Detecting Robotic Anomalies using RobotChain

Vasco Lopes*, Luís A. Alexandre†
Universidade da Beira Interior, Instituto de Telecomunicações
Rua Marquês d'Ávila e Bolama, 6201-001, Covilhã, Portugal
*Email: vasco.lopes{at}ubi.pt
†Email: luis.alexandre{at}ubi.pt

***Abstract*—Robotic events can provide notable amounts of information regarding a robot's status, which can be extrapolated to detect productivity, anomalies, malfunctions and used for monitorization. However, when problems occur in sensitive environments like a factory, the logs of a machine may be discarded because they are susceptible to chances and malicious intents. In this paper we propose to use RobotChain for anomaly detection. RobotChain is a method to securely register robotic events, using a blockchain, which ensures that once an event gets registered on it, it's secured and cannot be tampered with. We show how this system can be leveraged with the module for anomaly detection, that uses the information contained on the blockchain to detect anomalies on a UR3 robot.**

I. Introduction

Registering robotic events with proof that they can't be tampered with is often disregarded because of the complexity of such systems, which leads to slow adoption. Insecurity breaches and file tampering can be done to protect robot manufacturers when their robots have abnormal behavior and jeopardize a factory production line. Blockchain is one answer to this problem. Since it's first idealization by Leslie Lamport in 1998 [1], to the implementation of this technology to serve as the base for Bitcoin [2], many have leveraged this technology to have a secure and validated data storage in a decentralized way. Many blockchain approaches have surfaced with different characteristics, but the underlying idea is practically the same: to have a way to store, share, distribute, compute and to monitoring data. Despite the security guaranties that this technology can bring to a system, it can also be very energy and time consuming to validate the transactions and to be apart of the consensus mechanism [3]. A naive implementation of a blockchain in sensitive environments, like a factory, could lead to stolen computation power from the robots resulting in lower productivity, ultimately leading to a disbelief in the technology.

Despite all the development conducted to improve blockchain technology and to boost the underlying smart-contract features, first introduced by Ethereum [4], there isn't much development in integrating robotics with it. Incipient work has been presented by Castelló et al. [5], which presents a framework for safe sharing of human-robot interactions. Strobel and Dorigo on their paper [6] introduce how the blockchain can be used to provide a mechanism to have a shared knowledge system in a robotic swarm. Strobel et al. also conducted promising research that can be used to integrate the blockchain in real word environments to solve byzantine fault problems [7] in robot swarms. Some companies have also started working on robotics over blockchain, such as Kambria [8], that is building a platform similar to Android but with focus on robotics, in which high-level libraries, modular hardware, and software are available in order to speed up robotic development. Although there are many methods and proposals using the blockchain technology, most of them are just focused on the token that it contains to have a financial interest, which leads to slow development of the technology for other purposes. There are however approaches that try to use blockchain within the robotics context and on a more high-level, by defining standards to it [9], [10], [11], [12], even though these are incipient works [13], we will see this technology surfacing in new fields as it has potential to store data, incorporate decentralized application among others.

The authors of [14] proposed RobotChain, a method that integrates the blockchain with Robotics and that can have extended use cases, like the usage of Artificial Intelligence to detect anomalies in robots and machinery as we present in this paper. They use the blockchain as a ledger to register all robotic events and validate them. This is what we propose in this paper: by having an oracle interacting with the smart-contracts that use the information that is inserted into the blockchain we will be able to detect anomalies. This method is useful as it provides a ledger that can register information from the robots in a way that it cannot be tampered with and a way of leveraging Artificial Intelligence methods over it, as it can have new modules integrated into the system without the need to redesign it. Our proposal integrates three technologies that are disrupting our daily life's in a unique system, Robotics, Artificial Intelligence, and Blockchain.

The document is structured as follows: section II does an overview of the state-of-the-art. Section III describes RobotChain. Section IV describes our proposal. Section V presents the experiences conducted to validate the method, and the last section, contains our conclusions.

II. Related Work

In [15], a method to detect anomalies based on summarization and data storage is proposed. By storing large amounts of data in an organized way, by using metadata to know the localization of the data and by compressing it with summarization techniques, the authors were capable of reducing the space required to store large amounts of data and improve the efficiency when searching for specific

contents when compared to traditional methods. Then, by using representations of the data, the authors cluster them to detect anomalies. The clustering process is done in two steps, the first one uses Clustream algorithm [16] in order to obtain representations of the acquired data. The second step is the application of X-means algorithm [17] to cluster the data incoming from step one. Finally, a threshold is compared with the results to define if they represent anomalies or not.

A method inspired in the Natural Immune System for Robot Anomaly Detection, with focus in detecting failures in autonomous robot systems is proposed in [18]. The system was tested with the aid of the robot OSCAR [19], which has 6 legs with 3 motors per leg. In this method, the authors use clonal selection, combined with fuzzy logic for representing the information. Fuzzy logic allows the method to categorize anomaly detection within a range of values. The method works by defining two sets of rules, one set of rules defines all the rules that detect when there is some anomaly present and the other set defines the rules that represent a state where an anomaly isn't present. These rules also have a "weight", which is the part of the Artificial Immune System (AIS) in the method. This weight serves to increment or decrement the value of the anomaly presence by a constant. The authors were capable of detecting anomalous situations in their experiments against the normal behavior of the robot.

Huimin Lu et al. presented a method that based on reinforcement learning, that can detect motor anomalies in drones [20]. The algorithm works by analyzing the information about the temperature change and deciding if the change is critical or not, indicating the presence of an anomaly. This analysis is complemented by using the information about the speed within the reinforcement learning algorithm to adjust the threshold temperature. This threshold indicates whether the drone should continue with the same speed or if it should decelerate or stop to cool down and check if the anomaly persists.

Dong Zang et al. [21] present a method that uses Markov Chain to extract an array of features, which they entitled "Markov Feature", which consists on a vector with all the probabilities of state transitions. The state transition frequency was extracted from the dataset, where the authors defined 4 regions of interest in order to detect abnormal occurrences. The dataset used to build and test their method consists on time series information of the pressure from experimental pipelines over the time. From the data, the authors also extract the mean and the variance and use them as features. With these features and data, the authors used 4 different algorithms, k-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machine. All of the algorithms surpassed the classic methods of feature extraction, i.e., Statistic-based, Wavelet-based and time and frequent domain method.

The work presented by Wallace Lawnson et al. [22] uses Generative Adversarial Networks (GAN) [23] to identify indoor environment anomalies. The GAN was trained by teleoperating a robot within an indoor environment in which the images were split into patches and then used to train the algorithm. By doing so, the authors were able to create a method capable of locating anomalies in indoor environments without the need to store images for comparative analysis.

HuiKeng Lau et al. developed an immuno-engineering approach to detect anomalies in robotic swarms [24]. The work focus on creating a simulation of a robotic swarm containing 8 robots in which their task is a collective work to pick food. In this simulation, the authors induced anomalies and show that the individual performance of the robots, in a swarm environment, suffered when one robot had an anomaly and that anomalies that influence the whole swarm can be detected.

In [25], the authors generated a dataset with the Mackey-Glass equation and by numerically solving it using fourth-order Runge-Kutta method. With this, the authors generated 1500 normal samples and discarded the first 1000 to eliminate the "initial value effect" and did the same to generate abnormal samples by changing one of the equation parameters within 100 of the final samples. Then, compared a simple Multi-layer Perceptron approach against a Self-organizing Map trained only on normal samples and concluded that both methods achieve similar results in the detection of the abnormal samples.

III. RobotChain

RobotChain is a system to register robotic events in a trust and secure way [14]. By having a blockchain as a ledger, it's ensured that every event sent to the network is validated and if inserted in the blockchain, it's valid and secure. This blockchain is based on the one developed by Tezos [26], [27] as it incorporates characteristics that are important for systems that are intended to work in sensitive environments like RobotChain. The major properties of this blockchain technology are the support for formal verification of the code and the self-amending property that allow to perform changes on the blockchain by voting on-chain, without the need to conduct hard-forks when a core change needs to be executed. Tezos developed a new smart-contract language called Michelson, which is a stack-based and strongly typed language that facilitates formal verification. The formal verification of code ensures that it always does what it's supposed to do, regarding the specifications. This blockchain uses as a consensus algorithm the Delegated Proof-of-Stake, which is more energy efficient than Proof-of-Work and it also as the important property that not everyone needs to be a baker (miners are called bakers on Tezos), which is important when we think on a heavily energy and time-dependent environment like a production line. RobotChain also has modules upon it, as it's a modular approach, decentralized applications and other systems can be built using the information that it's on the blockchain. In this paper, we propose a method for detecting anomalies on robots illustrating the creation of a module on top of RobotChain. This method leverages smart-contracts to acquire information about a robot and by having an oracle computing this data off the chain, it's capable of detecting when anomalies occurred in a specific robot.

The blockchain in this system is a consortium blockchain [28], which consists of a private blockchain where the

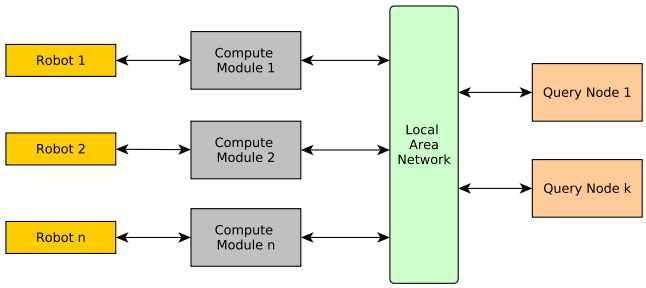

Fig. 1. Description of how the RobotChain works. Figure from [14].

nodes are from multiple manufacturers. This type of use for blockchain also leverages from one important characteristic present in Tezos' blockchain, which is the self-amendment process and the ability to change rules, like the consensus algorithm, without the need for hard-forks, which ultimately adds a layer of security [29].

The creation of the private blockchain was done by cloning the main net of Tezos and conducted several changes to it. These changes were made in an attempt to understand inner workings and improve it in the context of a consortium blockchain. The main changes were the addition of a transaction parameter named "Transaction Description", providing a field to document the transaction or a smart contract call, allowing texts on transactions. The second change was the parameters of smart-contracts storage, which were hard-coded to have a small limit and were increased in order to store more information on the smart contracts. The other modification made to the network was the change of the genesis public key, allowing the creation of a private network in a controlled environment. A representation of how RobotChain works is presented in Fig. 1, where a blockchain working over the Local Area Network is shown, where Robots are coupled to computing devices that insert values into it and it's also possible to query the blockchain to read the values it contains.

IV. Our Proposal

To register the robotic events we used the storage of a simple smart contract, which is called by the computer coupled to a robot that adds to the storage the new information about the robot joints. The smart-contract used for this was written in Liquidity, which is a high-level language that is compiled to Michelson, the Tezos main smart-contract language. Liquidity is useful as it's statically-typed, which reduces the number of induced errors by the programmer and as it has a compiler to Michelson and a decompiler that translates Michelson to Liquidity, it becomes easier to audit smart-contracts. The smart-contract written had only a method and a storage variable. The storage is a list that stores strings. The method serves the function of receiving a string, checks if the caller of the smart-contract is the owner of the smart-contract, and if so, appends the new string to the storage.

Using RobotChain and a real UR3 robot [30], we created a scenario where it's possible to detect anomalies. The scenario created was a pick and place task performed by the UR3, where the robot has to pick the soap, place it in a different position, return to the home position and then do the reverse by putting the soap in the initial place and returning to the home position. The main reason to illustrate this method on such a task is due to the fact that this type of task is one of the most common in factory environments, where robots are used to transport materials from one place to another, e.g. pick a door from a conveyor and place it on a car. The chain of events of the described task is presented in Fig. 2, where the images represent parts of the movement the robot does. We connected the robot to the Robot Operating System (ROS) and with the ur_modern_driver and the universal_driver packages which are compatible with ROS, we were able to receive information at a rate of $125Hz$ regarding effort, velocity and position of the 6 joints the robot has. With this system, we acquired information about the robot for 25 complete sequences tasks. In the last one, we induced anomalies by holding the robot for brief seconds, counteracting its movement, enforcing the need to adjust the robot joints to correct its behavior. This enforced anomaly is used to simulate when a robot suffers from some internal failure, weather by faulty components or by power issues, or by colliding with another object.

The acquired information consists of multiple signals that contain noise, as it's acquired using a real robot, so, to remove part of the noise, we smoothed every point of the data with the following filter:

$$g(i) = \begin{cases} 1/3 \sum_{k=i-1}^{i+1} f(k), & i \in \{2, \ldots, n-1\} \\ f(i), & \text{otherwise} \end{cases} \quad (1)$$

Where $i = 1, ..., n$, n is the number of samples on each signal, $f(i)$ represents sample i from one of the collected signals and $g(i)$ its smoothed version.

As the dataset contains multiple repetitions of the same task, we used Autocorrelation to find the period of each signal in order to create a model of it. We then divided the dataset into two separate ones, the initial 80% of the data for the train set and the rest for the validation set. Using the train set, we created a model for each signal, where the model is the mean of each point for the different periods. This ensures that we have a representation of the signal that is robust and at the same time it's low computation cost approach. For each point, we also determined the standard deviation. In the end, the model to tackle the anomaly detection consisted of a new signal for the type of signals contained in the dataset and a vector with length equal to the period size that contains the values of the standard deviation for each point of a period. The creation of this model is an unsupervised task, as we don't label the data.

To detect anomalies, we compare a period with the model signals and we use the following equation for detection:

$$\mathbf{A}_{j,i} = \begin{cases} 1, & \text{if } \mathbf{S}_{j,i} > \boldsymbol{\mu}_i + k\boldsymbol{\sigma}_i \\ 1, & \text{if } \mathbf{S}_{j,i} < \boldsymbol{\mu}_i - k\boldsymbol{\sigma}_i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

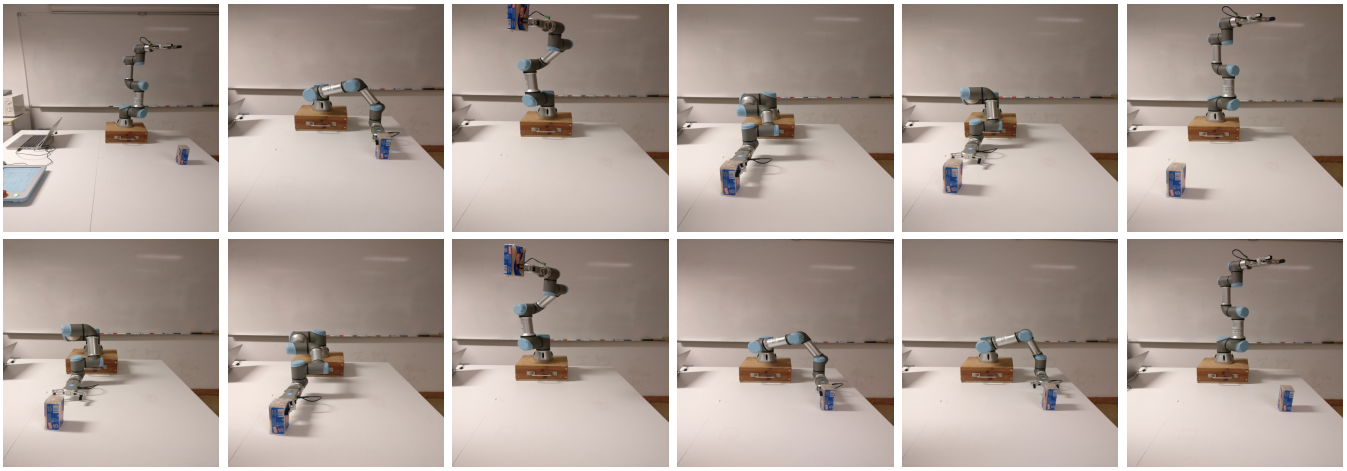

Fig. 2. Pick and place task. Robot start in the home position, picks the object of position 1, leaves it on position 2, returns to the home position and then does the inverse, picking the object of position 2, dropping it on position 1 and returning to home position.

Where j represents the columns (different signals), i represents the indexes (rows), $\mathbf{A}$ is a variable that stores the existence of anomalies, $\mathbf{S}$ contains the values of the signal that is being checked for anomalies, $\boldsymbol{\mu}$ contains the model signals, k is a constant and $\boldsymbol{\sigma}$ is a vector that contains the standard deviations for each point.

Equation (2) considers as anomaly each point that deviates from the mean k times. This constant is useful to fine-tune the model to reduce the false-positives detected, as a higher k defines that points are considered anomalous only if they deviate from the mean with higher values than with a smaller k. In the following section, we explain in detail the experiences conducted to adjust the k and to validate the method.

V. Experiences

By using the equation (2) with the train set and by validating it with the validation set, we concluded that $k = 16$ was the best value to detect the anomalies and at the same time assure the smallest number of false positives as possible, which was done by evaluating the differences between the different signals and experimenting different k values. By using the train set we also perceived that some points on the model have a standard deviation close to zero, which implies that a small variation on a new signal in that point has a high chance of being detected as anomalous. By focusing on the effort signals of the joints, we can see an example of this problem occurring in Fig. 3. To solve this problem, we defined that an anomaly only occurs if there are more than 5 detections in a range of 30 points. This is defined as so because the rate at which the robot publishes the information $125Hz$, so an appears in multiple sequential points, as the robot adjusts to compensate the occurrence.

Then, after adjusting the model, we used the validation set to verify its efficiency. The validation set contains 20% of the initial data, which translates into 5 periods of the signal, and as the data acquired represents a time series data, none of the points present in the validation set were used to create the model. In this set, the first periods contains no anomalies, and the last one contains anomalies that were induced as mentioned before. This can be seen in Table I, where each one of the values represents the Mean Squared Error (MSE) of the signal model against the signals in the validation set. In this, we can see that the MSE is higher in the last period, where the anomalies are located, except for the last joint, which has similar MSE for all periods because this joint is the closest one to the gripper and does not suffer from the anomalies induced. By using equation 2 with $k = 16$, the method was capable of detecting every anomaly present in the validation set and had 0 false-positive, which means that the method only detected anomalies in the last period of the validation set, which is the one containing anomalies. The detection of the anomalies is shown in figure 4 for the joint 2.

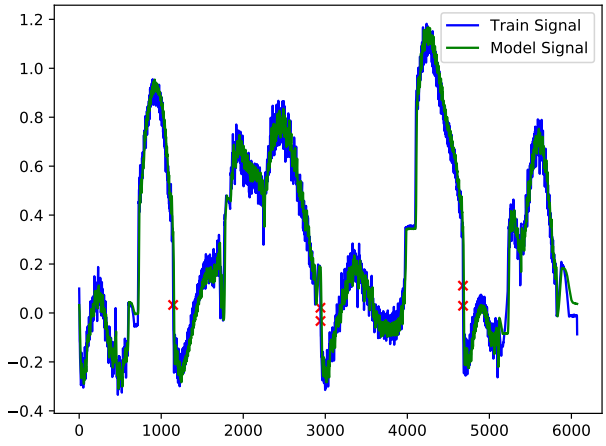

Fig. 3. False-positive anomaly detection example by comparing effort signal of joint 4 with the model signal for effort in joint 4. Anomalies are represented as Xs.

To ensure that the method proposed is suitable for the

TABLE I

MSE OF THE DIFFERENT PERIODS CONTAINED IN THE VALIDATION SET WITH THE MODEL SIGNAL. ONLY THE EFFORT FOR EACH JOINT SIGNAL IS CONSIDERED.

Period	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
1	0.0018	0.0037	0.0039	0.0013	0.0008	**0.0041**
2	0.0023	0.0048	0.0045	0.0017	0.0013	0.0021
3	0.0022	0.0048	0.0043	0.0019	0.0014	0.0018
4	0.0026	0.0061	0.0053	0.0023	0.0013	0.0032
5	**0.0181**	**0.2626**	**0.1470**	**0.0420**	**0.0036**	0.0039

TABLE II

MSE FOR THE DIFFERENT PERIOD OF THE TWO TEST DATASETS REGARDING THE EFFORT SIGNAL. THE DATASET A REPRESENTS THE TEST SET WITH NO ANOMALIES AND THE DATASET B REPRESENTS THE TEST SET WITH ANOMALIES.

Dataset	Period	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
A	**1**	0.0011	0.0020	0.0024	0.0013	0.0017	0.0011
	2	0.0021	0.0030	**0.0037**	0.0017	0.0019	0.0023
	3	**0.0040**	**0.0090**	0.010	**0.0047**	**0.0030**	**0.0071**
B	**1**	0.0145	0.4037	0.1203	0.0034	0.0022	**0.0043**
	2	0.0121	0.5355	0.1590	0.0044	0.0020	0.0021
	3	**0.0236**	**0.6131**	**0.2180**	**0.0618**	**0.0056**	0.0042

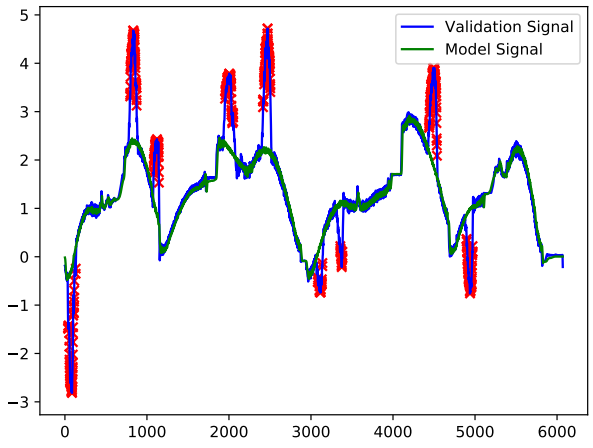

Fig. 4. Detection of anomalies in the validation set where anomalies are present. Anomalies are represented as Xs.

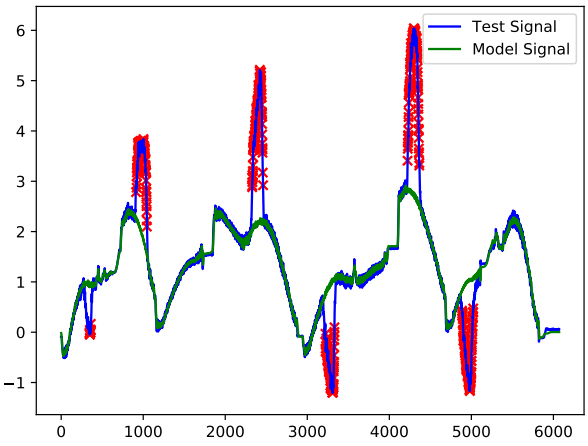

Fig. 5. Detection of anomalies in effort signal of joint 2 in period 1 of the test dataset that contains anomalies.

problem, we acquired two new datasets with the system aforementioned and used it to test the model created for detecting anomalies. The first test dataset contains no anomalies, while the second one contains anomalies in every period. We used cross-correlation to ensure that when comparing the signals, we are comparing the right points. So, with the correlation of both signals, we know how much data to remove at the beginning of the new datasets so that every signal starts at the same point as the signal models. In table II we present the values of the MSE per period per test dataset, A represents the test set with no anomalies and the dataset B represents the test set with anomalies. The MSE value is useful to check if there are anomalies in our experiments, but it's not a robust method and for other anomalies, it can dilute the anomalies becoming impossible to perceive if there are any. Using equation 2 in both these datasets, no anomalies were found in the first one and in the second one the method was capable of detecting the anomalies in all off the periods. An illustrative example of this detection can be seen in Fig. 5.

VI. CONCLUSION

This paper describes the creation of a module of RobotChain, a novel method to store robotic events in a secure way. By using a blockchain, we increase the security of the data that is registered in an environment that is susceptible to human interaction. This ensures that if a robotic failure occurs, it gets registered in the blockchain and once there, it can't be manually altered. By having such a system, we use smart-contracts to store the information about a UR3 robot in the blockchain and created four datasets, one train, one validation, and two test sets to develop, validate and test a method that can leverage the information acquired from the blockchain to detect robotic anomalies. Our method for detecting anomalies was capable of detecting anomalies induced by counteracting the movement of the arm while performing a pick and place repetitive task. This method shows that it's possible to use the blockchain with robotics and with such a modular system, innovative methods for different purposes can be added by using oracles.

ACKNOWLEDGMENTS

This work was partially supported by the Tezos Fundation through a grant for project Robotchain.

REFERENCES

[1] L. Lamport, "The part-time parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, May 1998. [Online]. Available: http://doi.acm.org/10.1145/279227.279229

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Www.Bitcoin.Org*, p. 9, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, June 2017, pp. 557–564.

[4] V. Buterin, "A next-generation smart contract and decentralized application platform," *Etherum*, no. January, pp. 1–36, 2014. [Online]. Available: http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

[5] E. C. Ferrer, O. Rudovic, T. Hardjono, and A. Pentland, "Robochain: A secure data-sharing framework for human-robot interaction," *CoRR*, vol. abs/1802.04480, 2018.

[6] V. Strobel and M. Dorigo, "Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation," IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2018-009, May 2018.

[7] V. Strobel, E. Castelló Ferrer, and M. Dorigo, "Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 541–549. [Online]. Available: http://dl.acm.org/citation.cfm?id=3237383.3237464

[8] Kambria, "Fueling the robotics economy," Kambria, Tech. Rep., accessed: 2018-10-17. [Online]. Available: https://kambria.io/Kambria_White_Paper_v2_20180615.pdf

[9] Teslya, Nikolay and Smirnov, Alexander, "Blockchain-based framework for ontology-oriented robots' coalition formation in cyberphysical systems," *MATEC Web Conf.*, vol. 161 EDP Sciences, p. 03018, 2018. [Online]. Available: https://doi.org/10.1051/matecconf/201816103018

[10] T. Marwala and B. Xing, "Blockchain and Artificial Intelligence," *arXiv preprint arXiv:1802.04451*, p. 13, 2018. [Online]. Available: http://arxiv.org/abs/1802.04451

[11] J. Chen, K. Duan, R. Zhang, L. Zeng, and W. Wang, "An AI Based Super Nodes Selection Algorithm in BlockChain Networks." [Online]. Available: https://arxiv.org/pdf/1808.00216.pdf

[12] E. C. Ferrer, O. Rudovic, T. Hardjono, and A. Pentland, "RoboChain: A Secure Data-Sharing Framework for Human-Robot Interaction," *eTELEMED conference*, 2018. [Online]. Available: http://arxiv.org/abs/1802.04480

[13] V. Lopes and L. A. Alexandre, "An overview of blockchain integration with robotics and artificial intelligence," *CoRR*, vol. abs/1810.00329, 2018. [Online]. Available: https://arxiv.org/abs/1810.00329

[14] M. Fernandes and L. A. Alexandre, "Robotchain: Using tezos technology for robot event management," in *Symposium on Blockchain for Robotic Systems, MIT Media Lab*, Cambridge, MA, USA, December 2018.

[15] A. Bagozi, D. Bianchini, V. De Antonellis, A. Marini, and D. Ragazzi, "Big data summarisation and relevance evaluation for anomaly detection in cyber physical systems," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2017, pp. 429–447.

[16] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th international conference on Very large data bases-Volume 29*. VLDB Endowment, 2003, pp. 81–92.

[17] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters." in *Icml*, vol. 1, 2000, pp. 727–734.

[18] B. Jakimovski and E. Maehle, "Artificial immune system based robot anomaly detection engine for fault tolerant robots," in *Autonomic and Trusted Computing*, C. Rong, M. G. Jaatun, F. E. Sandnes, L. T. Yang, and J. Ma, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 177–190.

[19] B. Jakimovski, M. Litza, F. Mösch, and A. E. S. Auf, "Development of an organic computing architecture for robot control," in *GI Jahrestagung*, 2006.

[20] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2315–2322, Aug 2018.

[21] "Markov chain-based feature extraction for anomaly detection in time series and its industrial application," *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 1059–1063, 2018.

[22] W. Lawson, E. Bekele, and K. Sullivan, "Finding Anomalies with Generative Adversarial Networks for a Patrolbot," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 484–485, 2017.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[24] H. Lau, I. Bate, and J. Timmis, "An immuno-engineering approach for anomaly detection in swarm robotics," in *ARTIFICIAL IMMUNE SYSTEMS, PROCEEDINGS*, P. Andrews, J. Timmis, N. Owens, U. Aickelin, E. Hart, A. Hone, and A. Tyrrell, Eds., vol. 5666 LNCS. SPRINGER-VERLAG BERLIN, 2009, pp. 136–150.

[25] F. Gonzalez and D. Dasgupta, "Neuro-immune and self-organizing map approaches to anomaly detection: a comparison," *Proceedings of the First International Conference on Artificial Immune Systems*, pp. 203–211, 2002.

[26] L. M. Goodman, "Tezos - a self-amending crypto-ledger," pp. 1–18, August 2014.

[27] L. M˙ Goodman, "Tezos - white paper," no. July 2016, pp. 1–17, September 2014.

[28] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, Aug 2018.

[29] C. Natoli and V. Gramoli, "The balance attack or why forkable blockchains are ill-suited for consortium," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2017, pp. 579–590.

[30] "Ur3 robot," https://www.universal-robots.com/products/ur3-robot/.