Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-022-01422-4

An improved Kalman filter using ANN-based learning module to predict transaction throughput of blockchain network in clinical trials

Lei Hang[1] · Israr Ullah[2] · Jun Yang[3] · Chun Chen[3]

Received: 8 August 2022 / Accepted: 17 November 2022

Abstract

Clinical trials have been made transparent and accessible because of the widespread adoption of blockchain technology. Its distinctive characteristics, such as data immutability and transparency, could increase public trust in a fair and transparent manner among all stakeholders. However, blockchain systems cannot handle the requirement of processing huge volumes of data in real time. Scalability becomes a severe issue when implementing decentralized applications for clinical studies. With an abrupt expansion in the number of transaction exchanges happening consistently and the capital associated with those exchanges, there is an urgent demand for developers and users to know blockchain systems' performance limits to determine if requirements can be fulfilled; however, little is known about the prediction of blockchain system behaviors. This paper shows the feasibility of using machine learning technologies to predict the transaction throughput of blockchain-based systems in clinical trials. A learning to prediction model is proposed, in which the Kalman filter is used to predict the transaction throughput, and the Artificial Neural Network (ANN) is utilized to enhance the Kalman filter's prediction accuracy. A real dataset generated from a clinical trial testbed using Hyperledger Fabric is utilized to demonstrate the feasibility of the proposed approach. Moreover, we compare the Kalman filter with other learning modules, and the results indicate that the ANN performs best. Furthermore, we apply the proposed approach to different blockchain platforms, and the experiment results indicate the efficiency and universality of the designed approach.

Keywords Blockchain · Kalman filter · Artificial neural network · Learning to prediction · Transaction throughput · Clinical trials

1 Introduction

As an emerging distributed ledger technology, blockchain takes into account safe and verifiable transactions without requiring a reliable third party [1]. All participants in a blockchain network contribute to a shared public record of all past and present trades. The business logic is represented by a chain of interconnected blocks known as transactions. When all nodes in the network attain consensus, every node updates the ledger with new blocks and keeps a duplicate copy [2]. Smart contracts on blockchain characterize programmable business logic and automate transactions without human interruption [3]. This technology has evolved into applications in different fields, such as supply chains, fintech, medical services, tourism industry, and the Internet of Things (IoT) [4–9].

To be beneficial, blockchain technology will need to support transaction rates on par with conventional database

✉ Chun Chen
acbbaz@sjtu.edu.cn

Lei Hang
hanglei@jejunu.ac.kr

Israr Ullah
israr.ullah@vu.edu.pk

Jun Yang
yangjun@xinhuamed.com.cn

1 Business School, Shanghai Normal University Tianhua College, Shanghai, China

2 Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan

3 Department of Otorhinolaryngology-Head&Neck Surgery, Xinhua Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai, China

management systems while offering some of the same transactional assurances. Implementing blockchain technologies to replace existing centralized servers remains a number of challenges. One of the most significant challenges is transaction processing capability [10], which results from the inability to confirm the entire transaction until all nodes in the network have completed and reached a consensus. Each node in a blockchain network is independent of the others' ledgers. The system's overall performance exhibits an apparent barrel effect since it mostly depends on the node with the lowest performance. Critical transaction throughput and latency constraints prevent blockchain technologies' application in large-scale deployment scenarios [11].

The Bitcoin network, for instance, creates a new block every 10 min and limits the size of individual blocks to 1 MB. As a result, the Bitcoin network can only process 7 transactions per second [12], making it inappropriate for fast-paced markets. While the average transaction confirmation time for Ethereum is now under 15 s [13], this time can expand tremendously when network circumstances shift. The Inefficient transaction processing capability in current blockchain platforms arises as a rigorous issue when creating blockchain-based clinical trial systems as they cannot meet requirements of high-throughput and high-user-concurrency in the production environment, which would result in generating a large number of transactions per second [14].

Blockchain technology can promote the transformation of clinical trials in various scenarios, including patient recruitment, consent traceability, persistent monitoring, data management, and data analysis [15]. Some scenarios are not sensitive to the speed requirement of data processing, while others require high demand to handle high-volume health records in real time. As a result, predicting system-level throughput during clinical trial system design will be critical to determine if these requirements can be fulfilled.

Predictions can be made using analytical solvers or simulation engines [16]. Over the past decade, prediction algorithms have been widely used in the price prediction of cryptocurrencies. For example, the authors in [17] utilize machine learning algorithms to predict the sign of price change of Bitcoin. They model the price prediction model as a binomial classification problem. Random forests and generalized linear models are leveraged in the experiment. The authors in [18] predict the Bitcoin price using Recurrent Neural Networks (RNNs), Long Short-term Memory (LSTM), and Autoregressive Integrated Moving Average (ARIMA) models.

Similarly, a Bayesian Neural Networks (BNNs) based approach [19] is proposed to predict the Bitcoin price in terms of blockchain information, including block size, hash rate, difficulty, etc. Some researchers apply prediction algorithms to estimate cryptocurrency prices rather than Bitcoin. For instance, the authors in [20] perform the prediction of Ether with Linear Regression (LR) and Support Vector Machine (SVM) by using a time series of daily Ether closing prices. Similarly, the authors in [21] present Ether's return rate predictive approach using the LSTM model.

To the best of the authors' knowledge, most existing studies focus on the cryptocurrency price prediction. Few studies investigate machine learning to predict non-functional properties of blockchain behaviors, including transaction throughput, transaction latency, and resource usage. In most of these existing studies, supervised machine learning models are used, in which labeled data is used to train the prediction model. The hidden relationship between the input and output parameters is captured by the trained prediction model. This relationship is further by the prediction model to forecast results for given input data, even in situations that have never been encountered before. However, when the machine may consider various input variables, a trained machine can be complex and suffer from an overfitting problem. Moreover, these algorithms have one thing in common: they are trained initially upon historical data. Consequently, the prediction model is fixed and only can be applied in the specified application domain after training.

In this study, we utilize the Kalman filter, a lightweight algorithm that can intelligently anticipate the system's current state using only historical data about its prior states. However, the problem with the Kalman filter is the lack of ability to adapt to dynamic environments and changing elements, which is a common issue for conventional prediction algorithms, as mentioned. Many ensemble methods have been developed to predict and classify [22], such as mixture-of-experts and layered generalization. Simple machine learning methods perform worse than ensemble methods; for example, layered generalization outperforms single neural network approaches [23]. Another ensemble method is a mixture of experts, which improves performance by combining various statistical estimations. The prediction accuracy of the mixture-of-experts strategy is higher than the other methods due to statistical estimates.

Enabling the Kalman filter to cope with dynamic data or changing environmental conditions is challenging as the network condition is continuously changing. This paper proposes a new methodology for tuning the Kalman filter parameters based on ANN to improve the prediction accuracy of transaction throughput. To address the issue of static parameters, we add a learning module to the conventional Kalman filter. The effectiveness of the proposed method is verified with a clinical trial testbed implemented on Hyperledger Fabric [24]. The experiment results show that the learning module can increase the Kalman filter algorithm's prediction accuracy. Moreover, we test the proposed approach with other blockchain networks to validate the universality. The results indicate that modeling concepts are well-aligned with component-based development and encourage the reuse of constructed models and components.

This paper makes three-fold contributions which are summarized as follows:

- Novelty: A learning to prediction module is presented to estimate the transaction throughput of blockchain systems. The Kalman filter predicts the transaction throughput, and the ANN enhances the Kalman filter's prediction accuracy.
- Feasibility: A clinical trial testbed implemented in Hyperledger Fabric verified the feasibility of the proposed method. The outcomes of the experiments demonstrate the effectiveness of the designed approach.
- Universality: The universality of the proposed approach has been validated by integrating it with other blockchain platforms. The results demonstrate that the proposed model performs well with trusty stability and reliability.

The rest of this paper is organized as follows: Section 2 overviews some related work. Section 3 illustrates the system architecture of the proposed approach. Section 4 details the development environment and the experiment setup. Section 5 discusses the experimental analysis and results of the proposed approach. Section 6 presents the limitation of this study and highlights some future research directions. Finally, Section 7 concludes the paper.

2 Related work

2.1 Kalman filter

With the advancement of computer storage and processing capacity, there has been a significant increase in machine learning algorithms and the invention of novel algorithms for handling various issues. Machine learning methods, ANNs, and deep learning have all been employed by various authors for various objectives. To minimize performance loss, a method must be in place to detect environmental changes and make relevant adjustments to the prediction algorithm, allowing the prediction algorithm to adapt to constantly changing environmental conditions. By continually combining a learning module with a prediction algorithm, this approach may be utilized to modify its performance. The authors of [25] present two unique Kalman filter-based techniques for reliably estimating mobile robot attitudes. The first approach optimizes Kalman filter parameters using the measurements in a simulation environment based on low-cost accelerometers and gyroscopes. The second approach measures magnitudes and utilizes fuzzy logic to modify the filter parameters in real time. The experiment validates the proposed algorithms under various dynamic conditions.

Researchers have adjusted the Kalman filter over time to improve its performance. The extended Kalman filter, a nonlinear version of the conventional Kalman filter, is one of these methods. The existing mean and covariance estimations are linearized using the extended Kalman filter. The iterated extended Kalman filter is a variation of the Kalman filter that uses a maximum posterior estimate to produce state estimations. In [26], the Gauss–Newton method is utilized to update the Kalman filter parameters. When several factors are considered, the ensemble Kalman filter is a recursive form of the Kalman filter that comes in helpful. With the exception of the ensemble Kalman filter, which assumes that all probabilities are Gaussian [27, 28], the extended and particle filters have identical features. A hidden Markov model (HMM) [29] is presented and trained to utilize static accelerometer measurements for the same purpose. The Kalman filter uses other techniques, such as the unbiased finite-impulse response (UFIR) filter. Although the fusion filter is dependable, it may not always yield the most remarkable results. The fusion filter decreases error by merging the Kalman and UFIR filters [30]. As a result, the Kalman filter has been utilized to tackle problems in various domains.

2.2 ANN

Machine learning algorithms are essential and have been utilized in various domains. The authors in [31] introduce a new technique for generating alternative climatic dataset settings based on the K-nearest neighbor algorithm. Text and image classification are both accomplished with a novel method termed K-nearest Neighbor (KNN) multi-label multi-instance learning [32]. The authors of [33] discuss machine learning strategies for massive data analysis classification.

As a general-purpose tool that can be applied to a wide range of problems, ANN has become one of the most extensively used prediction algorithms. The ANN can be used to solve issues with classification, regression, clustering, and pattern recognition. The number of classes in a classification task will dictate the size of the hidden layer and the number of neurons used in the output layers. ANN algorithms can handle problems such as classification, regression, clustering, pattern recognition, forecasting, and time series [34]. The perceptron, an atomic functional unit of ANN, simulates the neuron's function. Perceptron operations include calculating inputs with suitable weights, summating outcomes with a bias, and generating output using a sigmoid activation function. An ANN architecture is formed by connecting the required number of perceptrons in a layered method, depending on the size and type of the problem (input layer, hidden layer, output layer).

Learning in ANN algorithms entails systematically altering the weights of perceptrons in the network using various methodologies such as error backpropagation, gradient calculation methods [35], etc. The performance and accuracy of traditional ANN algorithms are determined by input preprocessing, normalization, and feature extraction procedures. Trial-and-error procedures are the most popular methods for determining the number of layers, the number of neurons in each layer, the activation function, and the connection between the layers. Processing power limits historically restricted the size of an ANN network. With the evolution of modern high-speed multi-core computers, the ANN network may virtually have any size, leading to a new field known as deep learning or deep neural networks beneath the umbrella of neural networks.

2.3 Parameter tuing of Kalman filter

To allow the prediction algorithm to adapt to changing environmental circumstances, a method must be implemented to detect environmental changes and make necessary adjustments to the prediction algorithm to maximize forecasting performance. This technique can be implemented by merging the prediction algorithm with a learning module to tune its performance over time. However, only a few studies in the literature are closely related to this topic. The authors in [36] devise a fuzzy inference-based method for improving the performance of the Kalman filter methodology for accurate humanoid robot orientation estimation. The Kalman filter can successfully eliminate noise from gyro sensor signals in static situations, allowing the robot's exact orientation to be predicted. The gyro sensors' data get noisy when the robot is moving. They use accelerometer sensors to determine the robot's present state to address this issue, and the Kalman filter approach is altered accordingly, employing the fuzzy algorithm. Similar to the Kalman filter approach, which is used for precise attitude estimation using data from the gyroscope and accelerometer sensors, the authors propose an Adaptive Neuro-fuzzy Inference System (ANFIS) [37].

2.4 Blockchain behaviour prediction

The research on investigating machine learning to predict blockchain behaviors is relatively slight. For example, the authors in [38] explore classifiers-Bayes, Random Forest, and Multi-Layer Perceptron to predict the time frame a miner node will accept and include a transaction to a blockchain in Ethereum. The authors in [39] use architectural performance modeling and simulation tools to predict the latency of blockchain-based systems. Established tools and techniques are used, and new blockchain-specific issues, such as the number of confirmation blocks and inter-block times, are explored. The authors in [40] propose a prediction model derived from Ethereum's "World State" core structure. The proposed model predicts the performance and storage of executing contracts based on transaction volume. The authors in [41] propose a queueing model to study the relationship between the fee offered by a transaction and its expected consolidation time. This model is validated with data extracted from the Bitcoin blockchain and discrete event simulations.

Table 1 compares the proposed approach with the existing studies reviewed above. It is evident from the table that none of the existing studies investigate machine learning techniques to predict the transaction throughput of blockchain systems. Besides, these studies only utilize a single trained model to predict blockchain behaviors. The learning to prediction model in this paper focuses on the learning and maintenance of many training models simultaneously by the learning module. When environmental triggers are discovered, each trained model in the prediction algorithm could be activated by changing the tunable parameters or replacing the trained model entirely.

3 System architecture of the proposed approach

3.1 System architecture

Prediction algorithms are typically trained on historical data to uncover hidden patterns and links between input

Table 1 Comparison of the proposed approach to existing studies

Name	Machine Learning Approach	Behavior	Target Blockchain
[38]	Classifiers-Bayes, Random Forest and Multi-Layer Perceptron	Transaction confirmation time	Ethereum
[39]	None	Transaction latency	Ethereum
[40]	Patricia tree	Number of the transaction, storage	Ethereum
[41]	Queueing model	Consolidation time	Bitcoin
Proposed approach	Kalman filter with ANN-based learning	Transaction throughput	Hyperledger Fabric

and output parameters. For all input data, the outcome is predicted using trained models. The prediction method performs well only when the input data and application situation is consistent with the training data. In other words, the current prediction technique prevents the trained model from adapting to changing and dynamic input conditions. We propose the learning to prediction model to overcome this constraint. Figure 1 presents the proposed approach's conceptual system architecture, consisting of the learning to prediction module, blockchain network, and transaction traffic measurement analyzer. The blockchain network comprises miscellaneous peers that copy the distributed ledger and a smart contract. The transaction traffic analyzer generates performance statistics and stores benchmark results in the benchmark DB. The learning module fine-tunes the prediction algorithm to increase forecast accuracy. The learning module supervises and evaluates the prediction module's performance by analyzing the given output parameters. External parameters that may affect the prediction algorithm's performance can be considered input parameters by the learning module. Afterward, the learning module could modify the specific parameters of the prediction algorithm accordingly to improve its forecasting performance as long as environmental triggers are observed after the output of the prediction algorithm and the current external factors are analyzed.

Figure 2 describes the block diagram of ANN-based learning with Kalman filter for transaction throughput prediction. The Kalman filter estimates the transaction throughput $T_{\_t}$ in a noisy environment. As transaction throughput is severely impacted by transaction latency, noise is introduced into the environment. The ANN-based learning module uses input parameters for the measured transaction throughput $Z_{\_t}$, transaction latency $L_{\_t}$, and actual transaction throughput $R_{\_t}$. By eliminating the noise, the Kalman filter can estimate the transaction throughput at the time. The performance of the Kalman filter algorithm is mainly affected by a variable parameter, namely Kalman gain K, which is regenerated in terms of

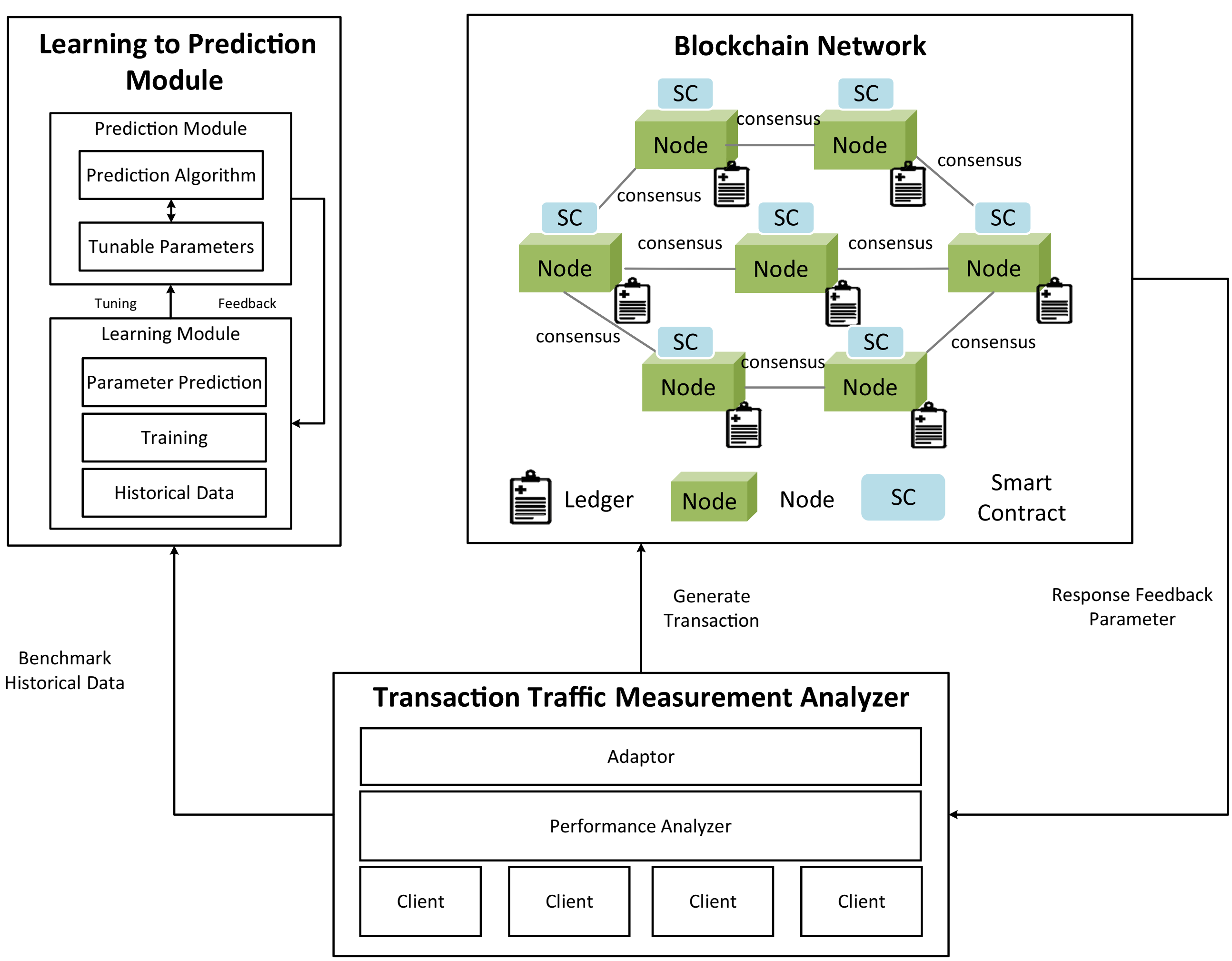


Fig. 1 The conceptual system architecture of the proposed approach

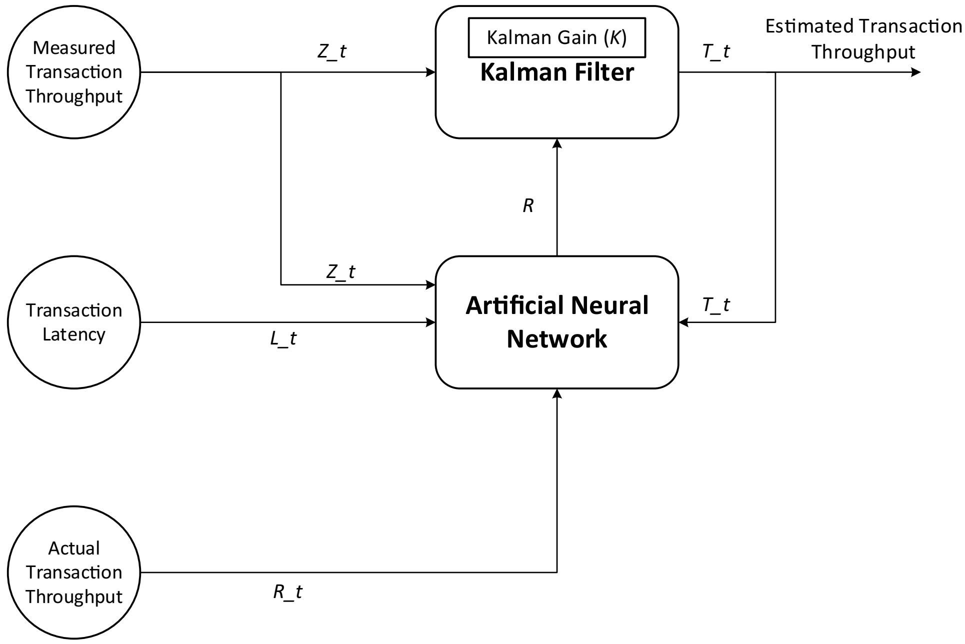


Fig. 2 Block diagram of ANN-based learning with Kalman filter for transaction throughput prediction

the variance matrix P and the calculable error R with each new iteration. It is the objective of the learning module to estimate the calculable error R so as to update the Kalman gain K dynamically.

Hyperledger Caliper [42] is utilized as the transaction traffic measurement analyzer to measure the performance of the blockchain. It is a universal blockchain benchmark framework that creates a standard interface layer for use with different blockchain implementations. Performance parameters, such as throughput, latency, and resource usage, are all included in Caliper's report after running its performance tests. In order to test customized applications, it also offers a wide variety of blockchain configurations, network settings, and use cases.

Figure 3 depicts the Caliper framework's layer-based architecture, which comprises the benchmark layer, the interface and core layer, the adaptation layer, and the network layer. The benchmark layer provides scenarios for verifying the blockchain's supporting infrastructure. The interface and core layer provides a Command Line Interface (CLI) for basic network administration tasks like starting and stopping the network. It provides other features, such as resource monitoring, performance analysis, and report creation. The adaption layer can connect the blockchain network

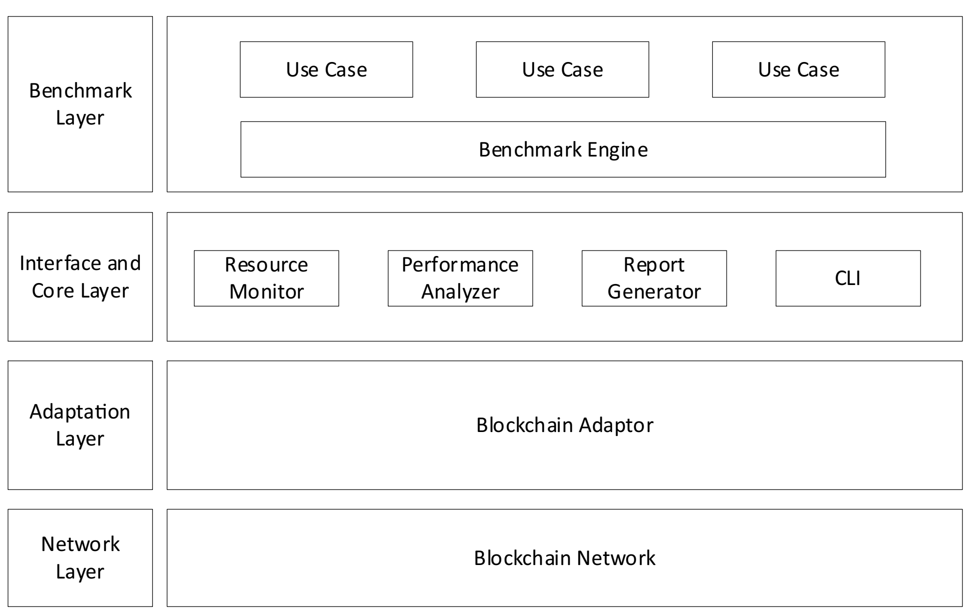


Fig. 3 The layer-based system architecture of Hyperledger Caliper

to external client apps by mapping activities like invoking or querying states from the ledger using relevant blockchain Software Development Kits (SDKs). It facilitates interoperability with other blockchain systems by translating blockchain backend operations into corresponding blockchain protocols via platform-specific adaptors. The network layer provides the blockchain infrastructure in which numerous peers hold distributed ledgers and smart contracts.

3.2 Conventional Kalman filter (Table 2)

In the commencement, Eq. (1) is used to calculate the estimated transaction throughput from the previously determined value:

$$T_k = A \cdot T_{k-1} + B \cdot u_k \tag{1}$$

The estimated transaction throughput is represented byT_k, where the state transition and control matrices are presented by A, and$B$, respectively. The transaction throughput at time k-1 is represented byT_{k-1}, and the control vector is represented byu_k.P_k, which reflects the covariance factor and determines the estimated transaction throughput T_k, as shown in Eq. (2).

$$P_k = A \cdot P_{k-1} \cdot A^T + Q \tag{2}$$

P_{k-1} is the previous covariance value with a process error Q, while A and A^T stand for the state transition matrix and its transpose. The Kalman gain (K_k) is calculated using the estimated transaction throughput and the updated covariance value, as shown in Eq. (3):

Table 2 Formula notes of Kalman Filter

Letter symbol	Description
T_k	estimated transaction throughput
T_{k-1}	previous transaction throughput
A	state transition matrix
A^T	state transition matrix transpose
B	control matrices
u_k	control vector
P_k	covariance factor
P_{k-1}	previous covariance factor
Q	process error
K_k	Kalman gain
H	observation matrix
H^T	observation matrix transpose
R	measurement error
z_k	current measured transaction throughput
T_e	updated transaction throughput
P_e	updated covariance factor

$$K_k = \frac{P_k \cdot H^T}{H \cdot P_k \cdot H^T + R} \tag{3}$$

The observation matrix, as well as its transpose, are represented by H and H^T, respectively. The measurement error is represented by R. The current measured transaction throughput at time k is presented as z_k. The updated transaction throughput for the next stage is calculated, as expressed in Eq. (4):

$$T_e = T_k + K_k(z_k - H \cdot T_k) \tag{4}$$

Equation (5) is used to update the covariance value for the next iteration:

$$P_e = \left(I - K_k \cdot H\right)P_k \tag{5}$$

With only previous state knowledge, the Kalman filter may estimate the actual state of the system. It regulates the weights assigned to the system's estimated state or sensing values by updating the Kalman gain K_k value in terms of the condition. Figure 4 depicts the Kalman filter's essential components and operations. Noise effects exist in any blockchain network and can significantly affect the transaction throughput measurement. This paper considers a noise-filled continuous throughout the measurement and takes Tt as the transaction throughput at time t. The Kalman filter incorporates a model that can forecast the system state, i.e., estimated transaction throughput, and then compare this value to the current measured transaction throughput value to predict the transaction throughput T_{t+1} at time t + 1.

3.3 ANN-based learning module for Kalman filter

The conventional Kalman filter algorithm has remarkable performance if the estimated error in the state measurement is static, as shown in Fig. 4. However, suppose the estimated error R in the measurements varies dynamically because of external factors like network congestion and latency. In that case, we must update the estimated measurement error R. Under these dynamic conditions, the traditional Kalman filter technique fails to forecast actual transaction throughput. The detailed structure of the proposed learning to prediction module for blockchain transaction throughput is given in Fig. 5. The ANN-based learning module takes three parameters as inputs, including transaction latency L_t, measured transaction throughput Z_t, and actual transaction throughput R_t. The predicted error in transaction throughput measurement is the ANN's generated output. This error is further divided by a constant factor F to compute the estimated error R. The measured transaction throughput Z_t and the updated value of R is then used as the input of the Kalman filter to enhance the prediction accuracy by adequately adjusting the Kalman gain K accordingly. The learning to prediction

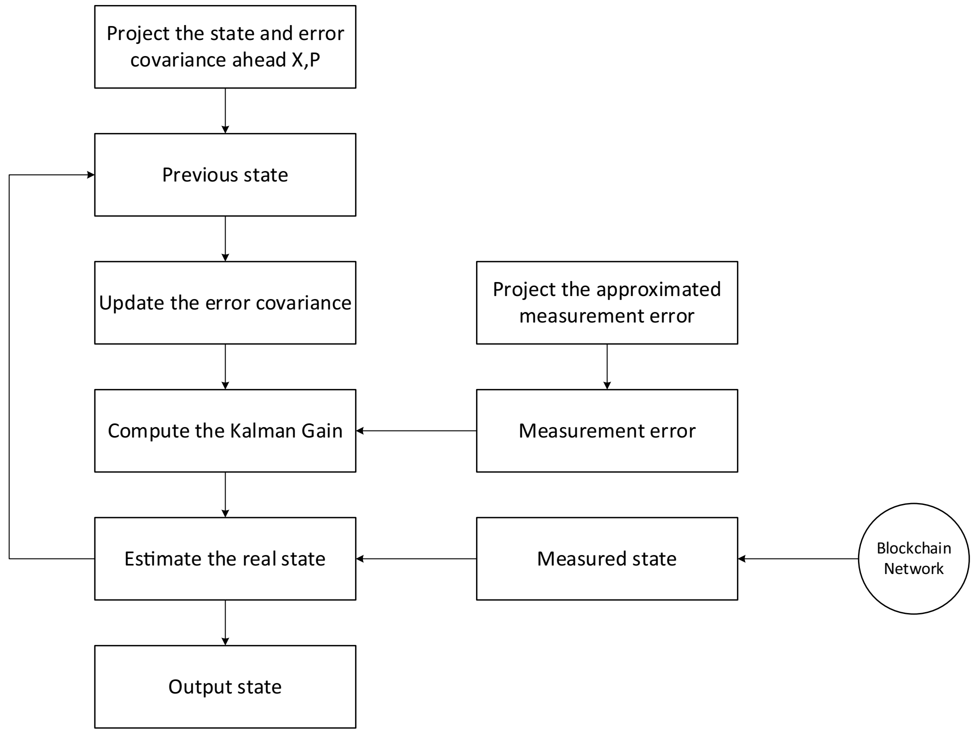


Fig. 4 Flow chart of the transaction throughput prediction using the Kalman filter

model allows the Kalman filter to estimate the actual transaction throughput under the changing network environment with a dynamic error rate.

4 Experiment setup

4.1 Clinical trial testbed

In this section, we utilize the blockchain-based clinical trial testbed from the previous work [6] to verify the efficiency and usability of the proposed approach. As represented in Fig. 6, the data is transparent and available to all stakeholders in the clinical trial testbed. All trial-related data, such as the clinical protocol, visit history, subject information, etc., are preserved in a complete and up-to-date form on the blockchain, allowing for tracking the enrolled subject throughout the clinical trial study. Off-chain storage, which is what the data lake is used for, is a remote data repository. Each trusted validating peer in the blockchain network keeps a copy of the distributed ledger to ensure consistency. The clinical director can access the study data privately from any connected device. To guarantee the system's security, all interaction between end users and the blockchain is encrypted and signed digitally.

The Membership Service Provider (MSP), a collection of cryptographic processes and protocols for creating and validating certificates or identities in the blockchain network, ensures access control on transactions and data in the clinical trial testbed. Different trial-related organizations have created and are using the blockchain network following corporately achieved and signed agreements. A controlled collection of individuals is referred to as an organization, including a home, a clinical site, and a Clinical Research Organization (CRO). In independent clinical trial research, an organization maintains its members under several organizational categories using a single MSP. The policies of the MSP can be used to assess identities issued within its jurisdiction.

Moreover, additional access control rules are set in the smart contract to permit or prohibit access to resources based on the user's identification related to a particular participant. As shown in Fig. 7, each rule has a description that explains the rule in simple terms. The participant identifies the different types of participants (e.g., subject, CRC, CRA) to which the rule applies. The resource specifies the kind of resource (e.g., subject info, eCRF) subject to the rule, the operation specifies the activity that the rule permits or forbids (e.g., READ, CREATE), and the action specifies the type of authorization (ALLOW or DENY). For instance, the CRA can only view the profile, whereas the CRC has access to the whole eCRF.

4.2 Development environment

The technological stack utilized to develop the learning to prediction module is shown in Table 3. This module is implemented with the Intel Core i5-8500 @ 3.00Ghz CPU,

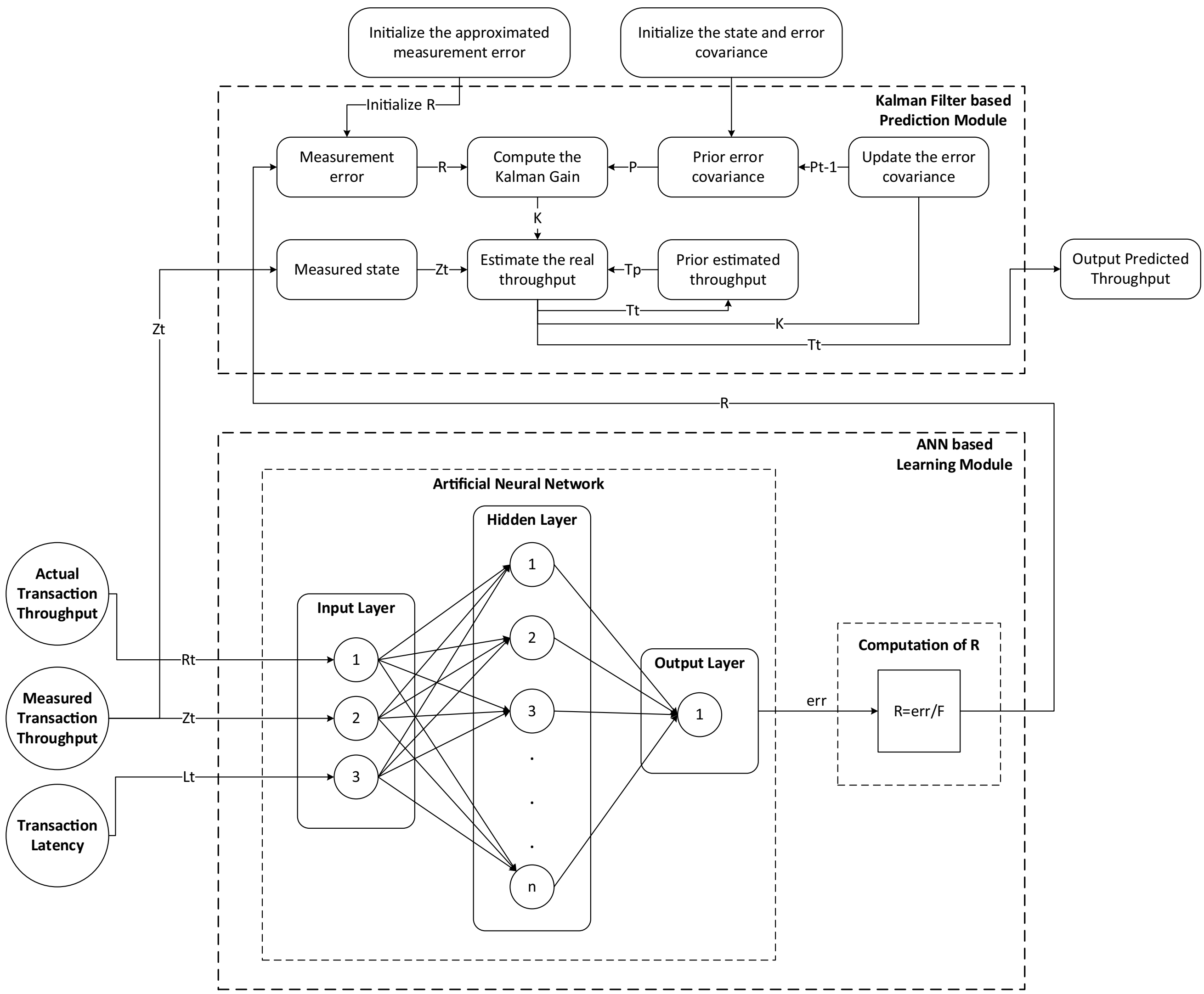


Fig. 5 Detailed diagram of the learning to prediction module

12 GB memory, in Windows 10 Enterprise 64-bits. C# is used as the programming language in Visual Studio Community. Accord. Neuro is used to implement the ANN, and Newtonsoft.Json is a JSON-based framework for.Net. The Kalman filter is implemented in the native C# language.

4.3 Setup for experiment

The default experiment setting and workload for evaluating the proposed approach are represented in Table 4. This study is conducted on a single-channel network with 4 organizations and 6 endorser peers. With a default block size of 10 transactions, a new block is generated every 250 ms. The default ordering service is Solo, with only one ordering node. In this experiment, LevelDB is used as the default state database. The rest of the experiment parameters are described in Table 4. The evaluation tests in this section are averaged over ten repetitions to reduce errors brought on by system overload and network congestion. Each round of testing has a transaction duration of 60 s. The experiment's scripts are updated to target the prototype's eCRF lab data creation function as it is the users' most commonly executed transaction.

4.4 Dataset setup

As presented in Table 5, the dataset used for the learning module has four features: transaction latency, send rate, transaction throughput, and error. We utilize the Hyperledger Caliper to perform the benchmark test with the clinical trial testbed and keep a record of these benchmark results to generate the dataset. This dataset has

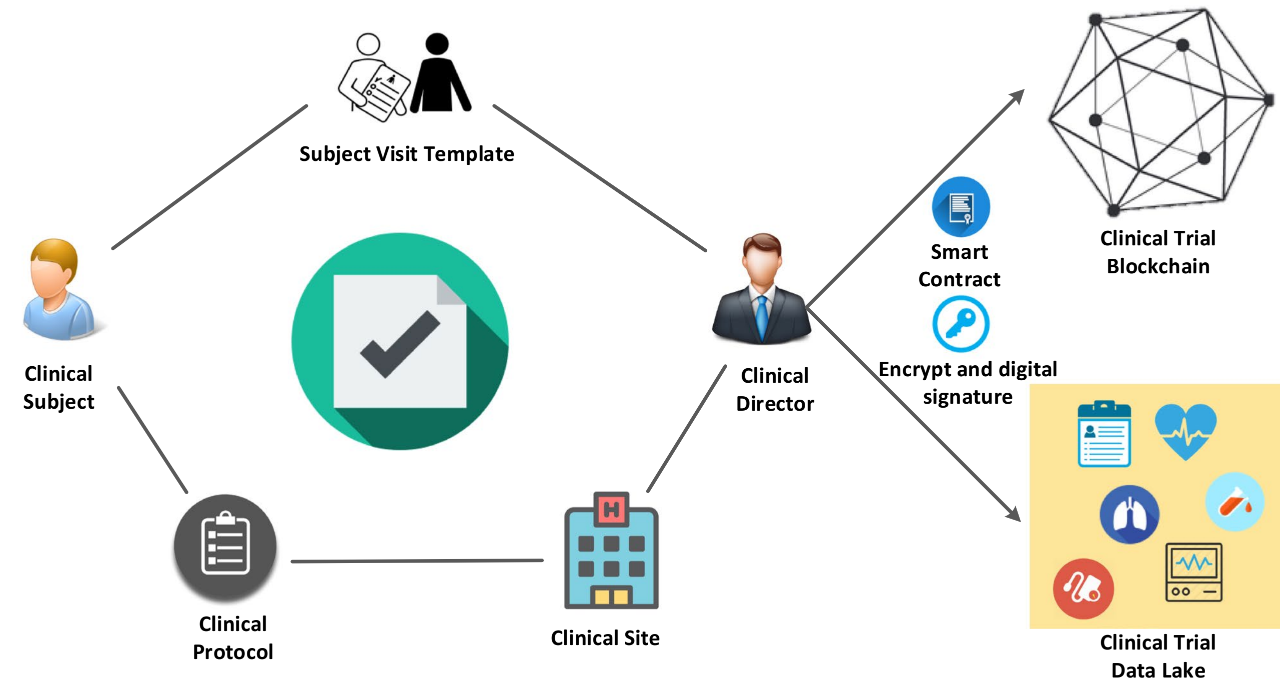


Fig. 6 Flow diagram of the trial-related process in the blockchain-based clinical trial testbed

10,080 rows, representing the performance benchmark profile throughout a week. Each row describes the benchmark statistics observed in 60 s. Two critical indicators for measuring the efficiency of a blockchain network are transaction throughput and latency. Transaction throughput, measured in transactions per second (tps), is the number of valid transactions the blockchain can process at a particular duration. The time it takes for a network as a whole to validate a transaction, including the time it takes for the transaction to be propagated across the network and settled as a result of the consensus, is the transaction latency. The send rate is the rate at which clients submit transactions. The error represents the difference between the send rate and transaction throughput.

```
rule CRC_to_eCRF{
        description: "Grant CRC access to the eCRF created"
        participant: "org.clinical.trial.CRC"
        operation: ALL
        resource: "org.clinical.trial.eCRF"
        action: ALLOW
}
rule CRA_to_eCRF{
        description: "Grant CRA access to the eCRF created"
        participant: "org.clinical.trial.CRA"
        operation: READ
        resource: "org.clinical.trial.eCRF"
        action: ALLOW
}
rule CRC_to_Subject {
        description: "Grant CRC access to the subjects created"
        participant: "org.clinical.trial.CRC"
        operation: ALL
        resource: "org.clinical.trial.Subject"
        action: ALLOW
}
...
```

Fig. 7 Sample access control rules in the smart contract

Different configurations are tested to determine the ideal training module for the ANN by altering the number of neurons within the hidden layer, learning rates, and activation functions. Experiments are conducted in many rounds for each network configuration for training, and average results are recorded to examine the random factor for initializing weights of the ANN. In addition, a fourfold cross-validation technique is used for each configuration across all tests to eliminate bias in training, as shown in Fig. 8. We partition the original dataset into four equal-sized sections for this experiment (2520 instances

Table 3 Development environment for the learning to prediction module

Component	Description
CPU	Intel Core i5-8500 @ 3.00 GHz
Memory	12 GB
OS	Windows 10 Enterprise 64bit
Library	Accord. Neuro, Newtonsoft.Json
Programming Language	C#
IDE	Visual Studio Community 2019

Table 4 Default experiment setup

Parameters	Values
Number of Endorser Peers	6
Number of Orgs	4
Ordering Service	Solo
Endorsement Policy	AND (a, b, c)
Block Frequency (maximum timeout to create a block)	250 ms
Block Size	10 transactions per block
Number of Clients	5
State Database	LevelDB
Programming Language	Node.js
Use of TLS	No
Transaction Duration	60 s
Target Function	eCRF lab data creation

in every subset). In each test round, 75% of the data is used for training, and the rest is used for testing under the given setup.

The following test evaluates each model to find out the best ANN structure. The configuration chosen and the related performance evaluated by Root Mean Square Error (RMSE) are presented in Table 6. The Levenberg–Marquardt algorithm, one of the most successful and quickest approaches for moderately-sized neural networks, supports the training process. We set the maximum number of epochs to 50 for training the ANN model in this test. The best network structure consists of 3 inputs, 10 neurons in the hidden layer, and 1 output. We apply the Sigmoid activation function with a learning rate of 0.1 and the Levenberg–Marquardt algorithm for learning.

Table 5 Dataset used for the learning module

No	Transaction Latency (s)	Send Rate (tps)	Transaction Throughput (tps)	Error (tps)
1	0.22	146.5	145.8	0.7
2	0.22	146.9	146.3	0.6
3	0.19	151.3	150.7	0.6
4	0.18	152.1	151.4	0.7
5	0.19	151.1	150.3	0.8
6	0.17	161.9	161.1	0.8
7	0.15	158	157.3	0.7
8	0.16	157.6	156.9	0.7
9	0.19	161.4	160.5	0.9
10	0.13	160.6	159.8	0.8
…	…	…	…	…
…	…	…	…	…
10080	0.88	161.1	159.7	1.4

5 Performance evaluation of the ANN-based learning module

5.1 Performance metrics of the learning module

We perform a quantified comparative analysis by applying a variety of statistical indicators to describe performance outcomes using a single statistical value. For performance comparisons, three statistical measures are utilized, and the formulae for these measures are as follows:

The average variance discovered in projected values from actual values is computed using mean absolute deviation (MAD). In Eq. (6), MAD is computed by dividing the sum of absolute errors by the actual transaction throughput Actual_i and the predicted transaction throughput Predicted_i with the whole count of items, i.e., n, using the Kalman filter.

$$MAD = \frac{\sum_{n}^{i=1} |Actual_i - Predicted_i|}{n} \tag{6}$$

The mean squared error (MSE) is one of the most commonly used statistical metrics to evaluate prediction systems. When the error magnitude is squared, it eliminates negative and positive error issues while raising the penalty for greater mispredictions relative to low errors. Equation (7) is used to compute the MSE.

$$MSE = \frac{\sum_{n}^{i=1} (Actual_i - Predicted_i)^2}{n} \tag{7}$$

MSE can amplify the actual error, making it challenging to realize and interpret the actual mistake amount. As a result, RMSE is used to solve this problem simply by calculating the square root of MSE. The RMSE is computed by Eq. (8).

$$RMSE = \sqrt{\frac{\sum_{n}^{i=1} (Actual_i - Predicted_i)^2}{n}} \tag{8}$$

5.2 Performance evaluation of the learning module

Table 7 compares the Kalman filter with and without the learning module to validate the efficiency of the proposed approach. The results of tests performed without the learning module are summarized using a variety of R values. Similarly, various F, which represent error factor values, are used to summarize the statistical results of the Kalman filter with the ANN-based learning module. The Kalman filter with the learning model has an error factor F = 0.02, superior to all other alternatives on all statistical metrics. The Kalman filter has the best performance of 2.697 in RMSE at

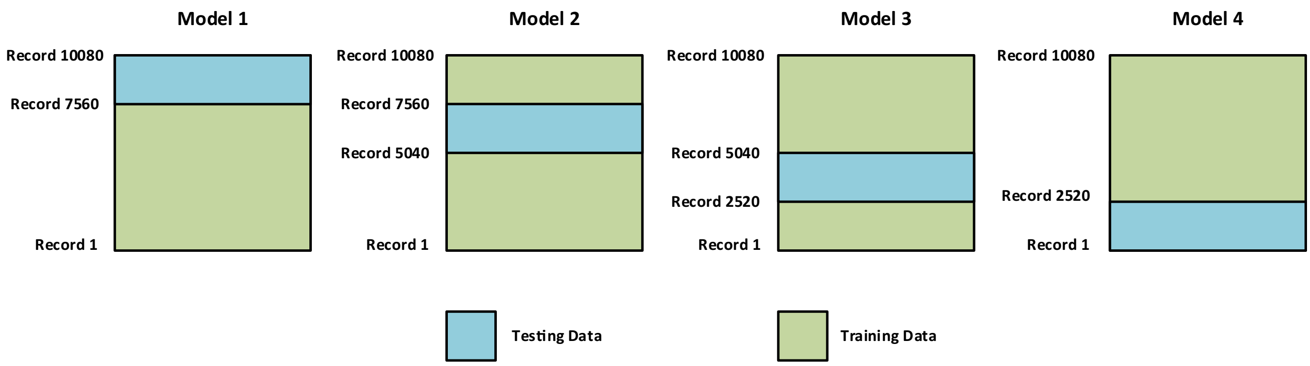


Fig. 8 The fourfold cross-validation model used for the experiment

R = 20 without the learning module. A similar RMSE value of 2.589 is achieved using the Kalman filter in conjunction with the learning module. The best Kalman filter results over one day, with and without the learning module, are shown in Fig. 9. Compared to the Kalman filter's best and worst cases without the learning module, the proposed model (best case) improves the prediction accuracy by 5.41% and 12.29%, respectively.

Table 8 presents the experiment results of the Kalman filter with other learning modules to verify the efficiency of the proposed approach. In this experiment, we compare the Kalman filter using the ANN learning module with two other methods, CNN and CNN-LSTM. These two models are configured using the same number of layers as the ANN model. When combined with a learning model, the Kalman filter achieves an error factor of F = 0.02, making it statistically superior to all other options. With the CNN learning module, the best RMSE value achieved by the Kalman filter is 2.823. In a similar vein, the Kalman filter achieves a best-case RMSE of 2.712 when used with the CNN-LSTM learning module. Compared to these two learning modules, the proposed model improves the prediction accuracy by 9.04% and 4.75%, respectively.

We perform another test to evaluate the impact of hidden layer numbers on the ANN model. In this experiment, we reuse the trained ANN model and extend the scale of the model by adding more hidden layers, each with 10 neurons. The configuration chosen and the related performance evaluated by RMSE are presented in Table 9. The maximum number of epochs is 50 for training the ANN model. The results show that the ANN model with a single hidden layer performs best, with an RMSE value of 0.48. The average error for the model with 2 and 3 hidden layers is 0.75 and 0.91, respectively. Compared to these two extended modules, the proposed model improves the prediction accuracy by 36% and 47.3%, respectively.

Table 6 Parameter configuration for the ANN

Number of Neurons in Hidden Layer	Activation Function	Learning Rate	Experiment ID	Average (Test Cases)	Experiment Average (Test Cases)
5	Sigmoid	0.1	1	0.69	0.50
5	Sigmoid	0.1	2	0.48	
5	Sigmoid	0.1	3	0.31	
5	Sigmoid	0.1	4	0.53	
5	Sigmoid	0.1	1	0.51	0.61
5	Sigmoid	0.1	2	0.72	
5	Sigmoid	0.1	3	0.53	
5	Sigmoid	0.1	4	0.68	
10	Sigmoid	0.1	1	0.38	0.48
10	Sigmoid	0.1	2	0.58	
10	Sigmoid	0.1	3	0.50	
10	Sigmoid	0.1	4	0.45	
15	Sigmoid	0.1	1	0.64	0.67
15	Sigmoid	0.1	2	0.64	
15	Sigmoid	0.1	3	0.74	
15	Sigmoid	0.1	4	0.64	

Table 7 Statistical analysis of the prediction results

Metric	Original Data	Conventional Kalman filter				Improved Kalman filter with Learning Module				
		R = 5	R = 10	R = 15	R = 20	F = 0.005	F = 0.008	F = 0.01	F = 0.02	F = 0.05
MAD	0.565	0.188	0.176	0.173	0.167	0.163	0.167	0.166	0.150	0.155
MSE	20.254	8.224	7.388	7.284	7.274	6.914	6.807	6.770	6.701	6.844
RMSE	4.500	2.868	2.718	2.798	2.697	2.629	2.609	2.602	2.589	2.616

5.3 Performance evaluation with other blockchain networks

This section validates the university of the proposed approach with some other blockchain platforms. We utilize the Hyperledger Caliper to measure the performance of Hyperledger Besu, Hyperledger Burrow, and Hyperledger Sawtooth, respectively. A separate dataset is generated for each test according to the methodology mentioned in Section 4. The parameter configuration of the ANN is set according to the best-case results in Section 5. The proposed approach is further tested with other blockchain networks, and a statistical summary of the experimental results is presented in Table 10. Prediction accuracy for Hyperledger Besu is 2.689 in RMSE when using the Kalman filter with the learning model and an error factor F = 0.02. Without the learning module, the Kalman filter achieves the best RMSE of 2.897 at R = 20 for its predictions. The Kalman filter coupled with the learning model yields an error factor F = 0.02, which yields an RMSE of 2.889 for Hyperledger Burrow's prediction accuracy. The Kalman filter's best prediction result without the learning module is 3.107 in RMSE with R = 20, which is the best result with the learning module enabled. The RMSE prediction accuracy for Hyperledger Sawtooth is 2.979 when using the Kalman filter with the learning model, which yields an error factor F = 0.02. The best prediction result for the Kalman filter without the learning module is 3.212 in RMSE with R = 20.

Figures 10, 11 and 12 presents the best scenarios for different blockchain networks using the Kalman filter with and without the ANN-based learning module over one day. Compared to the Kalman filter's best and worst case results without the learning module, the learning to prediction model (best case) enhanced the prediction accuracy by 7.18% and 9.40%, 7.02%, and 12.40%, 7.25%, and 12.33%, respectively. The experiment results prove that the proposed model has trusty stability, reliability, and good universality with different blockchain networks.

6 Discussion and future research directions

This paper provides a novel learning to prediction approach for evaluating the transaction throughput of blockchain systems in clinical trials. All tests are performed on a single-host virtual system, which is one of the study's limitations. In particular, the blockchain network operates on a local network that is not ideal for use in a production environment. This implies that neither transactions nor block propagation across peers experiences any appreciable network latency. The prototype for assessing the practical use of the designed architecture will be refined to suit the production environment in

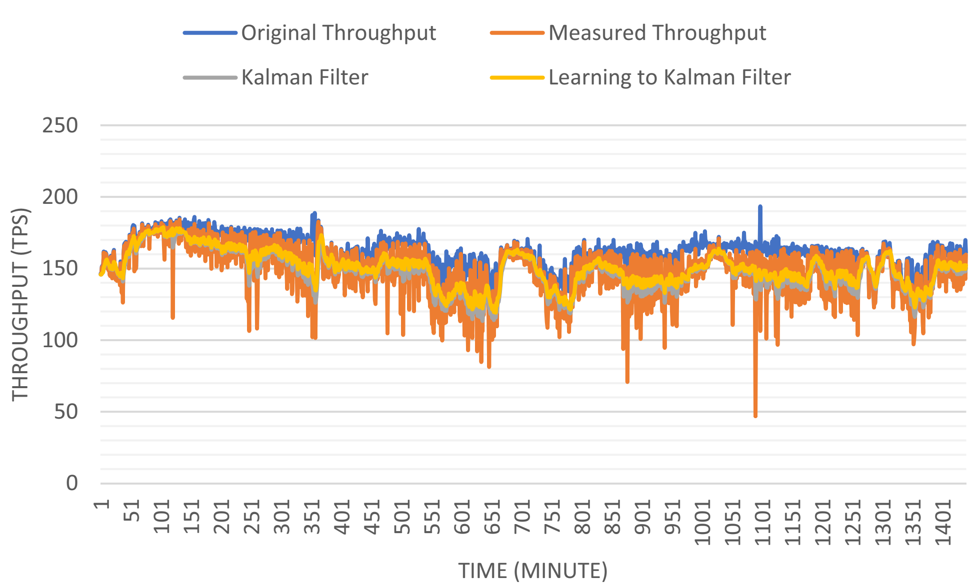


Fig. 9 The Kalman filter's best case results with and without the learning module (one day)

Table 8 Comparative analysis of the proposed approach with other prediction methods

Metric	Original Data	Kalman filter with ANN Learning Module				
		F = 0.005	F = 0.008	F = 0.01	F = 0.02	F = 0.05
MAD	0.565	0.163	0.167	0.166	0.150	0.155
MSE	20.254	6.914	6.807	6.770	6.701	6.844
RMSE	4.500	2.629	2.609	2.602	2.589	2.616
		Kalman filter with CNN Learning Module				
		F = 0.005	F = 0.008	F = 0.01	F = 0.02	F = 0.05
MAD	0.565	0.193	0.185	0.183	0.175	0.173
MSE	20.254	7.321	7.278	7.112	7.012	7.245
RMSE	4.500	2.967	2.923	2.836	2.823	2.912
		Kalman filter with CNN-LSTM Learning Module				
		F = 0.005	F = 0.008	F = 0.01	F = 0.02	F = 0.05
MAD	0.565	0.185	0.176	0.172	0.164	0.169
MSE	20.254	7.134	7.036	6.923	6.878	7.142
RMSE	4.500	2.856	2.845	2.798	2.712	2.851

Table 9 Parameter configuration for the hidden layer number impact on ANN

Hidden Layer Number	Activation Function	Learning Rate	Experiment ID	Average (Test Cases)	Experiment Average (Test Cases)
1	Sigmoid	0.1	1	0.38	0.48
1	Sigmoid	0.1	2	0.58	
1	Sigmoid	0.1	3	0.50	
1	Sigmoid	0.1	4	0.45	
2	Sigmoid	0.1	1	0.62	0.75
2	Sigmoid	0.1	2	0.86	
2	Sigmoid	0.1	3	0.56	
2	Sigmoid	0.1	4	0.95	
3	Sigmoid	0.1	1	0.83	0.91
3	Sigmoid	0.1	2	0.87	
3	Sigmoid	0.1	3	0.91	
3	Sigmoid	0.1	4	1.01	

Table 10 Statistical analysis of the prediction results with different blockchain networks

Name	Metric	Original Data	Kalman filter				Kalman filter with Learning Module				
			R = 5	R = 10	R = 15	R = 20	F = 0.005	F = 0.008	F = 0.01	F = 0.02	F = 0.05
Hyperledger Besu	MAD	0.767	0.288	0.276	0.273	0.267	0.263	0.267	0.266	0.250	0.255
	MSE	22.254	9.224	8.388	8.284	8.274	7.914	7.807	7.770	7.701	7.844
	RMSE	4.717	2.968	2.918	2.898	2.897	2.729	2.809	2.802	2.689	2.716
Hyperledger Burrow	MAD	0.665	0.285	0.275	0.274	0.263	0.261	0.264	0.265	0.251	0.255
	MSE	23.365	9.344	8.345	8.274	8.224	7.804	7.762	7.520	7.401	7.868
	RMSE	4.833	3.168	3.118	3.298	3.107	3.229	2.989	2.912	2.889	2.845
Hyperledger Sawtooth	MAD	0.865	0.294	0.296	0.293	0.297	0.293	0.287	0.289	0.262	0.265
	MSE	24.351	8.224	7.388	7.284	7.274	6.914	6.807	6.770	6.701	6.844
	RMSE	4.935	3.368	3.318	3.398	3.212	2.929	2.958	2.944	2.979	2.926

Fig. 10 The Kalman filter's best case results with and without the learning module (one day) for Hyperledger Besu

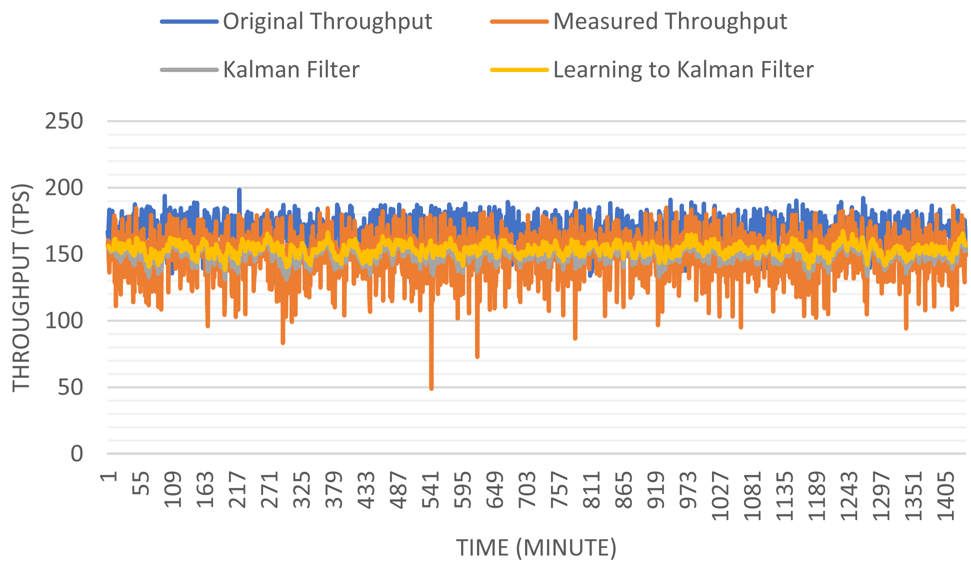


Fig. 11 The Kalman filter's best case results with and without the learning module (one day) for Hyperledger Burrow

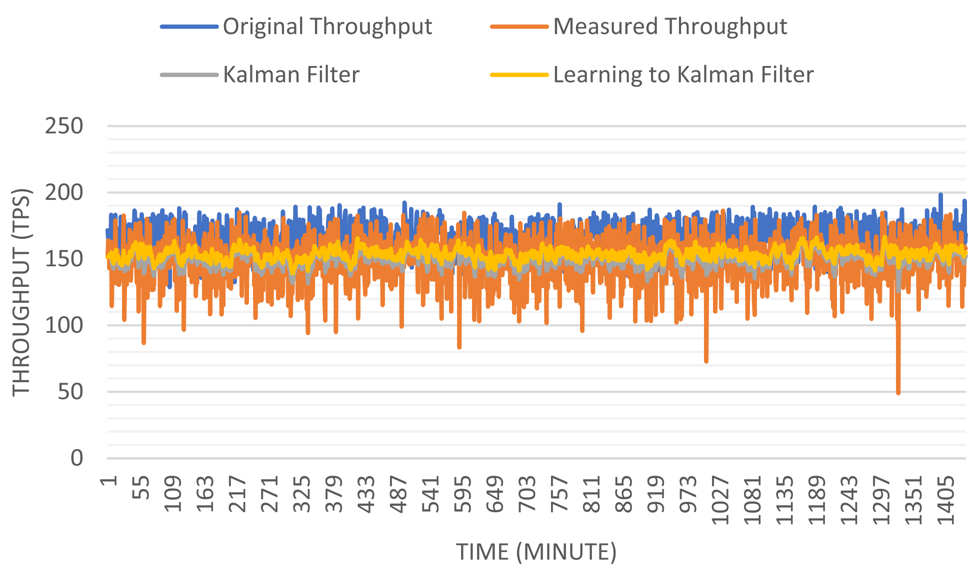
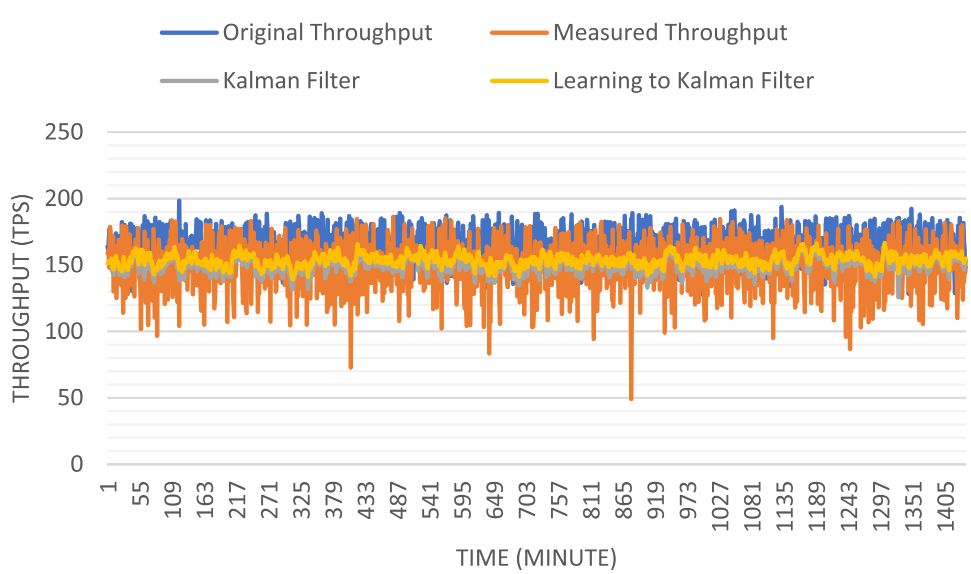


Fig. 12 The Kalman filter's best case results with and without the learning module (one day) for Hyperledger Sawtooth



further work. To ease the setting and management of the clinical trial testbed while ignoring the underlying technology, the blockchain infrastructure will be constructed as a customizable blockchain-as-a-service (BaaS) utilizing a cloud provider like Amazon Web Services (AWS) or IBM Blockchain.

Furthermore, developing a model that could make these predictions with lower execution time and better accuracy would be prudent. The idea of predicting transaction throughput for blockchain systems is relatively new and should be explored further to utilize different algorithms for prediction. While the Kalman filter with ANN, as observed, has provided quite good results, it would be interesting to find out which of these prediction algorithms would be more accurate for blockchain transaction throughput predictions. In future work, the research will be extended in the following directions (1) a more comprehensive experimental analysis with a larger dataset will be conducted to further demonstrate the validity of the proposed learning to prediction approach. (2) deep learning algorithms instead of ANN will be used to tune the performance of other prediction algorithms for transaction throughput prediction.

7 Conclusion

Blockchain infrastructures offer a trustless system for peer-to-peer business network collaboration and trust building. Popular blockchain technologies like Bitcoin and Ethereum have been widely used in various industries. However, few studies intend to predict blockchain behaviors such as transaction throughput, transaction latency, and resource usage.

This study presents a learning to prediction approach to improve the algorithm's performance for transaction throughput prediction. The proposed method comprises two main components: a prediction module and a learning module. The Kalman filter is utilized in the prediction module, while the ANN is used in the learning module. The fundamental problem with the traditional Kalman filter is that parameter values are usually chosen based on the application requirement. Once the parameter values for a particular problem are chosen, they are kept throughout the process. To solve this problem, a learning module is designed and added to the traditional Kalman filter to increase prediction accuracy. A clinical trial testbed built on the permissioned blockchain Hyperledger Fabric is used for the experimental analysis. The experiment findings show that the proposed approach can significantly enhance prediction accuracy in different performance measures.

Moreover, we perform a comparative analysis of the ANN learning module with other learning methods. The proposed model with ANN improves the prediction accuracy by 9.04% and 4.75%, compared to CNN and CNN-LSTM learning modules. Furthermore, we test the proposed approach with other blockchain platforms. The results show that the proposed model has trusty stability, reliability, and good universality. In future work, we will further investigate the application of the proposed learning to the prediction model to improve other algorithms' performance in transaction throughput prediction with a larger dataset and use deep learning algorithms instead of ANN.

Acknowledgements This research was supported by Shanghai Chenguang Plan (under grant number 21CGB08).

Authors' contributions Conceptualization, C.C. and L.H.; Data analysis, L.H. and C.C.; writing—original draft preparation, C.C.; Methodology, L.H. and C.C. and I.U.; writing—review and editing, I.U. and C.C.; supervision, C.C. and J.Y. All authors have read and agreed to the published version of the manuscript.

Data Availability The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Ethical approval and Consent to participate Not applicable.

Human and animal ethics Not applicable.

Consent for publication All authors gave their consent for publication.

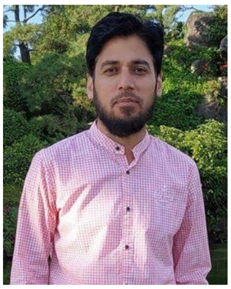
Competing interests The authors declare that they have no competing interests.

References

1. Maull R et al (2017) Distributed ledger technology: Applications and implications. Strateg Chang 26(5):481–489
2. Hang L, Kim D-H (2019) Design and implementation of an integrated IoT blockchain platform for sensing data integrity. Sensors 19:2228
3. Hang L, Kim D-H (2019) SLA-Based sharing economy service with smart contract for resource integrity in the internet of things. Appl Sci 9:3602
4. Hang L, Chen C, Zhang L, Yang J (2022) Blockchain for applications of clinical trials: Taxonomy, challenges, and future directions. IET Communications
5. Hang L, Choi E, Kim D-H (2019) A Novel EMR integrity management based on a medical blockchain platform in hospital. Electronics 8:467
6. Hang L, Kim BH, Kim KH, Kim DH (2021) A Permissioned Blockchain-Based Clinical Trial Service Platform to Improve Trial Data Transparency. BioMed Res Int 2021:22. https://doi.org/10.1155/2021/5554487. Article ID 5554487
7. Xu R, Hang L, Jin W, Kim D (2021) Distributed Secure Edge Computing Architecture Based on Blockchain for Real-Time Data Integrity in IoT Environments. Actuators 10:197. https://doi.org/10.3390/act10080197
8. Zhang L, Hang L, Jin W, Kim D (2021) Interoperable Multi-Blockchain Platform Based on Integrated REST APIs for Reliable Tourism Management. Electronics 10:2990. https://doi.org/10.3390/electronics10232990

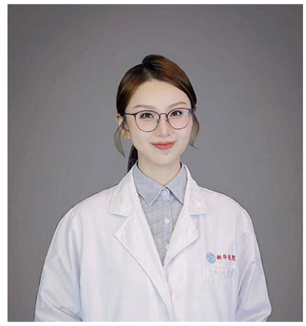
9. Fosso Wamba S et al (2020) Bitcoin, Blockchain and Fintech: a systematic review and case studies in the supply chain. Prod Plan Control 31(2–3):115–142
10. Underwood S (2016) Blockchain beyond bitcoin. Commun ACM 59(11):15–17
11. Hang L, Kim BH, Kim DH (2022) A Transaction Traffic Control Approach Based on Fuzzy Logic to Improve Hyperledger Fabric Performance. Wirel Commun Mob Comput 2022:19. https://doi.org/10.1155/2022/2032165. Article ID 2032165
12. Golosova J, Romanovs A (2018) The advantages and disadvantages of the blockchain technology. In 2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE). IEEE, pp 1–6
13. Weigend AS (2018) Time Series Prediction: Forecasting the Future and Understanding the Past. Routledge, Abington, UK
14. Omar IA, Jayaraman R, Salah K, Simsekler MCE, Yaqoob I, Ellahham S (2020) Ensuring protocol compliance and data transparency in clinical trials using Blockchain smart contracts. BMC Med Res Methodol 20(1):1–17
15. Omar IA, Jayaraman R, Salah K, Yaqoob I, Ellahham S (2021) Applications of blockchain technology in clinical trials: review and open challenges. Arab J Sci Eng 46(4):3001–3015
16. Brunnert A, van Hoorn A, Willnecker F, Danciu A, Hasselbring W, Heger C, Herbst N, Jamshidi P, Jung R, von Kistowski J et al (2015) Performance-oriented devops: A research agenda, arXiv preprint arXiv:1508.04752
17. Madan I, Saluja S, Zhao A (2015) Automated bitcoin trading via machine learning algorithms, p 20. http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf. Accessed 13 Dec 2022
18. McNally S, Roche J, Caton S (2018) Predicting the price of bitcoin using machine learning. In 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP). IEEE, pp 339–343
19. Jang H, Lee J (2017) An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. Ieee Access 6:5427–5437
20. Poongodi M et al (2020) Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. Comput Electr Eng 81:106527
21. Metawa N, Alghamdi MI, El-Hasnony IM, Elhoseny M (2021) Return Rate Prediction in Blockchain Financial Products Using Deep Learning. Sustainability 13:11901. https://doi.org/10.3390/su132111901
22. Yuksel E, Wilson JN, Gader PD (2012) Twenty years of mixture of experts. IEEE Trans. Neural Netw. Learn. Syst., vol. 23,no. 8, pp. 1177–1193. [Online]. Available: https://ieeexplore.ieee.org/document/6215056/
23. Wolpert DH (1992) 'Stacked generalization.' Neural Netw 5(2):241–259
24. Androulaki E et al (2018) Hyperledger fabric: a distributed operating system for permissioned blockchains. Proceedings of the thirteenth EuroSys conference
25. Odry Á et al (2018) Kalman filter for mobile-robot attitude estimation: Novel optimized and adaptive solutions. Mech Syst Signal Process 110:569–589
26. Havlík J, Straka O (2015) 'Performance evaluation of iterated extended Kalman filter with variable step-length. J Phys Conf Ser 659:012022
27. Huang J, McBratney AB, Minasny B, Triantafilis J (2017) Monitoring and modelling soil water dynamics using electromagnetic conductivity imaging and the ensemble Kalman filter. Geoderma 285:76–93
28. Połap D, Winnicka A, Serwata K, K ˛esik K, Wo´zniak M (2018) An Intelligent System for Monitoring Skin Diseases. Sensors 18:2552
29. Rong H, Peng C, Chen Y, Zou L, Zhu Y, Lv J (2018) Adaptive-Gain Regulation of Extended Kalman Filter for Use in Inertial and Magnetic Units Based on Hidden Markov Model. IEEE Sens J 18:3016–3027
30. Zhao S, Shmaliy YS, Shi P, Ahn CK (2017) Fusion Kalman/UFIR filter for state estimation with uncertain parameters and noise statistics. IEEE Trans Ind Electron 64:3075–3083
31. Yates D, Gangopadhyay S, Rajagopalan B, Strzepek K (2003) 'A technique for generating regional climate scenarios using a nearestneighbor algorithm.' Water Resour Res 39(7):1–15
32. Zhang M-L, Zhou Z-H (2005) A K-nearest neighbor based algorithm for multi-label classification, in Proc. IEEE Int Conf Granular Comput. 718–721
33. Suthaharan S (2016) 'Machine learning models and algorithms for big data classification.' Integr Ser Inf Syst 36:1–12
34 Abiodun OI et al (2018) State-of-the-art in artificial neural network applications: A survey. Heliyon 4(11):e00938
35. Wu Y-C, Feng J-W (2018) Development and application of artificial neural network. Wireless Pers Commun 102(2):1645–1656
36. Kang CW, Park CG (2009) Attitude estimation with accelerometers and gyros using fuzzy tuned Kalman filter. In 2009 European Control Conference (ECC). IEEE, pp 3713–3718
37. Ibarra-Bonilla MN, Escamilla-Ambrosio PJ, Ramirez-Cortes JM (2015) Attitude estimation using a Neuro-Fuzzy tuning based adaptive Kalman filter. J Intell Fuzzy Syst 29:479–488
38. Singh HJ, Senhaji Hafid A (2019) Prediction of transaction confirmation time in ethereum blockchain using machine learning. International Congress on Blockchain and Applications. Springer, Cham
39. Yasaweerasinghelage R, Staples M, Weber I (2017) Predicting latency of blockchain-based systems using architectural modelling and simulation. 2017 IEEE International Conference on Software Architecture (ICSA). IEEE
40. Zhang H, Jin C, Cui H (2018) A Method to Predict the Performance and Storage of Executing Contract for Ethereum Consortium-Blockchain. In: Chen, S., Wang, H., Zhang, LJ. (eds) Blockchain – ICBC 2018. ICBC 2018. Lecture Notes in Computer Science(), vol 10974. Springer, Cham. https://doi.org/10.1007/978-3-319-94478-4_5
41. Balsamo S, Marin A, Mitrani I, Rebagliati N (2021) Prediction of the consolidation delay in blockchain-based applications. In Proceedings of the ACM/SPEC. International Conference on Performance Engineering, pp 81–92
42. Hyperledger Caliper, Available online: https://www.hyperledger.org/projects/caliper. Accessed 10 Oct 2022

Lei Hang received the B.S. degree in computer engineering in 2015 and the M.S. degree from the Jeju National University, Korea, in 2017. He received the Ph.D degree at Jeju National University, South Korea. His area of interest includes wireless sensor networks, M2M/IOT, mobile computing, and blockchain.



Israr Ullah received the M.S. degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2009, and the Ph.D. degree in computer engineering from Jeju National University, Jeju, South Korea, in 2019. His research work is focused on the application of prediction and optimization algorithms to build IoT-based solutions. His research interests mainly include analytical modeling, network simulation, and analysis of optimization algorithms.

Jun Yang doctoral supervisor, professor, director of Otolaryngology Head and neck surgery, Xinhua Hospital, Medical School of Shanghai Jiao Tong University. The specialty is ear microscopy, ear nerve lateral skull surgery, and is good at cochlear implantation, diagnosis and treatment of vertigo disease, hearing impairment disease, facial nerve disease, acoustic neuroma, jugular foramen tumor, benign and malignant tumor of temporal bone. His research work is focused on auditory development and cochlear transduction mechanism.



Chun Chen was graduated from Medical School of Shanghai Jiao tong University in 2019 and worked in Xinhua Hospital affiliated to Shanghai Jiao Tong University. She presided independently and be awarded a city level fund with regards to the molecular mechanisms and clinical features of high incidence tumor in Shanghai. She had achieved 5 core journal manuscripts and 5 SCI manuscripts. Her research work is focused Malignant tumor comprehensive therapy, clinical data analysis and retrospective analysis.