT_thing_water_-tank* device (Receiver).

The Data Streams receive a *Stream ID* e.g., (707), whereas while the specific action of the shipment is

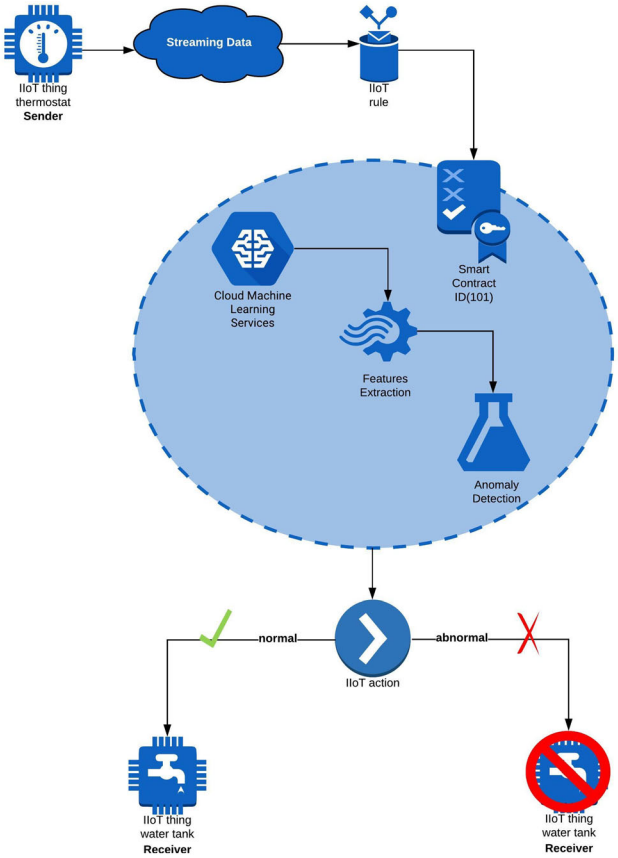

Fig. 3 Proposed approach of data transactions on blockchain

monitored by the IIOT rule on whether that particular transaction can take place.

The IIoT rule activates the *Smart Contract ID* (101). The particular Smart Contract ID (101), routes Stream IDs (707) to a cloud service, where features extraction is performed, and then the trained Deep Autoencoder checks whether the DataStream is normal or abnormal. This specification results from the *Reconstruction Error Threshold* (RET) [24, 25] which is defined in the training process for the characterization of the network's traffic and it is measured in RMSE. If the traffic is considered normal, then there is communication with the *IIoT_thing_water_tank* (Receiver), otherwise the communication is hidden and an alert is sent to the Security Operation Center for further inspection. Figure 3 presents the communication process between the devices, based on the *Deep Learning Smart Contract.*

The introduced BSA process (in pseudocode format) is described in Algorithm 1:

Algorithm 1. The proposed BSA approach

```
#MachineAccount
    IIoT_thing_thermostat;
#MachineAddress
    FE80:0000:0000:0000:0202:B3FF:FE1E:8329;
#MachineInternals
    Energy (GeV) = 1.320 | Lifetime (hours) = 9658.150 | Current (mA) =
4.210
    Last Refill = 23 May 22:11 | Mode = standby | Feedback Status = On
#MachineStatus
    Publisher; Sender;
#DataStream
    Size = 125 kb; TimeStamp = 201802305221100;
#DataStreamID
    ID = 101;
# FeaturesOfTrafficFlow
    command_address;
    response_address;
    command_memory;
    response_memory;
    command_memory_count;
    response_memory_count;
    comm_read_function;
    comm_write_fun;
    resp_read_fun;
    resp_write_fun;
    sub_function;
    command_length;
    resp_length;
    control_mode;
    control_scheme;
    pump;
    crc_rate;
    measurement;
#Functions
    function DataSender (MachineAccount, MachineAddress, MachineStatus);
    function DataReceiver (MachineAccount, MachineAddress,
MachineStatus);
    function DataTransaction (DataStream, DataStreamID);
    function DataFlow (TrafficFlow, TrafficFlowID);
    function TransactionSession (Session, SessionID);
    function FeaturesOfTrafficFlow;
    function AnomalyDetection;
#Loop
    start
        if
            DataSender to DataReceiver a DataTransaction
            then DataFlow process to FeaturesOfTrafficFlow
            and FeaturesOfTrafficFlow to AnomalyDetection
        else if
            AnomalyDetection normal then TransactionSession
    end
```

In order to implement the proposed Deep Learning Smart Contract procedure when communicating between devices in IIoT, extensive testing was performed with an appropriate dataset which is presented in detail below.

4 Description of the application case

The most common self-monitoring devices in the IIoT ecosystem are *Supervisory Control and Data Acquisition* (SCADA), *Distributed Control Systems* (DCS), implemented on the basis of *Programmable Logic Controller* (PLC) and sensors used in control loops for the collection of all kinds of measurements. The above systems, which are active devices of the infrastructure network, are properly connected to allow remote monitoring and control of

processes with high response rates, even where the devices are distributed between different remote locations.

It makes sense that this ecosystem and its specialized processes, in addition to the significant benefits and upgrades, have introduced a large number of new threats [26]. These threats are primarily related to the specific purposes these applications fulfill, their different design specifications, the specialized communication protocols they use, and the heterogeneous devices they are called upon to interconnect. For example, the communication in critical infrastructure management systems is achieved by the widespread SCADA MODBUS (SCMD) protocol [27].

SCADA MODBUS is a layer-by-layer message protocol mounted on layer 7 of the OSI model. It provides client/server communication between devices connected to different types of Bus or Network. It Provides client/server communication between devices connected to different types of Bus or Network. It is a method used to transmit information over serial lines between electronic devices. The device requesting the information is called the *Modbus* (*MDB*) *Master* and the device providing the information is the *Modbus slave*. The simplicity and effectiveness of the protocol have made it the most widely used network protocol in the industrial production environment. It has been implemented by hundreds of vendors on thousands of different devices, for transferring discrete/analog I/O and for the recording of data between control devices.

Internet access in Modbus is done at system port 502 on the TCP/IP stack. For all transactions performed under this protocol, a simple request-response scheme is used. The master device initiates a request and the slave responds. The MDB protocol application contains many vulnerabilities that could allow an attacker to perform recognition activities or issue arbitrary commands. For example, one vulnerability is the inability to recognize an illegal slave/master address on the network. An unauthorized, remote intruder could exploit this vulnerability by sending queries containing invalid addresses. In this way he will be able to collect information about network servers from the returned messages.

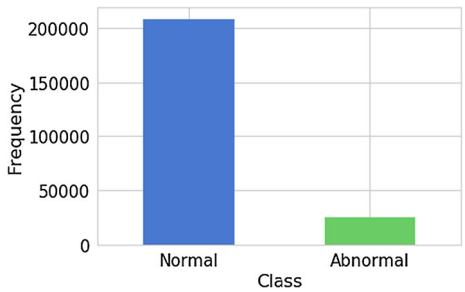

Fig. 4 IIoT transaction class distribution

Another vulnerability is due to the lack of adequate security controls in the implementation of the SCADA MODBUS protocol. The protocol specification does not include an ID/authentication mechanism to validate communication between master and slave MDB devices. This flaw could allow a remote attacker to issue arbitrary commands to any slave device via a MDB master without authentication [28].

The SCMD contains another vulnerability that could allow an attacker to trigger a Denial of Service (DoS) attack on a system. This vulnerability is due to a protocol implementation error when processing request and response read messages for discrete inputs. An unauthorized remote intruder could exploit the vulnerability by sending request or response parameters that contain malicious values for selecting the data field in a system containing a vulnerable MODBUS application.

The MDB Transfer Control Protocol (TCP) is commonly used in SCADA networks for process control. It limits the size of the Protocol Data Unit (PDU) to 253 bytes to allow the packet to be sent in a serial line, RS-485 interface. MDB TCP adds a 7 byte protocol header. This sets a ceiling on the legal packet size. An intruder creates a specially designed package, greater than 260 bytes and sends it to an MDB client or server. If the client computer or server are incorrectly programmed, this can lead to a successful buffer overflow attack [27].

Critical infrastructure attacks are referred to as Advanced Persistent Threats (APT) [29], as they can perform mechanical control, dynamic centrifugation or device reprogramming. The purpose of such attacks is to speed up or slow down their operations, leading to total industrial equipment being destroyed or permanently damaged. The cybercriminals who direct them are fully familiar with sophisticated methods and tools for exploiting unknown vulnerabilities in the general public (zero days). They try not to be perceived without taking into account time, but in most of the cases they are extremely competent, organized, financed and they have significant motivation. It is therefore understood, that in these cases any detection of malfunction in the operation of the devices in question is extremely important as it may reveal a general or ongoing attack.

One of the key automation tools of the anomaly recognition process is Deep Learning models, which are capable of learning to focus on the right features. Thus, they detect the interrelations that can lead to hidden knowledge being revealed. So, they finally arrive at safe and reliable detection, requiring little or no guidance from the developer.

4.1 Description of the dataset

The most appropriate data set that closely simulates IIoT communication and transaction data has been selected, in order to accurately illustrate the way industrial devices operate [30–33]. Specifically, the data set selected came from the MDB protocol application network data logging with an RS-232 connection. The system included two RS-232 serial ports connected to a serial USB converter. The recorder monitored each serial port for any network traffic/communication. The complete network traffic collection process as well as other details for managing the network flow are described in detail in [30–33].

The ideas of network traffic analysis and features extraction approach were based on the functional mode of the MDB protocol and on the acknowledgement method of the reliable submission and receipt of the data. The dataset used in this research contains network transaction data, which were preprocessed to strip lower layer transmission data (e.g., TCP, MAC). The *water_dataset* includes 18 independent parameters and 233,295 instances, from which 206,552 are normal and 26,743 abnormal.

A graphical depiction of the IIoT transaction class distribution in the *water dataset*, is shown in Fig. 4 below.

As can be easily seen and visualized, we have an *Imbalanced dataset* (IMD) as there is an uneven distribution of classes, with Normal occupying 88% of the majority class cases, while Abnormal occupying only 12% of the cases (minority class). This can lead to Imbalanced Learning. Machine Learning algorithms assume homogeneous allocation of classes or even misclassification costs. When presented in complex sets of IMD, they fail to represent the actual distribution characteristics and they provide unsatisfactory and inaccurate modeling of the data classes. Various methods have been proposed in the literature for handling Imbalanced datasets, such as: Resampling Techniques (*Random Under-Sampling, Random Over-Sampling, Cluster-Based Over-Sampling, Synthetic Minority Over-sampling, Modified Synthetic Minority Over-sampling*) and Algorithmic Ensemble approaches (*Bagging Based, Boosting-Based, Adaptive Boosting-Based, and Gradient Tree Boosting-Based*).

4.2 Data preprocessing

In this research, the problem of imbalanced data is addressed by an innovative application of the one-class classification (OCC) [34] methodology employing a Deep Learning Autoencoder. The system is trained exclusively with data that characterize the normal function of IIoTs, so that it can detect divergent behaviors and abnormalities associated with APT attacks. More specifically, the OCC method, also known as *Unary* classification, attempts to identify class-specific objects among all, by learning from a training set that contains only the objects of that class. Usually, these algorithms aim to implement classification models in which the negative class is absent because the missing class is not sampled as it is difficult to do so.

This operating condition, in which classifiers are called to efficiently and reliably determine classes, only with knowledge of the positive class, is a particularly complex Machine Learning (ML) problem. Assuming a uniform distribution of the other classes, the model selection method based on *Coherence* is considered as an effective one, used to select the most appropriate model parameters. It should be noted that the basic idea in OCC problem solving, is the opposite of the generalization sought in other ML cases.

Specifically, well-defined parameter tuning is sought, even where this exponentially increases the complexity of the classifier, provided that the classifier is able to categorize the target data correctly. The more complex the model, the smaller the classification range in the target data space, and the lower the likelihood of outliers being correctly categorized. In practice, one can create a complex model by adjusting all its possible parameters, without risking over-fitting. Essentially and in order to predict whether a new transaction is normal or fraudulent, we calculate the Deep Autoencoder reconstruction error from the transaction details themselves. If the error is greater than a predetermined threshold, we will classify it as abnormal, given that our model will have a low error in normal transactions.

Table 1 Confusion matrix of the deep autoencoder

Normal		Abnormal	
TP	185,557	365	FN
FP	2192	24,342	TN

4.3 Data preprocessing and dataset threshold criteria

Real-world data is often incomplete, inconsistent, they are lacking in certain behaviors or trends, and they are likely to contain many errors. Data preprocessing is a proven method of resolving such issues, including cleaning, instance selection, normalization, transformation, feature extraction and selection [35].

In the data preprocessing phase, the duplicate records and records with missing values were removed. Also, the datasets were determined and normalized to the interval [− 1, 1] in order to phase the problem of prevalence of features with wider range over the ones with a narrower range, without being more important.

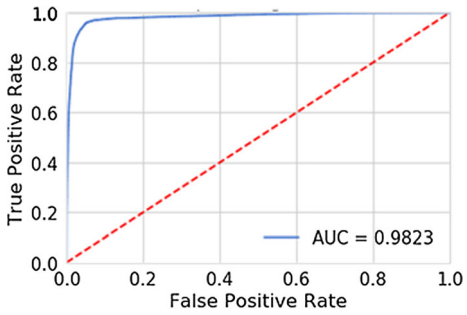

Fig. 5 Value of the AUC in the 1st fold CV

Optimal Dataset Threshold (ODT) refers to the safe way of rejecting positive (normal) class samples that are close to the edge of class separation. These samples may lead to misleading classification so they are not considered. As it has already been pointed out, the correct determination of the class separation threshold, plays the most important and critical role in the success of the OCC classification approach. The determination of the Optimal Dataset Threshold (ODT), is based on a reliable heuristic approach, based solely on evaluation criteria. In particular, the proposed algorithm assumes that the training phase defines a distance function d between the objects and the target category. The ODT separates classes (normal or abnormal) in order to reject a specified set of training samples, most of which deviate from the target class, thereby strengthening the classifier. Even when all samples are properly labeled, rejecting a small but representative percentage of training samples helps the classifier to learn the most representative ones. These vectors are not considered. The classifier becomes more robust as the classes become more distinct. It is like growing the width of the tube in Support Vector Machines.

The pseudocode of the proposed approach to determine the separation threshold of the classes is presented in Algorithm 2 below:

Algorithm 2: Optimal Dataset Threshold

Optimal Threshold:
1: Calculate the error using Euclidean distance between actual and predicted on each training data;
2: Arrange the error in decreasing order;
3: Set the threshold at rejection of 10% most erroneous data (false negative rate at the rate of 10%);

The final *water_dataset* includes 18 independent parameters and 212,456 instances, from which 185,922 are normal and 26,534 abnormal. It should be noted that the employment of the ODT, has leaded to the development of a powerful OCC classifier which is perfectly linked to the normal operating conditions of trading in the IIoT. This significantly enhances the active safety of the critical infrastructures.

5 Results

In cases of data that use a machine learning classifier in order to estimate the training error, the full probability density of both categories should be known [36, 37]. The classification performance of the *Autoencoder* approach, using 10-fold cross-validation, is presented in the following *Confusion Matrix* (CM) given in Table 1. The main diagonal values (top left corner to bottom right) correspond to correct classifications and the rest of the numbers correspond to a small number of cases that were misclassified. It must be clarified that the Training process is performed as a Unary classification effort, where only the Normal cases are determined as such. All of the cases that are not classified as Normal, constitute the second "abnormal" class.

The number of misclassifications are related to the False Positive (FP) and False Negative (FN) indices appearing in the CM. On the other hand, the True Positive (TP) and the True Negative (TN) correspond to the cases that were correctly classified as Positive or Negative respectively.

Table 2 Classification accuracy and performance metrics

Classifier	Fold	TA (%)	RMSE	Precision	Recall	*F*1-Score	AUC
Deep autoencoder	1st	98.30	0.117	0.982	0.982	0.982	0.9823
	2nd	98.86	0.113	0.988	0.988	0.988	0.9891
	3rd	98.87	0.112	0.986	0.986	0.986	0.9870
	4th	98.98	0.109	0.991	0.992	0.992	0.9992
	5th	98.72	0.118	0.987	0.987	0.987	0.9880
	6th	98.51	0.120	0.985	0.985	0.985	0.9860
	7th	99.03	0.078	0.991	0.991	0.991	0.9901
	8th	98.84	0.114	0.988	0.988	0.988	0.9889
	9th	98.88	0.112	0.989	0.988	0.988	0.9888
	10th	98.99	0.101	0.989	0.989	0.989	0.9892
	Avg	98.80	0.109	0.988	0.988	0.988	0.9891

Table 3 Comparison between other algorithms

Classifier	TA	RMSE	Precision	Recall	*F*1-Score	AUC
One-class SVM	97.12%	0.210	0.971	0.971	0.971	0.9710
Isolation forest	96.78%	0.287	0.968	0.968	0.968	0.9700
Minimum covariance determinant	95.63%	0.394	0.957	0.957	0.957	0.9600
Deep autoencoder	98.80%	0.109	0.988	0.988	0.988	0.9891

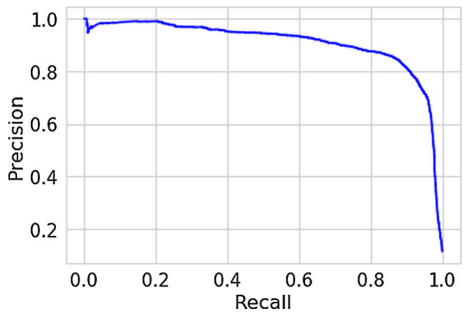

Fig. 6 Recall versus precision

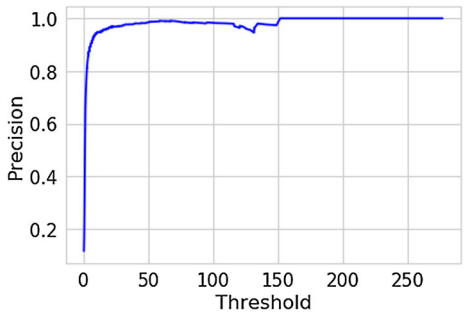

Fig. 7 Precision for different threshold values

Fig. 8 Recall for different threshold values

The *True Positive rate* (TPR) also known as *Sensitivity* the *True Negative rate* also known as *Specificity* (TNR) and the *False Positive Rate* (FPR) are defined by using Eqs. 8, 9, and 10 respectively [36, 37].

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8}$$

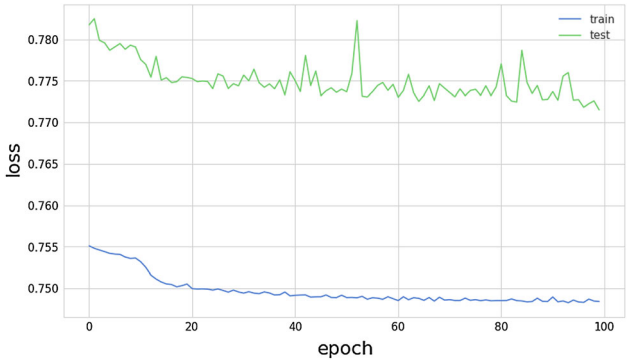

Fig. 9 Reconstruction error for training and test data

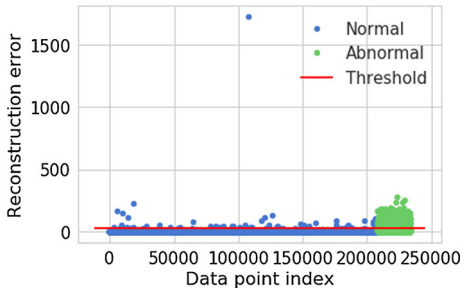

Fig. 10 Reconstruction error for different classes

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{9}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR} \tag{10}$$

The *Receiver Operating Characteristic* (ROC) curve plots the TPR versus the FPR, over different threshold values. The *Area under the Curve* (AUC) measures the entire two-dimensional area underneath the entire ROC curve from point (0, 0) to (1, 1). The AUC provides an aggregate measure of performance across all possible classification thresholds. This is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). Basically, we want the blue line to be as close as possible to the upper left corner. The *Precision* (PRE) the *Recall* (REC) and the *F1-Score* indices are defined as in Eqs. 11, 12 and 13 respectively [36, 37]:

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{11}$$

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{12}$$

$$\mathrm{F1} - \mathrm{Score} = 2\mathrm{X}\frac{\mathrm{PRE\ X\ REC}}{\mathrm{PRE}\ + \mathrm{REC}} \tag{13}$$

The *Total Accuracy* (TA) defined by using Eq. 14 [36, 37]:

$$\mathrm{TA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{N} + \mathrm{P}} \tag{14}$$

The *tenfold cross-validation* (10_FCV) was employed to obtain performance indices. Cross-validation is an approach that is used to evaluate predictive models. It shuffles the dataset randomly and then it partitions it into k groups. Each unique group is used once as a hold out or test data set, whereas the remaining groups are used as a training data set.

Indicatively, in Fig. 5 we observe that the value of the AUC for the case of the 1st fold validation process is equal to 0.9823.

Table 2 presents the values of the used indices.

Table 3 presents the classification accuracy and performance metrics of the proposed model and the equivalent results from competitive algorithms (one-class SVM, Isolation Forest and Minimum Covariance Determinant).

Figures 6, 7, 8 illustrate the tests performed in order to accurately interpret the high classification results that the proposed model achieves, as well as to prove the reliability of the method.

A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

In this case, we have the exact opposite situation. As the reconstruction error increases the recall decreases.

As stated in the *Deep Autoencoder* architecture, each transaction calculates the *Reconstruction Error* [21, 22], which is the key criterion for the safe calculation of a transaction as normal or abnormal. This is done based on a threshold which is indicative of the separation of classes and which is assumed to be called RET. In this specific task, RET was set to the value of 4.7, which came after exhaustive trials (trial and error) in order to give the system the optimal separation of classes.

The *Reconstruction Error Threshold* (RET) is concerned with calculating the error in class separation while training the Autoencoder algorithmic approach. Although these two thresholds have the same goal, in terms of optimal high-performance classification, they should not be confused. Figure 9 shows the reconstruction error for training and test data which seems to converge satisfactorily. This figure confirms the correct choice of the RET.

Finally, the reconstruction error for different classes as illustrated by RET is shown in Fig. 10. The RET designation was chosen to discard samples, most of which deviate from the target category, thereby strengthening the classifier. Even when discarded samples are normal, rejecting a small but representative percentage of them helps the classifier to become significantly stronger.

The testing hardware and software conditions for all simulations are listed as follows: Laptop Intel-i7 2.4G CPU, 16G DDR3 RAM, Ubuntu 18.04 LTS, Anaconda Keras/TensorFlow environment (Python). As it has been shown from the results and the analysis presented above, the proposed model is robust and it can successfully identify anomalies even in complex situations. Thus, it can be employed in actual cases of high complexity.

6 Contribution of the proposed approach

6.1 Conceptual characteristics of the proposed algorithm

The key features of the proposed algorithm, which give new functions and prospects to industrial production, are summarized below:

a. Quality

Providing high quality services with the active involvement of consumers.

b. Support

Activate and support new products, services and business prospects.

c. Reliability

Reliable and quality, uninterrupted service with resistance to natural and logical disasters.

d. Two-way communication

Collection and analysis of data with the capability of automated forecasting and real-time modeling of the production process, in all phases of the production chain, with dynamic feedback and communication.

e. Security

Infrastructure protection with surveillance, remote control and early warning systems.

f. Decentralization of services

Integrated scattered production from a variety of industrial automation

6.2 Applications of the proposed algorithm

Some decentralized industrial and construction applications that can take advantage of the proposed architecture are presented below.

a. On-Demand Manufacturing

The architecture of the proposed approach, enables *On-Demand Manufacturing* of services purchase, by ensuring a high-security environment which is based on advanced artificial intelligence technologies. Specifically, users of a *Blockchain* network will be able to trade with machines (Consumer-to-Machine Transactions—C2MT) which will have their own account on the network. In this way, they can enable production applications for custom-made products (e.g., 3D printing, raw materials processing). If different machine services are required to make a product, the machines will be able to send transactions to other machines. Also, Machine to Machine Transactions (M2MT) can be involved in organizing the production process by automatically interacting with the machines and exchanging messages about the products' readiness to move to the next stage of production.

b. Smart Diagnostics and Machine Maintenance

The reliable, direct communication provided by this architecture can be used to intelligently diagnose the functional condition of critical engineering equipment and to develop self-service applications. The equipment itself will be able to monitor current condition, to diagnose problems, to request replenishment of supplies and remote assistance from the maintenance team, or from production facilities and specialized diagnostic centers. Moreover, it will be able to perform autonomous scheduling of spare parts' requests, rebooting of equipment, shutdowns to replace problematic or obsolete hardware, software upgrades, and maintenance works' automation in general. All of the above will be easily and safely allowed.

c. Traceability

This architecture is also suitable for the development of traceability applications, related to industrial products and supply chain. Specifically, within an intelligent industrial environment, production logs can be kept between consumers and producers so that it is known, for example, which factory and in particular which machines at the plant were used to manufacture a particular product. In the case of defective raw material, manufacturing defects, material failure, after delivery of the products, traceability applications will be able to help identify the affected parts.

d. Product Certification

Another important application of the proposed architecture, is related to the performance of product certification. Manufacturing information of a product, such as machine details, date of manufacture, details of installation-production, raw materials, by-products or components and their corresponding manufacturing details, can contribute to the authenticity proof of the products. This can eliminate the need for physical certificates that may be prone to falsification.

e. Predictive Manufacturing

The collection, processing and analysis of large datasets related to the function of an industrial environment, is useful in predicting production, which is the most important issue for the development and optimization of industrial technology. This process allows manufacturers to maintain a competitive advantage in the operational control of managing their function, while improving the efficiency of their production and exploiting the manufacturing sector. These specialized processes are usually provided as *Outsourcing Big Data Analytics services*. In this case, the proposed architecture ensures the integrity of provided services and the professional confidentiality of the parties involved.

f. Tracking *Supplier Identity and Reputation*

Finally, another use of the proposed architecture can focus in the development of applications for the management of manufacturer, supplier and machine identities. Moreover, it can contribute significantly, in the management of reputation, related to a variety of performance parameters such as delivery times, customer reviews and supplier ratings.

7 Discussion

The hybrid architectural standardization and development of the proposed anomaly detection via BLO Deep learning smart contracts is employed by an intelligent cybersecurity monitoring, modeling and management system. It is based on intelligent methods that have been widely used by our research team [38–40]. This system implements sophisticated anomaly detection functions through the two-way bilateral agreement provided by Smart Contracts, taking advantage of the *Blockchain* network features. It thus ensures in the most efficient and intelligent way, secure network communication between the trading devices in the IIoT ecosystem. The proposed Deep Learning Smart Contract, which incorporates a sophisticated Deep Autoencoder, provides an intelligent mechanism that can

accurately classify harmful anomalies in IIoT transactions, which are mainly cyber-attacks. The proposed system, greatly enhances critical infrastructure security mechanisms, which are the main target of APT attacks. It also links in the most efficient and ambitious way, artificial intelligence and *Blockchain*, which are the two most important and timely areas of research. The implementation of the proposed approach, was based on the unary classification philosophy, based on which a Deep Autoencoder was trained on a dataset related only to normal IIoT behavior. The aim was to ensure the absolute validity of the classifier.

The performance of the proposed system was tested on a multidimensional dataset of high complexity, which emerged after extensive research into the operation of the IIoT devices (SCADA, DCS, PLC) and after comparisons, controls and tests. This was done in order to identify the most appropriate limits, which realistically express the operating conditions of these devices, separating them into normal or abnormal. The high precision results obtained after exhaustive testing, significantly enhance the employed methodology. It is important to note that the deployment of Artificial Intelligence in the *Blockchain* network, greatly enhances the security mechanisms of critical infrastructures and it creates new perspectives on how to tackle cyber piracy. The application of the proposed methodology can simplify and minimize the cost and the time to detect anomalies in the labyrinthine industrial networks. The extensive recording and collection of the above data is a prerequisite for the development of a risk management and prevention system, in order to protect critical infrastructures and the IIOT ecosystem.

The superiority of the proposed novel model focuses on the robustness, accuracy, and generalization ability that offer, as the overall behavior of the model is comparable than a corresponding one. Specifically, the proposed model reduces overfitting, decreases variance or bias, and without reducing significant the precision of the model can fit unseen patterns. This is a major innovation that significantly improves the overall reliability of the proposed novel model. Generally speaking, our main concern is to prove that this model produces remarkable results compared to theoretically superior models.

Although they are capable of learning complex feature representations and of reducing dimensionality, the largest pitfall of Autoencoders lies in their interpretability. It is impossible to visualize and understand the latent features of non-visual data. Also, they are computationally expensive and extremely uninterpretable, the underlying math is more complicated and prone to overfitting, though this can be mitigated via regularization. The mode relies more on the input data and it may be tuned.

Unfortunately, writing and running an algorithm such as the ones proposed herein, is highly complicated. Developing distributed smart contracts becomes a difficult task, due to the dependency on the distributed platform. Also, there are no standard measures to evaluate distributed algorithms.

On the other hand, one of the main disadvantages of the proposed system is its traditional performance. It is designed by assuming that data can be easily accessed, which means that the same data may be accessed many times. Also, the proposed system is not able to handle very large data sets (terabytes) and to develop efficient and scalable smart contracts, regarding accuracy and computation requirements (memory, time and communication needs). For example, if the computational complexity of the smart contract outpaces the main memory, then the algorithm will not scale well and will not be able to process the training data set or it will not run due to memory restrictions. A scalable learning algorithm can deal with any volume of data, without consuming ever-growing amounts of resources like memory.

8 Conclusions

This research paper presented an innovative, reliable, low-demand, and highly effective anomaly detection system, based on sophisticated computational intelligence methods that flow into the *Blockchain* network. The most important innovation of the proposed system is the strengthening of the *Blockchain* network by deploying artificial intelligence (AI), which does not behave as a supporting framework, but as an active structural component of the network. To the best of our knowledge, this model is introduced to the literature for the first time, and it is an important driver for further exploitation of intelligent technologies in the Blockchain network. It is the first *Deep Learning Smart Contract* network, which programmatically implements the bi-directional agreement, based on AI. Another very important innovation is the employment of a *Deep Autoencoder* and its individual configurations (e.g., the determination of RET) in the implementation of an anomaly detection system, solving a multidimensional and complex security problem related to the IIOT ecosystem.

Deep learning techniques, and in particular *Autoencoder* networks, simulate the function of biological brain cells in the most realistic way and they realistically model space time. They are capable of simulating complex functions, using abstractions, intermediate representations and feedback relationships, capturing multiple levels of operation and optimally matching input data to the desired network output responses. This makes perfect tuning of the target database possible and enables the development of highly

accurate classification or correlation models, which is very important in the detection of anomalies.

Moreover, innovation lies in the employment of AI to the level of analysis of the way real-time industrial equipment work. This fact significantly enhances critical infrastructure defense mechanisms. Checking the interoperability of IIoT devices at any time makes it much easier to detect serious cyber-attacks that can cause irreparable damage.

Finally, there is an innovation in the way data is collected and selected. This approach emerged after extensive research into the way IIoT operates and after comparisons, checks and tests on their inherent behavioral boundaries in order to classify them accordingly. An important task performed was the calculation of the ODT, which strongly supports the success of the classification.

Suggestions for development and future improvements of this system should focus on further optimizing the *Deep Autoencoder* parameters used. Thus, the proposed approach will become an even more efficient, precise and faster classification process, capable of separating even more precisely the boundaries between the states of the IIOT systems.

Also, it would be important for the proposed framework to be expanded by employing methods of self-improvement and automatic redefinition of its parameters (meta-learning). In this way, the full automation of APT attacks detection, will become possible.

An additional element that can be considered in a future extension of this application is the creation of an additional cross-sectional anomaly analysis system. This would work counter-diametrically to the *Deep Autoencoder's* philosophy of classification and it could operate additively to achieve the desired result for solving linear bandit problems in peer-to-peer networks with limited communication capabilities [41].

In addition, a future article would aim to provide readers with different angles to view several anomaly detection approaches such as Collaborative Filtering Bandits [42], Mining λ-Maximal Cliques from a Fuzzy Graph [43], the Art of Clustering Bandits [44], which are also in the same real-time data analytics in order to be compared on common grounds and they can be analyzed easily.

Moreover, future extensions-improvements should focus on further optimizing the parameters of the employed algorithm. This can be achieved by a big data scalable approach like Lambda architecture in a parallel and distributed data analysis system (Hadoop) or in projects like Spark and Stratosphere/Flink. In this way more useful abstractions can be achieved, that can enhance the scalability of a project.

Finally, the most important development in this proposal, is the addition of *Federated Learning* capabilities to the model, which will allow decentralized retraining of the system and upgrade of its classification capabilities with additional samples. These samples will be the new trades of IIoT devices based on *Deep Learning Smart Contracts*.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bassi L (2017) Industry 4.0: hope, hype or revolution? In: 2017 IEEE 3rd international forum on research and technologies for society and industry (RTSI), Modena, 2017, pp 1–6
2. Lee J, Bagheri B, Kao H-A (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems, vol 3, 2015, pp 18–23
3. Gilchrist A (2016) Industry 4.0: the industrial internet of things, Publisher: A press, Release Date: June 2016, ISBN: 9781484220474
4. Xu LD, Xu EL, Li L (2018) Industry 4.0: state of the art and future trends. Int J Prod Res 56(8):2941–2962
5. Zhou K, Liu T, Zhou L (2015) Industry 4.0: towards future industrial opportunities and challenges. In: 2015 12th international conference on fuzzy systems and knowledge discovery (FSKD), Zhangjiajie, 2015, pp 2147–2152
6. El Hamdi S, Abouabdellah A, Oudani M (2019) Industry 4.0: fundamentals and main challenges. In: 2019 international colloquium on logistics and supply chain management (LOGISTIQUA), Montreuil—Paris, France, 2019, pp 1–5
7. Culot G, Fattori F, Podrecca M, Sartor M (2019) Addressing industry 4.0 cybersecurity challenges. IEEE Eng Manag Rev 47(3):79–86
8. Sklyar V, Kharchenko V (2019) ENISA documents in cybersecurity assurance for industry 4.0: IIoT threats and attacks scenarios. In: 2019 10th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS), Metz, France, 2019, pp 1046–1049
9. Meneghello F, Calore M, Zucchetto D, Polese M, Zanella A (2019) IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. IEEE Internet Things J 6(5):8182–8201
10. Sample C, Schaffer K (2013) An overview of anomaly detection. IT Prof 15(1):8–11
11. Dixit M, Tiwari A, Pathak H, Astya R (2018) An overview of deep learning architectures, libraries and its applications areas. In: 2018 international conference on advances in computing, communication control and networking (ICACCCN), Greater Noida (UP), India, 2018, pp 293–297
12. Chen CLP (2015) Deep learning for pattern learning and recognition. In: 2015 IEEE 10th jubilee international symposium on applied computational intelligence and informatics, Timisoara, 2015, pp 17–17
13. Schmidhuber J Deep learning in neural networks: an overview. arXiv:1404.7828v4 [cs.NE]
14. Chatterjee R, Chatterjee R (2017) An overview of the emerging technology: blockchain. In: 2017 3rd international conference on computational intelligence and networks (CINE), Odisha, 2017, pp 126–127

15. Mermer GB, Zeydan E, Arslan SS (2018) An overview of blockchain technologies: principles, opportunities and challenges. In: 2018 26th signal processing and communications applications conference (SIU), Izmir, 2018, pp 1–4
16. Zhao S, Li S, Yao Y (2019) Blockchain enabled industrial internet of things technology. IEEE Trans Comput Soc Syst 6(6):1442–1453
17. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F (2019) Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans Syst Man Cybern Syst 49(11):2266–2277
18. Mohanta BK, Panda SS, Jena D (2018) An overview of smart contract and use cases in blockchain technology. In: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), Bangalore, 2018, pp 1–4
19. Abbas QE, Sung-Bong J (2019) A survey of blockchain and its applications. In: 2019 international conference on artificial intelligence in information and communication (ICAIIC), Okinawa, Japan, 2019, pp 001–003
20. Andersen MP, Kolb J, Chen K, Fierro G, Culler DE, Popa RA (2017) WAVE: a decentralized authorization system for IoT via blockchain smart contracts. Technical report no. UCB/EECS-2017-234, Dec 29, 2017
21. Schmidhuber Jürgen (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
22. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. http://www.deeplearningbook.org. Accessed 22 Feb 2019
23. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B (Stat Methodol) 67(2):301–320
24. Chen Z, Yeo CK, Lee BS, Lau CT (2018) Autoencoder-based network anomaly detection. In: 2018 wireless telecommunications symposium (WTS), Phoenix, AZ, 2018, pp 1–5
25. Fernández-Saúco IA, Acosta-Mendoza N, Gago-Alonso A, García-Reyes EB (2019) Computing anomaly score threshold with autoencoders pipeline. In: Vera-Rodriguez R, Fierrez J, Morales A (eds) Progress in pattern recognition, image analysis, computer vision, and applications. CIARP 2018. Lecture notes in computer science, vol 11401. Springer, Cham
26. Falco G, Caldera C, Shrobe H (2018) IIoT cybersecurity risk modeling for SCADA Systems. IEEE Internet Things J 5(6):4486–4495
27. Al-Dalky R, Abduljaleel O, Salah K, Otrok H, Al-Qutayri M (2014) A modbus traffic generator for evaluating the security of SCADA systems. In: 2014 9th international symposium on communication systems, networks & digital sign (CSNDSP), Manchester, 2014, pp 809–814
28. Fachkha C (2019) Cyber threat investigation of SCADA modbus activities. In: 2019 10th IFIP international conference on new technologies, mobility and security (NTMS), CANARY ISLANDS, Spain, 2019, pp 1–7
29. Grooby S, Dargahi T, Dehghantanha A (2019) Protecting IoT and ICS platforms against advanced persistent threat actors: analysis of APT1, silent chollima and molerats. In: Dehghantanha A, Choo KK (eds) Handbook of big data and IoT security. Springer, Cham
30. Pan S, Morris T, Adhikari U (2015) Developing a hybrid intrusion detection system using data mining for power systems. IEEE Trans Smart Grid. https://doi.org/10.1109/tsg.2015.2409775
31. Pan S, Morris T, Adhikari U (2015) Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. IEEE Trans Ind Inform 1:1. https://doi.org/10.1109/tii.2015.2420951
32. Pan S, Morris T, Adhikari U (2015) A specification-based intrusion detection framework for cyber-physical environment in electric power system. Int J Netw Secur 17(2):174–188
33. Beaver J, Borges R, Buckner M, Morris T, Adhikari U, Pan S (2014) Machine learning for power system disturbance and cyber-attack discrimination. In: Proceedings of the 7th international symposium on resilient control systems, Aug 19–21, 2014, Denver, CO, USA
34. Wenzhu S, Wenting H, Zufeng X, Jianping C (2019) Overview of one-class classification. In: 2019 IEEE 4th international conference on signal and image processing (ICSIP), Wuxi, China, 2019, pp 6–10
35. Zhang Shichao, Zhang Chengqi, Yang Qiang (2003) Data preparation for data mining. Appl Artif Intell 17(5–6):375–381
36. Mao J, Jain AK, Duin PW (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
37. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
38. Demertzis K, Kikiras P, Tziritas N, Sanchez SL, Iliadis L (2018) The next generation cognitive security operations center: network flow forensics using cybersecurity intelligence. Big Data Cogn Comput 2:35
39. Demertzis K, Tziritas N, Kikiras P, Sanchez SL, Iliadis L (2019) The next generation cognitive security operations center: adaptive analytic lambda architecture for efficient defense against adversarial attacks. Big Data Cogn Comput 3:6
40. Demertzis K, Iliadis L, Kikiras P, Tziritas N (2019) Cyber-typhon: an online multi-task anomaly detection framework. In: MacIntyre J, Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial intelligence applications and innovations. AIAI 2019. IFIP advances in information and communication technology, vol 559. Springer, Cham
41. Korda N, Szorenyi B, Li S (2016) Distributed clustering of linear bandits in peer to peer networks. In: ICML'16: proceedings of the 33rd international conference on international conference on machine learning, vol 48, June 2016, pp 1301–1309
42. Li S, Karatzoglou A, Gentile C (2016) Collaborative filtering bandits. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval (SIGIR'16). Association for Computing Machinery, New York, NY, USA, pp 539–548. https://doi.org/10.1145/2911451.2911548
43. Hao F, Park D-S, Li S, Lee HM (2016) Mining λ-maximal cliques from a fuzzy graph. Sustainability 8:553
44. Gentile C, Li S, Zappella G (2014) Online clustering of bandits. In: Proceedings of the 31st international conference on international conference on machine learning, vol 32 (ICML'14). JMLR.org, II–757–II–765