Proof-of-Work Consensus Approach in Blockchain Technology for Cloud and Fog Computing Using Maximization-Factorization Statistics

Gulshan Kumar, *Member, IEEE*, Rahul Saha, *Member, IEEE*, Mritunjay Kumar Rai, Reji Thomas, and Tai-Hoon Kim, *Member, IEEE*

***Abstract*—In this paper, we discussed an efficient statistical method with proof-of-work consensus approach for cloud and fog computing. With this method, solution with precise probability in minimal time is realized. We have used the expectation maximization algorithm and polynomial matrix factorization. The advantages of this statistical method are the less iteration to converge to the consensus solution and easiness to configure the complete mathematical model as per the requirement. Moreover, the energy and memory consumption are also less which make this approach appealing for cloud and fog computing. The experimental results also show that the proposed approach is significantly efficient in terms of time and memory consumption. This novel approach seems beneficial for Internet-of-Things (IoT), one of the most fast-growing technologies in network computing.**

***Index Terms*—Blockchain technology, cloud, expectation maximization (EM), fog, Internet-of-Things (IoT), matrix factorization, network computing, proof-of-work (PoW).**

I. Introduction

THE CURRENT trend of network computing technology is evolving around Internet-of-Things (IoT), a subgroup of cloud, fog, and edge computing. The scalability and security issues of IoT, viz., deficiency in addressing edge computing and vulnerable or compromised IoT-fog devices, are the main concerns to overcome. In this respect, the emerging blockchain and smart contracts technologies [1], [2] have brought in new features for security [3]. Among both, the distributed approach, consensus mechanism, and transparent transactions of blockchain have made it popular with IoT environment [4] and also for cloud, fog, and edge computing environments [5]–[7]. These benefits made it reliable technique for cryptocurrency and mining as well. However, various approaches are identified in recent years where blockchain and its consensus mechanism for IoT and cloud-fog computing are gaining wider acceptance. For example, blockchain has been applied in some recent work of cloud, fog, and edge computing [8]–[13] to provide new security features to networking technology [14]–[16]. Consensus is the core of blockchain to provide distributed security service. Different types of blockchain are introduced in recent year for various applications. But due to the problem of large resource requirements, proof-of-work (PoW) sets back in wide acceptability. Moreover, clouds and their allied applications majorly concern with data access control.

In fact, *cryptocurrency* mining is the by-product of blockchain technology for *Bitcoin* in 2008. It is a process by which users get virtual coins in exchange for validating the transactions and adding new blocks in a blockchain network. PoW consensus mechanism is the most favored process for mining in blockchain networks to achieve the decentralized attributes as it is one of the prime requirements in present network technologies to avoid centralized failures. PoW is a protocol for solving a mathematical puzzle and achieving a guaranteed consensus required to define an expensive computer calculation, also called mining, that needs to be performed to create a new group of trust less and decentralized transactions on a distributed ledger called blockchain. Mining is executed for two reasons: 1) to verify the validity of a transaction and avoiding the double-spending [22] which is the risk that a digital currency can be spent twice by copying the original digital token and sending it to the receiver and 2) to create new digital currencies by rewarding miners for performing the previous task. With time, competition, and technological urge have increased manifold and recently, miners are required to spend more computing power to generate the same number of Bitcoins. This resulted in using a specific hardware, application-specific integrated circuit (ASIC) [17].

A series of tasks are performed when you want to set a transaction in a blockchain [5]. First, the transactions are bundled together into a block. Second, miners verify that transactions within each block are legitimate using a mathematical puzzle known as PoW [18]. Third, reward is given to the miner who solves the puzzle first. At last, all the transactions are verified by the consensus [19]–[21] and are stored in the public blockchain. The "mathematical puzzle" in PoW exhibits an asymmetry feature. All the network miners in a blockchain network compete to be the first for searching a solution for the mathematical problem with a predefined difficulty. It is a problem that is not solvable in other ways and so, brute force

Manuscript received December 28, 2018; revised February 13, 2019 and April 5, 2019; accepted April 15, 2019. Date of publication April 18, 2019; date of current version July 31, 2019. *(Corresponding author: Rahul Saha.)*

G. Kumar, R. Saha, and R. Thomas are with the Division of Research and Development, Lovely Professional University, Phagwara 144 411, India (e-mail: gulshan3971@gmail.com; rsahaaot@gmail.com).

M. K. Rai is with the School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara 144 411, India.

T.-H. Kim is with the Department of Computing and IT, University of Tasmania, Hobart, TAS 7005, Australia.

Digital Object Identifier 10.1109/JIOT.2019.2911969

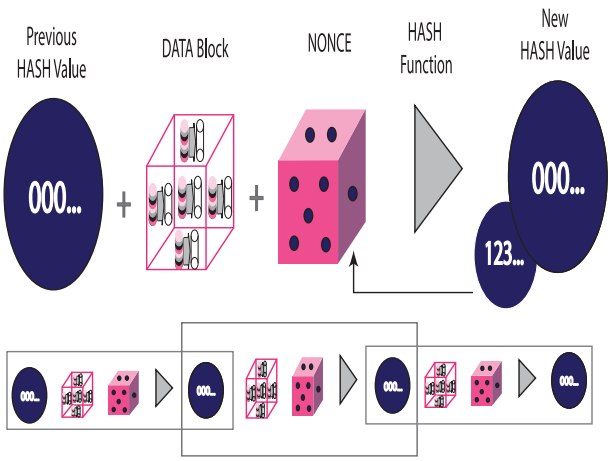

Fig. 1. Schematic of PoW functioning.

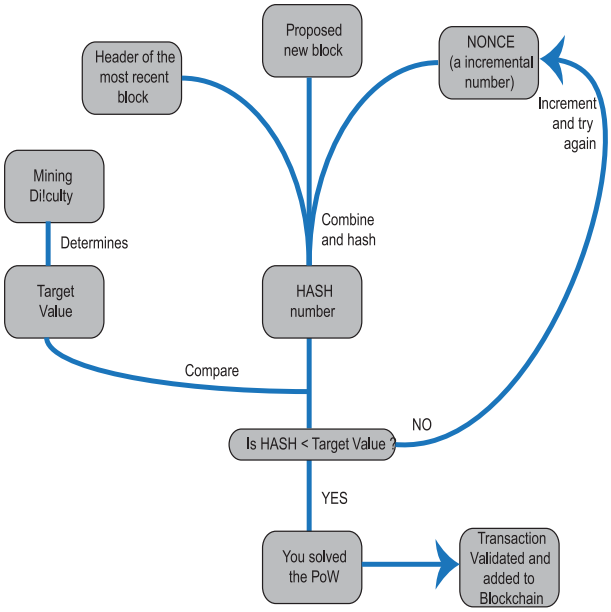

Fig. 2. Expanded process of PoW functioning with difficulty level.

essentially is required with a huge number of attempts. When a miner finally finds the right solution, the miner announces it to the whole network at the same time, receiving a cryptocurrency prize (the reward) provided by the protocol. From the technical point of view, mining process is an operation of inverse hashing: it determines a number (nonce), so the cryptographic hash algorithm of block data results in less than a given threshold. This is termed as difficulty value of mining, which makes competition among miners more significant. This method also increases the cost of the block creation, pushing miners to improve the efficiency of their mining systems to maintain a positive economic balance. Besides the digital currencies like Bitcoin and Ethereum, PoW has become a de facto standard for different blockchain applications. The basic functioning of PoW has been shown in Fig. 1 [23]. The figure explains the process of PoW, where it generates a new hash value depending upon the constraints of previous hash value, data block, and nonce. Fig. 1 is expanded to Fig. 2 [24] where the difficult level has been shown to depend on the parameters such as timing threshold (t) and also a target value where the value of the expected hashed output must contain a predefined number of 0 s at the start or end or in a fixed position.

A. Related Work

Different consensus approaches and applications have been researched in recent but significant approach for learning PoW based consensus has been found to be a missing link. In this section, we have explained some of the recent works in this dimension. Mainstream blockchain and modified Byzantine consensus algorithms have been are discussed recently [25]. A consensus solution for faulty and Byzantine nodes is shown in [26]. This paper uses a multiscale filtering algorithm based on local information. It considers both local and global bounds of threats. This resilient approach faces the problem of communication delay and external perturbations. A stochastic-based consensus condition also has been evaluated for multiagent systems and that considered both delays and noises [27]. Martingale convergence used strengthens the stochastic weak consensus by realizing exponential convergence. Further, it also considered degenerate Lyapunov function to find the sufficient consensus conditions. Another multiagent-based consensus approach with continuous and discrete time subsystems also recently tried [28]. This method proposes a switched filtering strategy for cooperative nodes based upon available local information. Such an approach guaranteed of resilient consensus. However, the problems for this consensus and resilient scaled formation generation are solved using network robustness and weighted-mean-subsequence-reduced (W-MSR) procedure. Asymptotic model for linear systems under a class of switching communication graphs is a recently studied consensus [29]. A bounded uniform rate of convergence to consensus is also established for a class of continuous time multiagent system. The multiagent systems-based global optimal consensus problem has been solved with this bounded control protocols over fixed and directed communication network [30]. Other multiagent-based solutions with guaranteed cost consensus and directed topology are considered in [31]. The distributed control protocols described here uses absolute and relative velocity damping. However, consensus is achieved using Lyapunov stability theory, some sufficient conditions are also given to achieve the so-called consensus.

Some of the consensus models included the collective behavior in an environment without a central control and that motivated works in spin systems and discrete opinion space [32], [33]. Among them, DeGroot model has been gained more takes using the Markov chain for probability distribution [34], [35]. A resilient approach of consensus against noncooperative behavior is shown in [36]. It basically deals with hybrid network of non rational agents. Expectation of convergence is estimated with a filtering strategy. Sure convergence is a problem in this algorithm. However, a consensus problem for multiple delta operator systems got more interest for describing continuous time processes at rapid rate sampling in distributed approach [37]. Two distributed consensus

protocols are constructed for both leaderless environment and leader-follower environment. Another delta operator-based consensus has been shown in [38]. From the afore-mentioned recent works, it can be seen that a significant research has been done on the multiagent consensus for various applications. But, statistical method of consensus expectation has been researched significantly less. Therefore, we propose a statistical method for solving the mathematical puzzle in PoW to get a guaranteed consensus. Apart from that, IoT are one of the most fast-growing technologies in recent years for network computing scheme also considered. As it is well known, cloud, fog, and edge computing, have become integral part of many IoT applications. The decentralized IoT demand the inclusion of blockchain technology in cloud-fog-edge environment and that provide transparent platform of transactions with consensus. These features have made blockchain widely accepted in various applications, such as access of electronic medical record (EMR) in healthcare, security solutions, research records, and many more. However, IoT devices are resource constrained and dynamic in nature and hence an efficient blockchain approach is the need of the hour. IoT should efficiently provide an expected solution so that the condition constrained devices can participate in consensus to validate the transactions for data access. Though there are various consensus processes exist, we have considered PoW in this paper. The problem of PoW is that it is more resource consumable and minimization of the time is the key point.

The main contribution of the presented work has been summarized below.

1) We have proposed a statistical method to solve the expectation value of a mathematical puzzle in PoW.
2) The method is basically depending upon a mathematical model of expectations. We have also used polynomial matrix factorization for the ease of obtaining the solution for the PoW considering all the constraints of the mathematical puzzle process.
3) The proposed approach is less complex, easy to model and configure, and less time and memory consuming. Besides, computational power is also less.
4) This primitive expectation model is applicable for all the consensus algorithms, including distributed multiagents, stochastic, continuous, and discrete time variability.

Being discussed the motivation of doing the present work, remaining paper has been organized as follows. We considered some consensus methods so that we develop a generalized statistical method for expectation maximization (EM) of consensus and is discussed in Section II. As can be seen Section III discusses the results and the conclusion from the present has been outlined in Section IV.

II. Proposed Work

We have proposed a solution framework with statistical method. The proposed method help to calculate the expected consensus of PoW to maximize by gather the data from the PoW processes and then calculating the expectation for the new iteration. The difficulty level therefore cannot be problem for the solution. With the initial starting, the resource and time consumption are higher as the system is to be gathering data from the environment of PoW containments and gradually it takes less resource and time for further execution scenarios. We have considered the following assumptions.

1) The difficulty level predefined is only about the number of zeros at the starting of the hash value.
2) The nonce and all other data are experimented with variation from 256 to 2048 bits.
3) SHA-512 is used for the hashing operation.

A. Statistical Approach

To execute the learning method, miners need to perform EM algorithm [39]. This algorithm is a useful iterative procedure to compute the maximum likelihood (ML) estimate in the presence of missing or hidden data. Let $\mathcal{R}$ is the set of random nonce which results from a parameterized family, $\mathcal{D}$ is the set of difficulty level of the mathematical puzzle expressed in terms of number of 0 s at the start of the hash output, $\mathcal{P}$ is the set of all previous blocks in the blockchain, and t is the given time limit for the solution. Following these parameters, we wish to find out hashed output $\mathcal{H}$ such that $P(\mathcal{H}|r.\acute{d}.p.t)$ is maximum where $r \in \mathcal{R}$, $\acute{d} \in \mathcal{D}$, and $p \in \mathcal{P}$. In order to calculate the $\mathcal{H}$ giving the other conditional variables we have used log (base 2) likelihood function as

$$\mathcal{L}(\mathcal{H}) = \ln P\big(\mathcal{H}|r.\acute{d}.p.t\big). \tag{1}$$

This is an iterative process to maximize the $\mathcal{L}(\mathcal{H})$. Assuming the nth iteration value for $\mathcal{H}$ is $\mathcal{H}_n$ so that the following equation holds:

$$\mathcal{L}(\mathcal{H}) - \mathcal{L}(\mathcal{H}_n) = \ln P\big(\mathcal{H}|r.\acute{d}.p.t\big) - \ln P\big(\mathcal{H}_n|r.\acute{d}.p.t\big). \tag{2}$$

Notice that in the above conditional probability of maximization, the above equation can be rewritten as

$$\mathcal{L}(\mathcal{H}) - \mathcal{L}(\mathcal{H}_n) = \ln \sum P(\mathcal{H}|z)P(z|\mathcal{H}) - \ln P\big(\mathcal{H}_n|r.\acute{d}.p.t\big)$$

where

$$z = P\big(\mathcal{H}|\acute{d}\big)P\big(\mathcal{H}|p\big)P(\mathcal{H}|t). \tag{3}$$

We have used further the Jensen's inequality [40] where f be a convex function defined on an interval of I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i). \tag{4}$$

Following (4), we can reconsider (3) as shown in (5) at the bottom of the next page.

Further, we can write

$$\mathcal{L}(\mathcal{H}) \geq \mathcal{L}(\mathcal{H}_n) + \Delta(\mathcal{H}|\mathcal{H}_n). \tag{6}$$

We can also rewrite (6) for convenience as

$$l(\mathcal{H}|\mathcal{H}_n)\Delta\mathcal{L}(\mathcal{H}_n) + \Delta(\mathcal{H}|\mathcal{H}_n). \tag{7}$$

Thus, the relationship becomes as: $\mathcal{L}(\mathcal{H}) \geq l(\mathcal{H}|\mathcal{H}_n)$ where the function $l(\mathcal{H}|\mathcal{H}_n)$ upper bounded by the likelihood function $\mathcal{L}(\mathcal{H})$. The objective of using EM algorithm is to obtain a value of $\mathcal{H}$ such that $\mathcal{L}(\mathcal{H})$ is maximized. As $l(\mathcal{H}|\mathcal{H}_n)$ upper bounded by the likelihood function $\mathcal{L}(\mathcal{H})$ and that the value

of the functions $l(\mathcal{H}|\mathcal{H}_n)$ and $\mathcal{L}(\mathcal{H})$ are equal at the current estimator for $\mathcal{H} = \mathcal{H}_n$. Therefore, any $\mathcal{H}$ which increases $\mathrm{l}l(\mathcal{H}|\mathcal{H}_n)$ also increases the $\mathcal{L}(\mathcal{H})$. In order to achieve the greatest possible increase in the value of $\mathcal{L}(\mathcal{H})$, the EM algorithm calls for iterative process and we denote the updated value as $\mathcal{H}_{n+1}$. In (8) and (9) $\mathrm{argmax}_{\mathcal{H}}$ denotes the maximized value of the objective function with respect to $\mathcal{H}$

$$\mathcal{H}_{n+1} = \mathrm{argmax}_{\mathcal{H}}\{l(\mathcal{H}|\mathcal{H}_n)\} \tag{8}$$

$$\mathrm{argmax}_{\mathcal{H}}\left\{E_{r.d.p.t|\mathcal{H},\mathcal{H}_n}\{\ln\ P(\mathcal{H}_n|r.d.p.t)\}\right\}. \tag{9}$$

With this solution of expectation–maximization, the proposed system gathers the data from the PoW solutions in first few iterations and then applies vector auto regressive moving average (VARMA) model [44].

B. Linear Model of Expectation

In the previous stage, we have calculated the expectation value for the required hash considering all the constraints of $r.d.p.t$. To continue further, we have modeled a linear rational expectation model [41]–[43] as

$$B_0 y_t + B_1 y_{t-1} + \cdots + B_p y_{t-p} + A_0 \widehat{y_t} + \cdots + A_p \widehat{(y_{t+p})} = \tau x_t + u_t \tag{10}$$

where, the matrices A_i and B_i are $n \times n$, y_t is an n-dimensional vector of endogenous variables (previous blocks p, difficulty level d and hasing output $\mathcal{H}$) at time t, x_t is m-dimensional vector for exogenous variable (random nonce r) at time t, τ is a $m \times k$ dimensional matrix of parameters, and u_t is an n-dimensional vector of disturbances which are not serially independent. The cap over the variables indicate its conditional expectation based on information in the duration of $t-1$. The term u_t is to be following a stationary stochastic process which is represented in linear form as:

$$u_t = \Delta L_{v_t} \tag{11}$$

where, v_t is an independent and distributed random vector generated from all the random nonce r of previous blocks with mean 0 and $\Delta(L)$ is the matrix polynomial in the lag operator L. To compute the future expectations, a stationary stochastic process for the exogenous variables has been postulated as

$$\mathcal{X}_t = \wedge(L)_{v_t} \tag{12}$$

where, $\wedge(L)$ is the matrix polynomial in the lag operator L.

It is assumed by the model that $\wedge(L)$ and previous values of $\mathcal{X}_t$ and y_t at time t are known. The random vector generated from r is evenly distributed with mean 0 and covariance matrix Ω. We have used normalization so that $\wedge(L)$ and $\Delta(L)$ are equal in identity matrices.

C. Obtaining the Solution

The solution method used here provides coefficients of constrained simultaneous VARMA model features [44] which are advantageous to be used in full information of ML estimation of EM problems. The process is stationary and is given by

$$B_0 y_t = X(L) y_t + Y(L) y_t + Z(L) v_t \tag{13}$$

where

$$X(L) = B_0 \theta_0^{-1}(\theta_0 - \Theta(L)) \tag{14}$$

$$Y(L) = \left(B_0 \theta_0^{-1} \beta_2(L) + \tau\right) \wedge^{-1} \Theta(L) \tag{15}$$

$$Z(L) = \left(B_0 \theta_0^{-1} \beta_1(L) + I\right). \tag{16}$$

In (14)–(16), $X(L)$, $Y(L)$, and $Z(L)$ are the matrix polynomials in lag operator (L) which are defined with other three polynomials $\Theta(L)$, $\beta_2(L)$, and $\beta_1(L)$ with zero order coefficient matrix θ_0 which are defined as

$$D(L) = B_1 L + \cdots + B_p L^p + (B_0 + A_0) + A_1 L^{-1} + \cdots + A_p L^{-p}. \tag{17}$$

Assuming that, there is a unique factorization of $D(L)$

$$D(L) = \phi(L^{-1})\Theta(L) \tag{18}$$

where, $\phi(L^{-1}) = I + \cdots + \phi_p L^{-p}$ and $\Theta(L) = \Theta_0 + \Theta_p L^p$ are real and the roots of the determinantal polynomials of $\Theta(L)$ and $\phi(L)$ lies outside the unit circle. Further, β_1 and β_2 are given by

$$\beta_1(L) = \left[\left(\phi\left(L^{-1}\right)\right)^{-1} \Delta(L)\right]^{+} \tag{19}$$

$$\beta_1(L) = \left[\left(\phi\left(L^{-1}\right)\right)^{-1} \tau\, \Delta(L)\right]^{+}. \tag{20}$$

In (19) and (20), $[.]^{+}$ denotes that all the terms with negative power of L have been removed.

The process for factorizing (18) is iterative and does not depend on the roots of the determinantal polynomials; rather a set of identities are implied which are obtained by equating the coefficients of the like-powered lag operators as

$$D_{-k} = \sum_{i=0}^{\min(p,m-k)} \phi_{k+i}\Theta_i, \quad k = 0, \ldots, p \tag{21}$$

$$D_k = \sum_{i=0}^{\min(m,p-k)} \phi_i \Theta_{i+k}, \quad k = 0, \ldots, m. \tag{22}$$

For a given $\Theta(L)$, $\phi(L^{-1})$ can be calculated following (21) as

$$\phi_k = \left[D_{-k} - \sum_{i=0}^{\min(p,m-k)} \phi_{k+i}\Theta_i, k = 0, \ldots, p\right]\theta_0^{-1}. \tag{23}$$

$$\begin{aligned}\mathcal{L}(\mathcal{H}) - \mathcal{L}(\mathcal{H}_n) &= \ln \sum P\big(\mathcal{H}|r.d.p.t\big) - P\big(r.d.p.t|\mathcal{H}\big) - \ln P\big(\mathcal{H}_n|r.d.p.t\big) \\ &\geq \sum P\big(\mathcal{H}|r.d.p.t\big) P\big(r.d.p.t|\mathcal{H}\big) \ln \frac{P\big(\mathcal{H}|r.d.p.t\big) P\big(r.d.p.t|\mathcal{H}\big)}{P\big(\mathcal{H}_n|r.d.p.t\big)} - \ln P\big(\mathcal{H}_n|r.d.p.t\big) \\ &\triangleq \Delta(\mathcal{H}|\mathcal{H}_n)\end{aligned} \tag{5}$$

Similarly, for a given $\phi(L), \Theta$ can also be calculated using (22). The process generates this form of equations for a series of successive approximations. Let $\Theta_{(j)}\ (L)$ and $\phi_{(j)}\ (L)$ are the values of $\Theta(L)$ and $\phi(L)$ at jth iteration, respectively. We then perform the following process for the convergence of the values.

Step 1: Initialize the process with value of $\Theta_{(0)}\ (L)$ and $\Theta(L)$ for the first iteration. There is no theoretical guideline for the selecting of the initialization value. Though, with the experimentation perspective we find it convenient to initialize the value with

$$\Theta_{(0)}(L) = D_0 + D_1 L + \cdots + D_p L^p \text{ or } \phi_{(o)}(L) = I(\text{Identity Matrix}). \quad (24)$$

Step 2: Compute $\phi_k^{(j)}$ using (21).

Step 3: Compute $\Theta_k^{(j)}$ using (21).

Step 4: Stop the process if $\Theta_k^{(j)}$ and $\Theta_k^{(j-1)}$ are within the specified tolerance range of variance. Else, repeat the process with increase j by 1.

Finally, we adopt the normalization concept of $\phi_0 = I$ and impose on the factorization for ease of convergence. Thus, at the mth iteration for the convergence

$$\phi(L) = \phi^{(m)}(L)\left[\phi_0^{(m)}\right]^{-1} \quad (25)$$

and

$$\Theta(L) = \left[\phi_0^{(m)}\right]\Theta^{(m)}(L). \quad (26)$$

The number of iterations m uses the identities k with a specific range of p to obtain a sure convergence. Once the value of m increases the minimum function in (23) provides the ranged summation for the convergence.

III. Results and Discussion

The present world of technology is emphasizing IoT applications considering cloud-fog-edge to be the sub parts. Therefore, any system or method development should be compatible with the above three computing environments, namely cloud, fog, and edge, considerably. In these environments, devices are dynamic. To access a data, they must start a transaction and create a data block with their {device_id, data_id, timestamp}. This data block is then verified using the proposed expectation-based consensus method to add data in the blockchain. Before starting the consensus process, the proposed work helps to calculate the precise expectation of a successful data validation and finally the data access. As this process is statistically sound, complexity becomes less and therefore the resource constrained devices can be easily managed for longer duration.

In the present case, performances are analyzed in three different aspects: 1) memory consumption; 2) electricity consumption; and 3) convergence time, to reach a guaranteed consensus. In this paper, we have also implemented a protocol named PoW as discussed elsewhere [45]. The implementation parameters in the expectation-based consensus method used in this paper are shown in Table I. Further, to check the validity and the merit of the proposed consensus expectation model, we have repeated the works described in [27]–[30] and the results are compared. These earlier works are, stochastic-based consensus, guaranteed resilient consensus, asymptotic model for linear systems, and bounded control protocols, respectively. Here, we have considered 50 blocks of data for doing the aforementioned experimental process.

TABLE I
Implementation Metrics

Consensus protocol	Proof-of-work as implemented in [45]
Geographic distribution of nodes	Campus area network (CAN) environment, 10 nodes
Hardware environment of all peers	3.3 GHz, 16 GB RAM, Octa-core, 2 TB HDD
Network model	We implemented with three firewalls, three access points, 10 routers
Number of nodes involved in the test transaction	5
Test tools and framework	Hyperledger composer (smoke tests)
Type of data store used	CouchDB

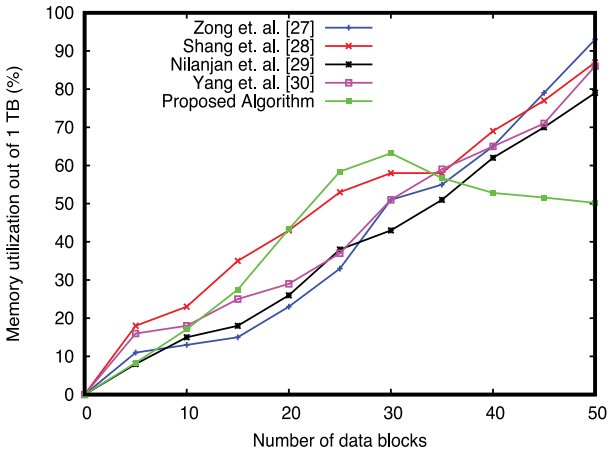

Fig. 3. Comparison of memory utilization.

For PoW to be implemented, we have used 1 TB of memory (L1-L2 cache) for the experimentation of the expectation-based algorithm. In this algorithm, PoW stores the hash values temporarily in the memory to check the convergence first. This memory utilization for implementing this algorithms is experimentally measured first and plotted in Fig. 3 as a function of number of data blocks. Figure also contains the results on the memory usage under identical conditions using the previously discussed four algorithms as well.

As it can be seen from Fig. 3, all the algorithms in comparison as well as the proposed one shows the same trend in the memory consumption as the number of blocks increases upto 30 blocks. However, as the number of blocks increased beyond 30, the proposed algorithm outperform other 4 algorithms in terms of memory usage because it uses L1-L2 cache to store the previous PoW calculations with expectation values used for further blocks. It consumes less memory for 35 blocks and thereafter then maintains a saturated value of memory utilization. In contrary, memory usage of the algorithms considered for the comparisons monotonically increases

TABLE II
COMPARISON OF EXISTING SOLUTION VERSUS CONTRIBUTION

Existing Progress	Problems	Solution by our contribution
Mainstream blockchain and byzantine issues [25], [26], [27]	Sure convergence problem	A statistical method to solve the expectation value of a mathematical puzzle in PoW.
Multiagent behavior based approach [28],[29],[30],[31]	Time consuming, cooperation tracking, not supportive for non-linear dynamics	Less complex and less time consuming and convergence is fast, supports non-linear dynamics with heterogeneous agents
Collective behavior and discrete opinion base [32],[33],[34],[35],[36]	Time consuming and complex	Less complex as it depends upon mathematical calculation only
Multiple delta operator base [37],[38]	Ill-conditioning problems when the shift operator is applied to represent the discrete-time system	Statistical method will provide precise calculation of iterative expectation values of convergence

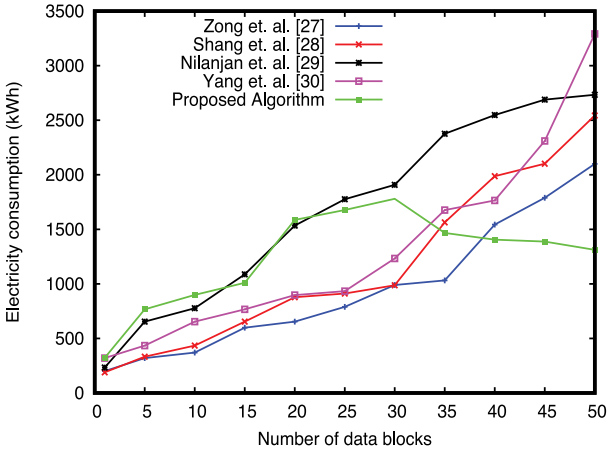

Fig. 4. Comparison of energy consumption.

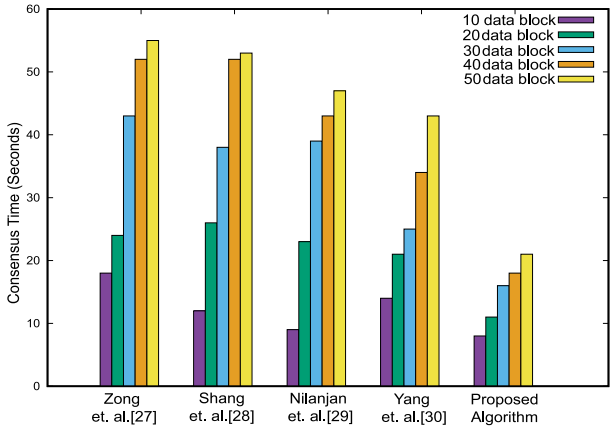

Fig. 5. Comparison of consensus time.

with number of blocks as they do not consider cache coherence and expected PoW. Statistically the proposed model uses <23.3% of memory consumption.

Energy consumption is another major factor in PoW to be considered for the scrutiny of any algorithm. So, the next parameter considered was the energy consumption of various algorithms as function of number of blocks. It is worth to mention that, energy consumption has been measured in average KWh only for consensus time. The results are shown in Fig. 4 for the proposed and existing algorithms.

It can be seen from Fig. 4 that the energy consumption of the proposed model for the initial blocks of data are higher than the other algorithms. It goes through a maximum (50% blocks generation) and reduces the energy consumption there after as the data blocks increases. Therefore, the average energy consumption of the proposed method is less as compared to others. Statistically, about the estimated energy consumption was <21.34% as PoW solution converges earlier as compared to algorithms in comparison.

Lastly, every operation has to be stopped after some time. Here, it is the "convergence time" to a get a consensus value. we have also estimated the convergence timing for all the five algorithms including the proposed method to get a consensus value. The consensus time for the algorithms are depending upon multiagent (multiple devices as per cloud environment concern) interactions in the algorithms which has been assumed to be identical for all. Therefore, we have considered consensus time delay to check the speed of operation. The result is shown in Fig. 5, the colored bars are used for variable data blocks.

The result shown in Fig. 5 depicts that with the help of proposed statistical method of EM, PoW converges faster. It has been observed that the proposed statistical method reduces the consensus delay by 13.8%.

The measured memory consumption, energy usage, and time convergence of the proposed signifies the fast processing of the method with less energy consumption and memory usage. Hence, the proposed method is efficient and powerful in terms of memory utilization, energy consumption, and convergence time.

We have analyzed our contribution as compared to the existing progress of the related work which is summarized in Table II.

IV. CONCLUSION

In this paper, we successfully executed a novel algorithm with PoW based on statistical likelihood maximization and

polynomial matrix factorization. It considered expectation of a guaranteed consensus and used linear rational expectation model and Jensen's equality to reach a logical conclusion. The advantage of the presented mathematical model is that it is flexible and exhibits better efficiency in memory usage, energy consumption, and converge time. Statistically, the presented work shows 23.3% less memory consumption, 21.34% less energy consumption, and 13.8% reduced convergence time. The results show that the proposed method is acceptable for cloud environments where devices are dynamic and constrained with different conditions and resources. Hence, the proposed work is efficient in cloud-fog-edge applications. There exists a number of open research questions related to the presented work that require future investigation. One important question is the behavior of the proposed system in presence of hybrid network consisting irrational multiagents. Moreover, the applicability of smart contracts for improvement in system adaptability can be considered as a future perspective. We hope our research will yield a new insight to the use of blockchain technology.

REFERENCES

[1] *The Great Chain of Being Sure About Things*, Blockchain Econ., 2015. [Online]. Available: https://gcalhoun.files.wordpress.com/2015/11/15-10-31-e-blockchains-the-great-chain-of-being-sure-about-things.pdf

[2] M. Niranjanamurthy, B. N. Nithya, and S. Jagannatha, "Analysis of blockchain technology: pros, cons and SWOT," *Clust. Comput.*, pp. 1–15, Mar. 2018. doi: 10.1007/s10586-018-2387-5.

[3] *Real World Blockchain Applications*. Accessed: Nov. 20, 2018. [Online]. Available: https://blockgeeks.com/guides/blockchain-applications-real-world/

[4] D. Li, Z. Cai, L. Deng, X. Yao, and H. H. Wang, "Information security model of block chain based on intrusion sensing in the IoT environment," *Clust. Comput.*, pp. 1–18, Mar. 2018. doi: 10.1007/s10586-018-2516-1.

[5] N. Radziwill, "Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world," *Qual. Manag. J.*, vol. 25, no. 1, pp. 64–65, 2018. doi: 10.1080/10686967.2018.1404373.

[6] *Blockchain for Dummies*. Accessed: Nov. 20, 2018. [Online]. Available: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=XIM12354USEN

[7] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[8] J. Pan *et al.*, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *CoRR*, vol. abs/1806.06185, pp. 1–14, Jun. 2018.

[9] J. Wang, A. Hester, J. Pan, and Y. Liu, "CreditCoin: Secure resource management using blockchain and smart contracts for edge-IoT system," in *Proc. Nat. Security Agency (NSA) NCCP CyberEd Workshop*, 2018.

[10] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, "Decentralized consensus for edge-centric Internet of Things: A review, taxonomy, and research issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2017. doi: 10.1109/ACCESS.2017.2779263.

[11] A. Stanciu, "Blockchain based distributed control system for edge computing," in *Proc. 21st Int. Conf. Control Syst. Comput. (CSCS)*, 2017, pp. 667–671.

[12] B.-K. Zheng *et al.*, "Scalable and privacy-preserving data sharing based on blockchain," *J. Comput. Sci. Technol.*, vol. 33, no. 3, pp. 557–567, 2018. doi: 10.1007/s11390-018-1840-5.

[13] J. Sun, J. Yan, and K. Z. K. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financ. Innov.*, vol. 2, p. 26, Dec. 2016. doi: 10.1186/s40854-016-0040-y.

[14] *Blockchain and Fog: Made for Each Other*. Accessed: Nov. 20, 2018. [Online]. Available: https://blogs.cisco.com/innovation/blockchain-and-fog-made-for-each-other

[15] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018. doi: 10.1109/ACCESS.2017.2757955.

[16] Z. Xiong *et al.*, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, to be published.

[17] *Bitcoin Mining Hardware Guide*. Accessed: Nov. 20, 2018. [Online]. Available: https://www.bitcoinmining.com/bitcoin-mining-hardware/

[18] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan. *Proofs of Useful Work*. Accessed: Feb. 2017. [Online]. Available: https://iacr.org/

[19] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Convent. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, 2018, pp. 1545–1550.

[20] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, 2017, pp. 2567–2572.

[21] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE 6th Int. Congr. Big Data BigData Congr.*, 2017, pp. 557–564.

[22] P. Ekparinya, V. Gramoli, and G. Jourjon, "Double spending risk quantification in private, consortium and public ethereum blockchains," *CoRR*, vol. abs/1805.05004, May 2018.

[23] *How Blockchain Works*. Accessed: Nov. 20, 2018. [Online]. Available: https://spectrum.ieee.org/computing/networks/how-blockchains-work

[24] *What is Proof-of-Work*. Accessed: Nov. 20, 2018. [Online]. Available: https://www.bitcoinmining.com/what-is-proof-of-work

[25] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Gener. Comput. Syst.*, Sep. 2017. [Online]. Available: https://doi.org/10.1016/j.future.2017.09.023

[26] Y. Shang, "Resilient multiscale coordination control against adversarial nodes," *Energies*, vol. 11, no. 7, p. 1844, 2018.

[27] X. Zong, T. Li, and J. Zhang, "Consensus conditions of continuous-time multi-agent systems with time-delays and measurement noises," *Automatica*, vol. 99, pp. 412–419, Jan. 2019.

[28] Y. Shang, "Resilient consensus of switched multi-agent systems," *Syst. Control Lett.*, vol. 122, pp. 12–18, Dec. 2018. doi: 10.1016/j.sysconle.2018.10.001.

[29] N. R. Chowdhury, S. Sukumar, and D. Chatterjee, "A new condition for asymptotic consensus over switching graphs," *Automatica*, vol. 97, pp. 18–26, Nov. 2018. doi: 10.1016/j.automatica.2018.07.018.

[30] T. Yang, Y. Wan, H. Wang, and Z. Lin, "Global optimal consensus for discrete-time multi-agent systems with bounded controls," *Automatica*, vol. 97, pp. 182–185, Nov. 2018.

[31] Z. Yu, H. Jiang, X. Mei, and C. Hu, "Guaranteed cost consensus for second-order multi-agent systems with heterogeneous inertias," *Appl. Math. Comput.*, vol. 338, pp. 739–757, Dec. 2018. doi: 10.1016/j.amc.2018.06.031.

[32] H. X. Yang, W. X. Wang, Y. C. Lai, and B. H. Wang, "Convergence to global consensus in opinion dynamics under a nonlinear voter model," *Phys. Lett. A*, vol. 376, no. 4, pp. 282–285, 2012.

[33] H.-Y. Xu, Y.-P. Luo, J.-W. Wu, and M.-C. Huang, "Hierarchical centralities of information transmissions in reaching a consensus," *Phys. Lett. A*, vol. 383, no. 5, pp. 432–439, 2018. [Online]. Available: https: doi.org/10.1016/j.physleta.2018.11.013

[34] R. L. Berger, "A necessary and sufficient condition for reaching a consensus using DeGroot's method," *J. Amer. Stat. Assoc.*, vol. 76, no. 374, pp. 415–418, 1981. doi: 10.1080/01621459.1981.10477662.

[35] M. Ye, J. Liu, B. D. O. Anderson, C. Yu, and T. Başar, "On the analysis of the DeGroot–Friedkin model with dynamic relative interaction matrices," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11902–11907, 2017. doi: 10.1016/j.ifacol.2017.08.1426.

[36] Y. Shang, "Hybrid consensus for averager-copier-voter networks with non-rational agents," *Chaos Solitons Fractals*, vol. 110, pp. 244–251, May 2018.

[37] S. Chen, D. W. C. Ho, and M. Liu, "Consensus protocol for multiple delta operator systems," *Syst. Control Lett.*, vol. 107, pp. 1–8, Sep. 2017. doi: 10.1016/j.sysconle.2017.07.001.

[38] D. Zheng, H. Zhang, J. A. Zhang, and G. Wang, "Consensus of multi-agent systems with faults and mismatches under switched topologies using a delta operator method," *Neurocomputing*, vol. 315, pp. 198–209, Nov. 2018. doi: 10.1016/j.neucom.2018.07.017.

[39] S. Borman. (2009). *The Expectation Maximization Algorithm a Short Tutorial*. [Online]. Available: www.emtut@seanborman.com

[40] S. S. Dragomir and C. J. Goh, "Some counterpart inequalities for a functional associated with Jensen's inequality," *J. Inequal Appl.*, vol. 1, no. 4, pp. 311–325, 1997.

[41] C. H. Whiteman, *Linear Rational Expectation Models: A User's Guide*. Minneapolis, MN, USA: Univ. Minnesota Press, 1983.

[42] R. C. Fair and J. B. Taylor, "Solution and estimation of dynamic non-linear rational expectations models," *Econometrica*, vol. 51, no. 4, pp. 1169–1185, 1983.
[43] O. Blanchard and C. Kahn, "The solution of linear difference models under rational expectations," *Econometrica*, vol. 48, no. 5, pp. 1305–1311, 1980.
[44] M. Guidolin and M. Pedio, "Vector autoregressive moving average (VARMA) models," in *Essentials of Time Series for Financial Applications*. London, U.K.: Academic, Feb. 2018.
[45] *Implementing Blockchain and Cryptocurrency With PoW Consensus Algorithm*. Accessed: Nov. 20, 2018. [Online]. Available: https://medium.com/coinmonks/part-3-implementing-blockchain-and-cryptocurrency-with-powconsensus-algorithm-d9b8cb928e3e

Gulshan Kumar (M'18) received the B.Tech. degree in computer science engineering from the Amritsar College of Engineering, Amritsar, India, in 2009, and the M.Tech. and Ph.D. degrees from Lovely Professional University, Phagwara, India, with an area of specialization in position and location computation in wireless sensor networks.

He is an Associate Professor with Lovely Professional University. He has authored or coauthored many publications in well renowned international journals and conferences.

Rahul Saha (M'18) received the B.Tech. degree in computer science engineering from the Academy of Technology, Hooghly, India, and the M.Tech. and Ph.D. degrees from Lovely Professional University, Phagwara, India, with an area of specialization in cryptography and position and location computation in wireless sensor networks.

He is an Associate Professor with Lovely Professional University. He has authored or coauthored many publications in well renowned international journals and conferences.

Mritunjay Kumar Rai received the Master of Engineering degree in digital system from the Motilal Nehru National Institute of Technology, Allahabad, India, and the doctoral degree from the ABV Indian Institute of Information Technology and Management, Gwalior, India.

He was an Associate Professor with Lovely Professional University, Phagwara, India. He has authored or coauthored over 50 research papers in reputed international conferences and international journals. His current research interests include wireless networks, network security, and cognitive radio networks.

Reji Thomas received the doctoral degree from the Indian Institute of Technology Delhi, New Delhi, India.

He is a Professor with Lovely Professional University, Phagwara, India. His current research interests include logic, memory, and energy storage devices.

Tai-Hoon Kim (M'17) received the B.E. and M.E. degrees from Sungkyunkwan University, Seoul, South Korea, and the Ph.D. degree from the University of Bristol, Bristol, U.K., and the University of Tasmania, Hobart, TAS, Australia.

His current research interests include security engineering for IT products, IT systems, development processes, and operational environments.