Amaurotic Entity-Based Consensus Selection in Blockchain-Enabled Industrial IoT

Riya Tapwal, Pallav Kumar Deb, *Graduate Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Surjya Kanta Pal

***Abstract*—In this paper, we propose a dynamic consensus-based blockchain system – A-Blocks – for efficiently managing the data produced by the sensors in an Industrial Internet of Things (IIoT) environment. Typically, industries deal with a heterogeneous set of data from a diverse range of sensors. Conventional blockchain adoptions are a popular choice in such scenarios for data security while satisfying both transparency and immutability. However, stringent consensus algorithms are inadequate for managing heterogeneous data, especially due to its implicit constraints. For instance, while PoW provides inevitable security and is highly distributive, it is not scalable and requires more energy. In contrast, PoS is energy-efficient but has reduced scalability and PBFT is suitable for faster processing. A-Blocks exploits the features of the available consensus algorithms and dynamically selects the best one in real-time. It operates in two phases: 1) Categorizing the data into groups based on their traits and then 2) Selecting the appropriate consensus algorithm. Extensive experimental results using open industrial datasets demonstrate the effectiveness of A-Blocks with 8% CPU and 78% memory consumptions on resource-constrained devices. Further, compared to existing methods, although A-Blocks increases energy consumption by 11%, it also reduces mining time by 7%.**

***Index Terms*—Blockchain, Consensus algorithms, Agglomerative Clustering, Industrial Internet of Things.**

I. Introduction

Deployments in Industrial Internet of Things (IIoT) environments involve the exchange of sizeable heterogeneous and sensitive data over the Internet (both private and public). Data security in such scenarios is a mandatory obligation to avoid data breaches and facilitate smooth processing. The adoption of cryptographic methods may not be suitable in such scenarios due to compulsory secured key exchange, and iterative encryption and decryption overheads. Further, the breaking point of the cryptographic methods depends upon the length of the key, which also adds to the tradeoff of managing long keys. For instance, operations in AES-128 and AES-256 needs 10 and 14 rounds, respectively. To organize the divergent data and provide security, blockchain is a one-stop solution that ensures security, immutability, and transparency. However, traditional blockchain with a single consensus algorithm is not suitable in dynamic scenarios such as IIoT. This is because data produced by the sensors may vary in terms of sampling rate, entropy, volume, and range. Stringent consensus mechanisms fail to exploit the advantages of the others, which leads to decisions that may not be suitable for the incoming data causing daunting effects on delay, power consumption, and processing routines. In summary, a system that employs consensus algorithms according to the nature of the incoming data is beneficial.

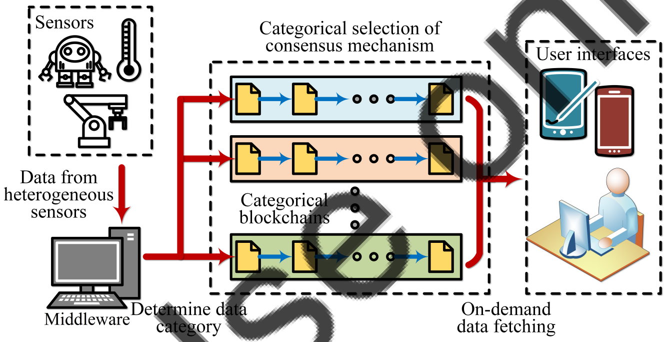


Figure 1: Overview of the proposed A-Blocks system and dynamic selection of consensus mechanisms.

In this work, we propose A-Blocks, a method for dynamically selecting a consensus routine based on the attributes of the incoming data in real-time for blockchain-enabled IIoT environments. As a proof of concept, we consider the Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT) consensus methods. In A-Blocks, the sensors in an environment such as in Fig. 1 produce a heterogeneous set of data and forward it to the blockchain system through a middleware (a gateway connecting the machines to the blockchain). The middleware forms clusters based on the variability, sampling rate, and range of the incoming data. The middleware is also responsible for the selection of the appropriate consensus mechanism and the creation of the blockchain. The authorized users may then fetch the data from the appropriate blockchain (refer Fig. 1). The proposed A-Blocks method is not limited to industrial scenarios and may be extended to smart cities, smart homes, healthcare, power plants, and security surveillance deployments for efficient data management and secure sharing.

A. Bias for clustering and dynamic consensus mechanism

The incoming data from sensors in a typical IIoT deployment is often subject to problems due to incomplete, corrupt, and incorrect formats, leading to inconsistencies. The attribute correction method-based clustering approach [1] helps in overcoming such issues with ease by removing the outliers from the clusters. This eases the decision on the appropriate

R. Tapwal, P. K. Deb, and S. Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. e-mail: tapwalriya@gmail.com, (pallv.deb,sudipm)@iitkgp.ac.in

S. K. Pal is with the Centre of Excellence in Advanced Manufacturing Technology, Indian Institute of Technology Kharagpur, India. e-mail:skpal@mech.iitkgp.ernet.in

consensus mechanism. The dynamic consensus mechanism helps in exploiting the benefits of multiple consensus algorithms in a single system such as scalability, enhanced security, and decentralization and a single consensus mechanism is inadequate for managing the trade-offs between them.

B. Motivation

Managing the data from a plethora of devices in an IIoT environment is challenging. These data are useful for making inferences in applications such as tracking jobs, defect identification, quality assurance, and maintaining supplier relations. While industries deploy various types of sensors to fetch data from the devices, the data is often divergent which increases the difficulty of removing inconsistencies and redundancies. This motivates us to segregate data into clusters with similar attributes. Moreover, there is a need for more decentralization, transparency, and scalability in industries and blockchain with a single consensus algorithm cannot provide this as there is trade-off between scalability, decentralization and security. It is also not efficient to store heterogeneous data using blockchain with a single consensus algorithm. These issues motivate us in developing the proposed A-Blocks system with a dynamic consensus algorithm for 1. storing the heterogeneous data efficiently in different blockchains with varying consensus mechanisms 2. ensuring low mining and response time.

C. Contribution

In this work, we propose A-Blocks, which is a dynamic blockchain system for the efficient management of data produced by sensors. The heterogeneous sensors produce variant data, which causes inconsistencies, and one solution for all does not suffice in handling all of them. We apply clustering to identify the category of incoming data and forward it to the blockchain using a suitable consensus mechanism. The major highlights of this work are as follows:

- ***A-Blocks:*** We propose A-Blocks for dynamically selecting the consensus mechanisms based on the attributes of data which is necessary for storing heterogeneous data efficiently by reducing the response and mining time. Further, A-Blocks eliminates the trade-off between scalability, security and decentralization by selecting the appropriate consensus algorithm.
- ***Data Categorization:*** We adopt agglomerative clustering to identify the category of the data using its attributes such as values, variability, and sampling rate. This is necessary for providing the best match of consensus mechanism according to the above mentioned attributes.
- ***Robustness:*** A-Blocks dynamically selects the consensus mechanisms according to the requirement of data and is independent of deployment architecture. It is suitable for every type of data that make it feasible for the industries incorporating with heterogeneous data.
- ***Evaluation:*** We discuss our observations while deploying A-Blocks on the resource-constrained devices such as a network of Raspberry Pis and demonstrate its advantages over conventional blockchain practices.

It may be noted that, in this work, we focus on the data-centric selection of the consensus mechanism in the blockchain and refrain from considering pricing policy and storage mechanisms. We plan to include the same in our extended work.

The organization of the rest of the paper is as follows. Section II contains the various research works done by the researchers in the field of clustering and blockchain. Section III describes the system model and the proposed solution. Section IV discusses the experimental setup, results, and the comparison of the proposed approach against other approaches, followed by the conclusion in Section V.

II. Related Work

A. Clustering and Blockchain for IIoT

Jia *et al.* [2] proposed a clustering method that clusters data based on a subset of attributes. He considered both numerical and categorical data and presented an attributed weighted clustering model to improve the stability and accuracy of data. Liang *et al.* [3] proposed a model to estimate the number of clusters. They proposed a hyper-correlation structure that affirms the compactness between the samples. Based on the triplet relationship, he calculated clustering assignments.

Pass *et al.* [4] proposed fruitchain protocol that did not store data directly into the blocks rather put the same inside the fruits. To access the data from the fruits, we need to solve the proof of work with different hardness parameters from the PoW used by the blocks. Bai *et al.* [5]proposed a blockchain system for IIoT with an off-chain network and on-chain network. They used an on-chain network to process all the transactions, such as providing permission and adding a digital signature. Rather, he used an off-chain network for storage and solving the problem which is not solved by blockchain. Further, Wang *et al.* [6] proposed chain-splitter, where the majority of the data is present at the clouds. They proposed hierarchical storage where only the recent blocks of the blockchain are present at the overlay network and rest at the cloud. Elli *et al.* [7] proposed a distributed operating systems for blockchains that follows execute-order-validate paradigm. Only the subset of peers executed the transactions, and whenever the client received enough endorsements, it assembled the transactions and submitted them to the ordering phase. Biswas *et al.* [8] proposed Proof of Block and trade before committing them. He used a lightweight consensus algorithm which is based on the miners present in the session. Xu *et al.* [9] proposed origin-chain having three types of users, including the administrator, traceability provider, and service user. They created three layers, i.e., management layer, off-chain, and on-chain. On-chain is responsible for implementing consensus, whereas off-chain is responsible for storing data. Lallas *et al.* [10] proposed a blockchain ledger for effective communication between the IoT devices and also improved supply chain services. The proposed framework is integrated with the decentralized cloud network of Industry 4.0. The proposed architecture hides the heterogeneity of application by acting as an upper abstraction layer, making the supply chain network more efficient. Further, Dua *et al.* [11] proposed a solution for managing e-waste by using 5G enabled blockchain.

Blockchain is used to keep track of e-waste generated and to dispose of it in an environment-friendly manner. Lamass and Carames [12] structured the benefits of using blockchain for Industry 4.0 to enhance transparency, robustness, data integrity, security, traceability, and anonymity. Similarly, Huang *et al.* [13] proposed a blockchain system using a credit-based consensus algorithm. In the proposed work, the proof of work is based on a credit system. Further, directed acyclic graph-structured blockchains are used to improve efficiency.

B. Blockchain for secure data sharing

Misra *et al.* [14] proposed a blockchain for boosting security in the IoT environment. An encrypted networked clock is implemented within the blockchain to synchronize non-real-time IoT nodes. Wang *et al.* [15] proposed two-layer blockchain architecture which used Byzantine Fault Tolerance as a consensus algorithm and combined various local blockchains with one state blockchain. Lin *et al.* [16] integrated the attribute signature with blockchain to provide fine-grained access control and mutual authentication. They provided confidentiality by using multi-receiver encryption. Srinivas *et al.* [17] proposed LBRAPS, based on one-way hash and bitwise XOR to secure data from various attacks. He used it to make the RFID system secure and ensured that RFID data should not be leaked. Further, Bera *et al.* [18] proposed a grid system for securely providing services by forming peer to peer network. The peer nodes gathered the data and added the same to the blockchain. Further, a system called BOSMOS for software status monitoring based on blockchain is proposed by Sen Hi *et al.* [19]. The system's snapshot is stored in the blockchain, which is used to identify malicious behavior. A method for monitoring the status based on the block hashing chain is proposed. Both, PoW and PBFT algorithms are applied to explain the working of the blockchain.

C. Synthesis

Blockchain, when applied to IIoT, suffers from various challenges such as managing heterogeneous data, trade-off between scalability, security, and decentralization. The data from the sensors present is subject to non-uniformity in terms of variability, sampling rate, volume, and others. Conventional blockchain-based practices in literature do not consider variations in these data. Data security in such scenarios is a mandatory obligation to avoid data breaches and facilitate smooth processing. Utilization of cryptographic methods results in extra overhead of iterative encryption and decryption. Also, the dynamic requirements of heterogeneous data, in terms of processing power, security, and energy requirements are not fulfilled by a single consensus algorithm. As shown in Table I, the existing works do not consider the selection of consensus mechanisms dynamically based on the requirement of data as well as there is also a trade-off between heterogeneity, scalability and energy consumption. In this paper, we propose a method for selecting consensus algorithms dynamically which considers the above mentioned features in a single system. We perform agglomerative clustering to segregate the data having similar traits into groups and apply blockchain with a dynamic consensus algorithm based on the nature of the incoming data from sensors.

Table I: Difference of A-Blocks with some existing works.

Paper	**Dynamic consensus**	**Heterogeneity**	**Energy requirements**	**Scalability**
Lallas *et al.* [10]	×	✓	×	×
Wang*et al.* [15]	×	✓	✓	✓
Biswas *et al.* [8]	×	×	✓	✓
Elli *et al.* [7]	×	×	✓	×
Xu *et al.* [20]	×	×	×	✓
A-Blocks	✓	✓	✓	✓

III. System Model

In this section, we introduce the network architecture, delay model, energy model of the system, preliminaries required to propose the scheme, and then propose a method to select the consensus algorithm dynamically based on data traits.

A. Network Architecture

We consider a set of sensors to sense the data produced by various IIoT devices and represent them as $S = \{s_1, s_2, s_3, \ldots, s_n\}$. These sensors sense data and forward it to the middleware for categorizing into groups with similar traits. The clustering layer (middleware) organizes the data into various clusters based on its attributes such as sampling rates, value ranges of data produced by multiple sensors, and variability. We use agglomerative clustering to categorize the data into various groups. The clustering layer (middleware) further forwards the data to the blockchain system, which selects the appropriate blockchain from the set of blockchains $B = \{b_1, b_2, b_3, \ldots, b_n\}$ implementing various consensus algorithms. We use PoW, PoS, and PBFT consensus algorithms for implementing the blockchains as proof of concept. We select consensus algorithms based on the attributes of data. Since PoW provides ultimate security, PoS is energy-efficient, and PBFT is fast. Therefore, the data which require extreme security use PoW, the data which require more energy use PoS. In contrast, the information high sampling rate, use PBFT. The authorized user uses the data from the blockchain and utilizes it in various applications such as job tracking, creating stocks, fault tracking, and establishing relations with suppliers.

B. Delay Model

The delay in offloading data depends on the delay in uploading the data and downloading the data. We only upload data in our architecture and consider the delay is downloading as 0. Further, the delay in uploading the data depends upon the transmission delay, propagation delay, queuing delay, and processing delay. The sensors transmit the data to the next layers and we calculate the delay involved in this as $T^{\theta}_{trans} = l^{\theta}/d_{u,v}$, where l^{θ} is the data size of θ and $d_{u,v}$ is the data rate between two layers u and v. We further calculate the data rate as $d_{u,v} = \beta log_2(1 + p_u g_{u,v}/(\sigma^2 + I_{u,v}))$, where β is the bandwidth of the channel, $g_{u,v}$ is the channel gain, $I_{u,v}$ is the inference gain and p_u, as well as σ, is the transmission power and noise power respectively. The propagation delay depends upon the propagation delay of various links and is represented as $T^{\theta}_{prop} = \Sigma_{u,v \epsilon L} \delta_{u,v} x^{\theta}_{u,v}$, where $\delta_{u,v\epsilon L}$ is the propagation delay of the link and $x^{\theta}_{u,v}$ represents the link selected for offloading the data θ. The queuing delay involved in offloading the data depends upon the arrival α^{θ} and departure rate λ^{θ} of the data θ and is represented as $T^{\theta}_{queue} = 1/(\lambda^{\theta} - \alpha^{\theta})$. The processing time depends upon the time taken to categorize data into groups, select the appropriate blockchain, and mine the data block. We calculate the processing time is as $T^{\theta}_{pt} = (t^{\theta}_{cluster}/\gamma) + (t^{\theta}_{select\_con}/\omega) + (t^{\theta}_{mine}/\eta)$, where $t_{cluster}$ is the time taken in categorizing the data θ in the appropriate cluster, γ is the processing capacity of the node performing clustering, t_{select_con} is the time taken to select the appropriate consensus algorithm, ω is the processing capacity of the blockchain system, t_{mine} is the time to add the block to the blockchain and η is the processing power of the blockchain used. We calculate the delay for uploading the data as:

$$T = T^{\theta}_{trans} + T^{\theta}_{prop} + T^{\theta}_{queue} + T^{\theta}_{pt} \tag{1}$$

C. Energy Model

We consider that the blockchain requires a computation task at a time to add blocks. To calculate the energy utilized by the blockchain system to add blocks into the blockchain, we use the energy consumption model described in $\epsilon = \kappa \times \zeta^2$, where κ is the energy coefficient depending on the chip used in the system, and f is the frequency of the processor used in the system. The energy utilized depends on the workload (w_{θ}) to add the data θ into the blockchain. We calculate the workload to add a block using the CPU usage required for the same. Further, we calculate the overall energy consumption to add blocks into the blockchain as:

$$E = \kappa \times \zeta^2 \times w_{\theta} \tag{2}$$

D. Preliminaries

We give a brief introduction to clustering and consensus algorithms in this section.

1) Clustering: Clustering determines the intrinsic grouping present in inconsistent data by partitioning the data sensed by various sensors based on its attributes. We use agglomerative clustering, which takes input from various sensors such as pressure sensors, temperature sensors, humidity sensors, and photodiodes and categorizes the data based on sampling rates, variability, and the value range. Initially, each data element is separate, and this algorithm calculates the dissimilarity between each data element using Euclidean Distance:

$$|a - b|_2 = \sqrt{\Sigma_i (a_i - b_i)} \tag{3}$$

The weights are various parameters such as sampling rates(SR), variability(v), and the value ranges(vr). The formula for Euclidean distance according to our data is given as:

$$|a-b|_2 = \sqrt{\Sigma_i ((v_{ai} - v_{ib}) + (vr_{ai} - vr_{ib}) + (SR_{ai} - SR_{ib}))} \tag{4}$$

Further, we calculate the centroid C of the new cluster formed by joining clusters having centroids C_s and C_t using $C = |C_s - C_t|$. These become new data points, and this method further merges the clusters considering these data points.

2) Blockchain: The blockchain is the distributed database consisting of blocks of all the transactions which builds trust through five attributes: 1. Distributed, 2. Secure, 3. Transparent, 4. Consensus, and 5. Flexible. Blockchain achieves reliability because of the consensus algorithm as all the miners reach a common agreement using various consensus algorithms such as PoW, PoB, PoS, PBFT. As proof of concept, we use three consensus algorithm to implement our blockchain.

- ***Proof of Work:*** This consensus algorithm utilizes mathematical puzzles to reach a standard agreement. The miner who solves the puzzle with less value than the hash of the block mines the next block. The computation power depends upon the number of nodes present in the environment. Since all miners solve a complex problem to reach an agreement, the energy consumption in PoW is very high. For instance, the miner who can produce a number (P) whose sum with previous proof (PPoW) is divisible by seven can only add a block to the blockchain. The value of the new proof is one more than the last proof as represented in the equation:

$$P = PPoW + 1 \tag{5}$$

- ***Proof of Stake:*** The miners validate the block by betting on it and get a reward proportionate to their bets and, accordingly, increase their stakes. The miner with a high economic stake adds a block to the blockchain.
- ***Practical Byzantine Fault Tolerance:*** The system reaches an agreement even if some nodes in the system are faulty. The system reaches an agreement if more than one-third of the miners are honest. This algorithm requires R number of nodes to handle f faulty nodes to reach an agreement and is represented by the equation:

$$|R| = 3 \times f + 1 \tag{6}$$

- ***Blockchain Models:*** There are two types of system models for blockchain: 1) Permissioned, 2) Permissionless. In permissioned blockchain, prior permission is required by the miners to participate in the consensus process. Unlike permissioned blockchain, permissionless blockchain does not require prior authorization to participate in the consensus process. We propose a system that is semi-permissioned. As proof of concept, we use PoW, PoS, and PBFT. The blockchains using PoW and PoS are permissionless, whereas PBFT is permissioned.
- ***Attack Models:*** Blockchains are secure ledgers; various attacks are possible on blockchains such as Sybil Attack, Byzantine, and Crash Attack. The possibility of attacks on the blockchains depends upon the consensus algorithms used. In Sybil Attack, multiple copies of a system are created to gain dis-appropriate influence in the miners' network. The possibility of a Sybil attack on the blockchain using PoW and PoS is almost negligible, whereas PBFT is susceptible to Sybil attacks. Further, in Byzantine Attack, the malicious nodes have control over the authenticated nodes, whereas, in crash attacks, the network is flooded with many requests, and the server cannot fulfill the demands. The possibility of a crash and

byzantine attack on the blockchain using PoW is 50%, PoS is 50%, and PBFT is 33%.

E. Proposed Solution

We consider an industry deployed with many sensors that sense data produced by various IIoT devices present in the industry. The data produced by these sensors is heterogeneous and is challenging to manage. To manage this, firstly, we use agglomerative clustering to group similar data into clusters. It is easier to collect comparable data as compared to the data which is not identical and inconsistent. Further, the clustering layer forwards the data to the blockchain system, where the blockchains with dynamic consensus algorithms store the data. As proof of concept, we use three blockchains using PoW, PoS, and PBFT, respectively. The delay involved in offloading the data is given by equation 1 whereas the energy required to add data to the blockchain is given by equation 2. The cluster which requires an ample amount of security uses blockchain with PoW to store its data. At the same time, the transactions that do not need confirmations to use blockchain with PBFT and the data require a large amount of energy use blockchain with PoS as it is an energy-efficient consensus algorithm. The blockchain system decides the appropriate type of consensus algorithm for the clustered data. The job tracker fetches the data stored in the blockchain to track the jobs in industries.

Algorithm 1: Algorithm for selecting blockchain with dynamic consensus algorithm.

```
Input: Heterogeneous data sensed by sensors
Output: Jobs status
Procedure:
while sensors $s_i \epsilon S$ senses data do
    Perform agglomerative clustering;
    if cluster of data is Type 1 then
        Store data in blockchain 1 with PBFT;
    end
    if cluster of data is Type 2 then
        Store data in blockchain 2 with PoS;
    end
    if cluster of data is Type 3 then
        Store data in blockchain 3 with PoW;
    end
    // chooses appropriate blockchain based on
       attributes of data
end
while blockchain $b_i \epsilon B$ do
    while block is not Null do
        fetch each data element and track
    end
end
```

1) Selection of Consensus Algorithms: The middleware, after performing clustering, selects the appropriate consensus algorithms based on the attributes of data. Since PoW provides ultimate security, PoS is energy-efficient, and PBFT is fast. Therefore, the data that require extreme security use PoW, the data that need more energy use PoS, whereas the data require faster transactions, i.e., the data with a high sampling rate use PBFT. For instance, cluster 1 and cluster 2 contain the data of photodiodes and humidity sensors, respectively. Whereas cluster 3 contains the data of temperature, pressure, and exhaust vacuum sensors as their value ranges and variability are relative. Further, cluster 1 applies PBFT as the sampling rates of these sensors are very high, and waiting for confirmations for adding data for these sensors leads to data loss. Cluster 2 applies PoS, as the variability of photodiodes and humidity sensors is very high, and it requires a high amount of energy to handle these data. To provide an energy-efficient solution to handle these data, it is necessary to use energy-efficient consensus such as PoS. Further, cluster 3 uses PoW as even a single-digit corruption leads to a significant change in the data, which results in various hazards as shown in Algorithm 1. To prevent this, it is necessary to utilize the consensus algorithms (PoW), which provides ultimate security.

2) Complexity: The time complexity of agglomerative clustering is $O(n^2 logn)$ where n is the number of data elements. Further, we apply the consensus algorithms, which depend upon the number of nodes present in the network. The complexity of PoW relies on the number of nodes, whereas in PoS, there are no miners. The computational power in PoS depends upon the stakes present in the wallet of the validator. Further, the complexity of PBFT relies on the number of messages exchanged (m) as well as the number of nodes (k) and is represented as $O(m^k)$.

Proposition 1. *Irrespective of energy-constrained devices, A-Blocks is an optimal solution for heterogeneous data.*

Proof. Typically, industries deal with a heterogeneous set of data coming from a myriad range of sensors $S = \{s_1, s_2, s_3, \ldots, s_n\}$. Handling these data using a single consensus algorithm is challenging due to the varying attributes and applications. A single consensus algorithm is not suitable for overcoming the challenges pertaining to the heterogeneous incoming data. A-Blocks dynamically selects the consensus algorithm and distributes the load and data over various nodes $N_o = \{n_1, n_8, n_15, \ldots, n_{n-3}\}$. Further, A-Blocks according to the need of data utilize the consensus algorithm, which reduces the energy requirement and make A-Blocks suitable for energy-constrained devices. □

IV. Performance Evaluation

A. Experiment Setup

In this work, we evaluate the proposed A-Blocks system using a set of resource-constrained Raspberry Pi devices. We set up the network using wireless communications (Wi-Fi) and consider a client-server-based system. The client system sends the sensors' data to the middleware, which clusters the data and selects the suitable consensus algorithm for the data based on its attributes. Further, the middleware sends the data to the appropriate blockchain server which stores the data according to its properties. We use Python 3 for executing our routines. We use the combined cycle power plant data set [21] for performing our experiment. We consider various sensors such as temperature sensors, pressure sensors, humidity sensors,

Table II: Various types of data.

Data type	Type 1	Type 2	Type 3	Type 4
Description	Data with high sampling rates	Data with high variability	Data with average variability and sampling rates	Random data generated by the sensors

Table III: Comparison of A-Blocks with existing works.

Scheme	Centralized/ Decentralized	Consensus	No. of Miners	Block Size
Yao *et al.* [22]	No	Static	No	Dynamic
Hei *et al.* [19]	Yes	Static	No	Dynamic
Cui *et al.* [23]	Yes	Static	Yes	Dynamic
A-Blocks (proposed)	Yes	Dynamic	Yes	Dynamic

exhaust vacuum sensors, and photodiodes to sense data. We recognize three attributes: 1. value ranges of data produced, 2. variability, and 3. sampling rates to categorize the data into various groups. We use three blockchains with three different consensus algorithms to store three different types of data as described in Table II. Type 1 data is the data of photodiodes and belongs to cluster 1. The sampling rate of Type 1 data is very high. Similarly, Type 2 data belongs to cluster 2 with high variability and is the humidity sensor data. The Type 3 data we obtain from temperature sensors, pressure sensors, and exhaust vacuum sensors are categorized into cluster 3.

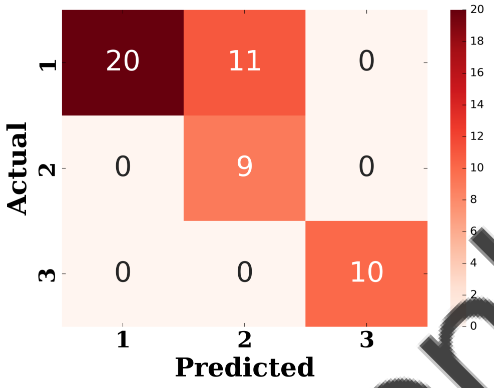
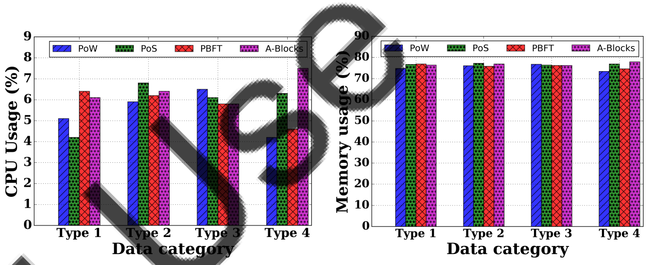


Figure 2: Confusion matrix of clusters formed for three different types of data.



(a) CPU.

(b) Memory (RAM).

Figure 3: Resource utilization by the devices.

B. Results

1) Comparison with existing schemes: The A-Blocks system is a dynamic consensus algorithm that selects the blockchain according to the traits of incoming data. To achieve this, it takes into account the varying block size, number of miners, data types, sampling rates, and variability of data. As shown in Table III, the current solutions in the literature do not focus on the data variability and A-Blocks handles it with increased efficiency by dynamically selecting the appropriate consensus algorithms. For instance, the methods such as in [19], [22] are static, centralized, and do not consider the varying number of miners which restricts the proper management of data. On the other hand, A-Blocks overcomes the aforementioned restrictions as it is decentralized, dynamic, and accounts for the varying number of miners as well as blocksize.

2) Clustering and data management: A-Blocks categorizes data into three types as mentioned in Table II by using agglomerative clustering and Fig. 2 represents the confusion matrix of the same. It categorizes the data with an overall accuracy of 78%. In particular, we observe that A-Blocks correctly identifies Type 2 and Type 3 data. However, in the case of Type 1 data, A-Blocks identifies it as Type 2. Intuitively, this is because the value ranges of Type 1 and Type 2 data are similar. However, due to the similarity in the value ranges of Type 1 and Type 2, the decision on the clustering mechanism does not have adverse impact. This is because, we apply PBFT for Type 1 as its sampling rate is high and PoS for Type 2, which also supports data with high sampling rate. We comment that the current classifier successfully identifies the data category, particularly for Type 2 and Type 3. However, some tuning may improve the developed model for improving the identification of Type 1 data.

3) CPU Usage: We record the CPU consumption while executing A-Blocks and compare them with solutions that use any one of PoW, PoS, and PBFT as the possible mining techniques and perform 40 iterations for each experiment. We observe that for Type 4 data (random data which is more likely to be generated in industries), A-Blocks shows a minimum increases in the CPU utilization by 5% (average), compared to conventional methods (Fig. 3a). We attribute this increase to the routines responsible for clustering and then the consensus algorithm. We also observe that Type 1 and Type 2 data blockchains with PBFT and PoS require more CPU because of their high sampling rates and high variability. In particular, the CPU utilization for Type 1 data with PBFT is 6.4%, and Type 2 data with PoS is 6.8%. On average, the CPU utilization of A-Blocks for Type 4 data is 8%. We infer from our observation that the choice of an appropriate consensus algorithm for a particular type of data is crucial towards CPU utilization. Further, Raspberry pi devices have a 1.2 GHz CPU with 4 cores and A-Blocks demonstrates a maximum requirement of only 8%, which is nominal. We comment that A-Blocks is suitable for resource-constrained devices.

4) Memory Usage: We record the memory consumption used by A-Blocks to execute its routine and observe that it utilizes approximately 80% of the memory as shown in Fig. 3b. On the other hand, Pow, PoS, and PBFT utilize memory only 74.5%, 79%, and 77.3%, respectively. This behavior is because the A-Blocks execution routine uses three different blockchains in a

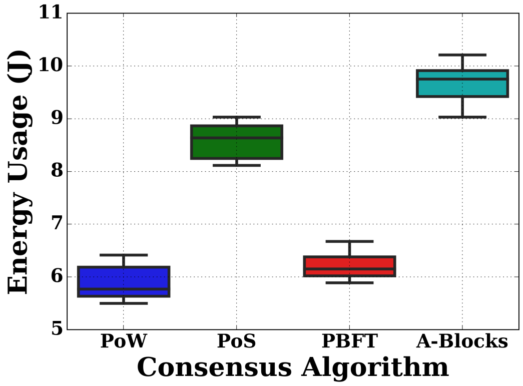


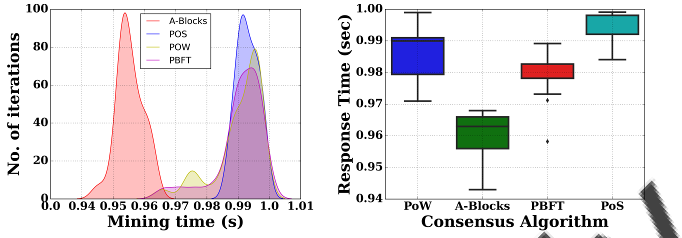
Figure 4: Energy consumption in the blockchain.

single system, which requires extra memory. The clustering technique also needs some memory for its execution which improves memory utilization. We entail from our observation that the selection of an appropriate consensus algorithm for a particular type of data is crucial to utilize memory more efficiently. On average, A-Blocks requires 3% more memory for Type 4 data compared to conventional methods. Raspberry pi devices have 1 GB of RAM and 3% is a diminutive increase, which is acceptable. In continuation to our observations in Section IV-B3, we comment that A-Blocks is suitable for resource-constrained devices.

5) Energy Consumption: We explore the energy for generating the blocks and adding the blocks with the A-Blocks system and we calculate it according to Equation 2. We use the CPU consumption from the observations in Section IV-B3. As expected, we observe in Fig. 4 that on average, A-Blocks requires $10J$ to add data blocks in the appropriate blockchain, which is $2J$ (average) more as compared to other schemes. We attribute this increase to the same comments as in Section IV-B3. We imply that the energy consumption increases as the number of computation tasks increases.

6) Mining Time: Mining time is necessary for generating and adding blocks into the blockchains. The mining time is directly proportional to the number of miners present in the network, which validates the block. We observe from Fig. 5a that the mining time for A-Blocks is $0.97s$, which is less as compared to other schemes. This is because, we utilize different blockchains for varying data which divides the load of achieving consensus on various miners' networks. The mining time of PoW, PoS, and PBFT are $0.995s$, $0.998s$, and $0.989s$, respectively. We imply that the selection of an appropriate consensus algorithm for a particular type of data reduces the mining time by 7% when applied to Type 4 data. We imply that the grouping of data results in the reduction of mining time which further increases the scalability of the system and supports the data with high sampling rate.

7) Response Time: We explore the response time of accessing data from the blockchain using the A-Blocks method. From Fig. 5b, we observe that the response time of accessing the data from a single blockchain system is higher as compared to that of A-Blocks. This is because A-Blocks groups the data into various categories and stores it into the respective blockchain using the appropriate consensus algorithm. This increases the effectiveness of the data management and its processing. Since parsing a single large blockchain is time consuming, A-Blocks facilitates data-centric dedicated blockchain. Such a blockchain is shorter in length, which further reduces the response time. We comment that with 30% reduction in response time under the A-Blocks system, it enhances the suitability of using blockchains in real-time scenarios.



(a) Mining time.

(b) Response time.

Figure 5: Time for different consensus algorithms.

In summary, compared to conventional blockchain practices, A-Blocks significantly reduces the mining and response time, which makes it suitable for real-time applications with negligible CPU and memory overheads. From our observation, we comment that it is best suitable for scenarios that satisfy the following conditions:

- C1: Resource-constrained devices: From Section IV-B-3 and IV-B-4, we observed that A-Blocks requires negligible CPU and memory overheads compared to conventional blockchain deployments. This makes it suitable for use in any legacy infrastructure, meaning that no additional setup is necessary.
- C2: Optimal mining time: From Section IV-B-6, we observed that A-Blocks significantly reduces the mining time compared to the conventional blockchain deployments which makes it suitable for the applications producing data with high sampling rate.
- C3: Real-time applications: From Section IV-B-7, we observed that A-Blocks reduces the response time as compared to the convention blockchain which makes it suitable for the real-time applications.

In summary, A-Blocks is suitable for applications producing data at a very high rate and utilizing variant types of sensors that produce heterogeneous data.

V. Conclusion

In this paper, we proposed a dynamic consensus-based blockchain (A-Blocks) to manage the data produced by the sensors deployed in industries. A-Blocks consists of two phases: 1) categorizing the data into groups based on their traits and 2) selection of appropriate consensus method for handling the same. In the first phase, we assorted data into various categories using agglomerative clustering. We used the value ranges, variability, and sampling rates of the data to categorize them into groups. In the second phase, we implemented a dynamic blockchain system using three consensus algorithms as proof of concept (PoW, PoS, and PBFT) to store the data generated in phase 1. We selected one of the chains to store data based on its traits and implemented the respective consensus algorithm. Finally, we proposed an algorithm to choose the consensus algorithm in the blockchain system. We presented extensive experimental results to show the efficiency of the proposed scheme. In this work, we abstained from assimilating the pricing policies and managing heterogeneous blockchains, which we plan to address in our extended work.

References

[1] R. K. Kumar and R. Chadrasekaran, "Attribute Correction - Data Cleaning Using Association Rule and Clustering Methods," *International Journal of Data Mining & Knowledge Management Process*, vol. 1, pp. 22–32, 2011.

[2] H. Jia and Y.-M. Cheung, "Subspace Clustering Of Categorical And Numerical Data With An Unknown Number Of Clusters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3308–3325, 2017.

[3] J. Liang, J. Yang, M.-M. Cheng, P. L. Rosin, and L. Wang, "Simultaneous Subspace Clustering And Cluster Number Estimating Based On Triplet Relationship," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3973–3985, 2019.

[4] R. Pass and E. Shi, "Fruitchains: A Fair Blockchain," in *Proceedings of the ACM Principles of Distributed Computing*, 2017, pp. 315–324.

[5] L. Bai, M. Hu, M. Liu, and J. Wang, "BPIIoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT," *IEEE Access*, vol. 7, pp. 58 381–58 393, 2019.

[6] G. Wang, Z. Shi, M. Nixon, and S. Han, "Chainsplitter: Towards Blockchain-Based Industrial IoT Architecture for Supporting Hierarchical Storage," in *Proceedings of the IEEE International Conference on Blockchain*. IEEE, 2019, pp. 166–175.

[7] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, and Laventman, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the* 13^{th} *EuroSys conference*, 2018, pp. 1–15.

[8] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2019.

[9] X. Xu, Q. Lu, Y. Liu, L. Zhu, H. Yao, and A. V. Vasilakos, "Designing Blockchain-Based Applications a Case Study for Imported Product Traceability," *Future Generation Computer Systems*, vol. 92, pp. 399–406, 2019.

[10] E. N. Lallas, A. Xenakis, and G. Stamoulis, "A Generic Framework For A Peer To Peer Blockchain Based Fog Architecture In Industrial Automation," in *Proceedings of the* 4^{th} *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE, 2019, pp. 1–5.

[11] A. Dua, A. Dutta, N. Zaman, and N. Kumar, "Blockchain-Based E-Waste Management In 5G Smart Communities," in *proceedings of the International Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 195–200.

[12] P. Fraga-Lamas and T. M. Fernández-Caramés, "A Review On Blockchain Technologies For An Advanced And Cyber-Resilient Automotive Industry," *IEEE Access*, vol. 7, pp. 17 578–17 598, 2019.

[13] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[14] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain At The Edge: Performance Of Resource-Constrained IoT Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174–183, 2020.

[15] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Smchain: A Scalable Blockchain Protocol for Secure Metering Systems in Distributed Industrial Plants," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, 2019, pp. 249–254.

[16] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "BSeIn: A Blockchain-based Secure Mutual Authentication with Fine-Grained Access Control System for Industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018.

[17] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment," *IEEE Transactions on Industrial Informatics*, 2019.

[18] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing Blockchain-Based Access Control Protocol in IoT-Enabled Smart-Grid System," *IEEE Internet of Things Journal*, 2020.

[19] S. He, W. Ren, T. Zhu, and K.-K. R. Choo, "BoSMoS: A Blockchain-Based Status Monitoring System for Defending Against Unauthorized Software Updating in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 948–959, 2019.

[20] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency Performance Modeling and Analysis for Hyperledger Fabric Blockchain Network," *Information Processing & Management*, vol. 58, no. 1, p. 102436, 2021.

[21] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[22] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource Trading in Blockchain-Based Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[23] L. Cui, S. Yang, Z. Chen, Y. Pan, M. Xu, and K. Xu, "An Efficient and Compacted DAG-Based Blockchain Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4134–4145, 2019.