Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.031088

ARTICLE

Privacy Enhanced Mobile User Authentication Method Using Motion Sensors

Chunlin Xiong[1,2], Zhengqiu Weng[3,4,*], Jia Liu[1], Liang Gu[2], Fayez Alqahtani[5], Amr Gafar[6] and Pradip Kumar Sharma[7]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518052, China

[2]Sangfor Technologies Inc., Shenzhen, 518055, China

[3]School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou, 325035, China

[4]College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, 310023, China

[5]Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh, 12372, Saudi Arabia

[6]Math & Computer Science Department, Faculty of Science, Menofia University, Shebin El-Kom, Egypt

[7]Computing Science Department, University of Aberdeen, Aberdeen, UK

*Corresponding Author: Zhengqiu Weng. Email: derisweng@163.com

Received: 13 May 2023 Accepted: 10 August 2023 Published: 15 December 2023

ABSTRACT

With the development of hardware devices and the upgrading of smartphones, a large number of users save privacy-related information in mobile devices, mainly smartphones, which puts forward higher demands on the protection of mobile users' privacy information. At present, mobile user authentication methods based on human-computer interaction have been extensively studied due to their advantages of high precision and non-perception, but there are still shortcomings such as low data collection efficiency, untrustworthy participating nodes, and lack of practicability. To this end, this paper proposes a privacy-enhanced mobile user authentication method with motion sensors, which mainly includes: (1) Construct a smart contract-based private chain and federated learning to improve the data collection efficiency of mobile user authentication, reduce the probability of the model being bypassed by attackers, and reduce the overhead of data centralized processing and the risk of privacy leakage; (2) Use certificateless encryption to realize the authentication of the device to ensure the credibility of the client nodes participating in the calculation; (3) Combine Variational Mode Decomposition (VMD) and Long Short-Term Memory (LSTM) to analyze and model the motion sensor data of mobile devices to improve the accuracy of model certification. The experimental results on the real environment dataset of 1513 people show that the method proposed in this paper can effectively resist poisoning attacks while ensuring the accuracy and efficiency of mobile user authentication.

KEYWORDS

Mobile authentication; blockchain; federated learning; smart contract; certificateless encryption; VMD; LSTM

1 Introduction

With the rapid development of mobile communication technology and the upgrading of hardware, mobile devices (especially mobile phones) have been rapidly popularized and widely used. According

to the forecast of CCS Insight (a global authoritative market research organization), by 2022, 840 million 5G-enabled mobile phones will be delivered to the mobile market, accounting for 42% of the total global shipments [1]. Nowadays, more and more private information is collected for the intelligentization of mobile devices. In order to prevent malicious attackers from illegally accessing private information stored on mobile devices, it is urgently necessary to analyze the software and hardware characteristics of mobile devices and their applicable application scenarios and design a suitable and robust authentication mode to protect the user's information.

Methods based on credentials such as text passwords and PIN codes are widely used for user authentication, but such traditional authentication mechanisms have inherent security problems and expose weaknesses that are vulnerable to various traditional attacks, such as brute-force cracking [2], smudge attack [3], shoulder peep attack [4], social engineering attack [5], etc. In addition, since mobile device interfaces are usually small, the memorization and input of complex passwords can cause additional headaches for users. Compared with the above authentication mechanisms, the authentication method based on the user's static characteristics and dynamic behavior can achieve higher authentication accuracy. It usually uses learning methods to extract information representing user characteristics from different types of sensors on mobile devices to build a model. Such as face [6,7], fingerprint [8,9], voice [10,11], environmental location [12,13], keystroke behavior [14,15], finger movement [16,17], etc. However, the above-mentioned biometric information collection usually requires invoking the privacy-related permissions of the mobile device, which makes users worry that their privacy-related information may be leaked, and cannot take into account the security, privacy, and usability requirements of mobile authentication jointly at the same time.

For ideal mobile device user authentication, it should be able to have privacy, security and availability at any time and any place. The user authentication requirements to be solved include: (1) Privacy protection. User privacy-related data cannot be called too frequently to avoid user concerns about privacy leakage. (2) For multi-application mobile devices, a general service for device-level detection needs to be established. Common services allow reuse of detection results and redundancy removal from a multi-application data perspective. (3) Services can be provided to any user in any state. The data training and verification process does not depend on the user's motion state or device location, thus forming continuous protection for the user. (4) Automatic training in noisy environments. It is unrealistic to assume that all training samples have labels, and it is virtually unknown whether training samples were collected while the device was being operated by an authorized owner. According to the above requirements, user authentication methods based on mobile device motion sensors (including accelerometers, gyroscopes and gravity sensors) have been proposed by scholars [18–28]. On smartphones, the invocation of motion sensors does not require privacy-related permissions, and any application can obtain relevant data through the interface of the system layer. During user authentication, to obtain a large amount of training data based on motion sensors, existing machine learning methods [18–28] all require that the private data representing authenticated users be aggregated on a central server for model training and analysis. This not only leads to higher communication and storage costs, but users also face serious privacy risks. To protect the user's private information as much as possible without affecting the model training results, federated learning was proposed by researchers [29]. In federated learning, each device iteratively trains a local model (including information such as model weights and gradients), and the respective raw data does not leave the local device. Usually, federated learning can be implemented with the help of a central requester, and the device side uses local data to train and improve the global model issued by the center. After each local training, the device sends local model updates to the center. By aggregating these local model updates, the center generates a new global model for the next iteration. The above process is

repeated for both the mobile device and the center until the global model converges [30]. However, in real tasks (i.e., security-related user authentication), the above methods have the following drawbacks: First, the data collection efficiency is low and there is no effective incentive mechanism. Most of the data collection steps in the existing machine learning tasks have been completed in advance, and the number of participants in the training is limited, and participants are required to collect new data under each behavior. These methods are very wasteful of human and material resources and have little regard for the efficiency of data collection in real scenarios, and the data collection rate largely depends on how often participants use mobile devices. Second, the traditional data collection method requires participants to collect data under each behavior and manage it centrally, but in the federated learning mode, it is impossible to determine whether the data collected by edge mobile devices is authentic and effective, and it may appear untrustworthy. There may be cases where untrusted nodes use fake data to train a model and return wrong gradients to achieve a poisoning attack on the model.

To this end, this paper proposes a privacy protection-oriented mobile user authentication method, which combines the smart contract-based private chain, federated learning, and certificateless encryption to ensure data privacy and node trustworthiness, and adopts VMD and LSTM network to improve the accuracy of model verification. The main contributions of this paper are as follows:

1. To prevent unreliable model updates, this paper uses the training quality proof based on the consensus mechanism to validate the model to effectively resist model poisoning attacks.
2. To improve the data collection rate, this paper uses a private chain based on smart contracts to send certain rewards to participants corresponding to each node that participates in training and helps optimize the model, to motivate nodes to collect data.
3. To ensure the safety and reliability of participating computing devices, this paper uses certificateless encryption to implement device authentication to ensure the credibility of client nodes participating in computing.
4. To improve the automation level and accuracy of mobile user authentication, this paper proposes a mobile device user authentication model based on LSTM and uses VMD for noise removal and signal reconstruction, which can make up for the lack of coverage and accuracy when modeling with traditional statistical features.

The remainder of this article is organized as follows: Section 2 describes the methodology of our work. Section 3 presents the overall evaluation of our method. Section 4 surveys the relevant work and Section 5 concludes our work.

2 Methodology

2.1 Overview

The method proposed in this paper consists of the following three steps, as shown in Fig. 1. To collect representative and content-rich data, the motion sensor built into the mobile device will trigger the collection of data that best represents the user's human-computer interaction characteristics according to actual needs. The requester Req will form a smart contract-based private blockchain with nodes (mobile devices) that are willing to participate in model training. On this basis, communication iterations are performed to train the global model. The requester Req will select the nodes participating in this iteration according to a certain strategy. The selected node will perform a series of operations in a predetermined order, including sensor data collection and storage, node trusted authentication, data preprocessing, global model download, local model training, etc., and finally upload the local model parameters after a certain number of rounds of training to Req and implement verification, Req then

packages and broadcasts part of the transaction content, rewards nodes that help the model converge, updates the global model through a predetermined algorithm, and pushes the updated global model to the new candidate in the next round of communication. Each step will be described separately in the next. All models are deployed on mobiles including Req and voluntary nodes.

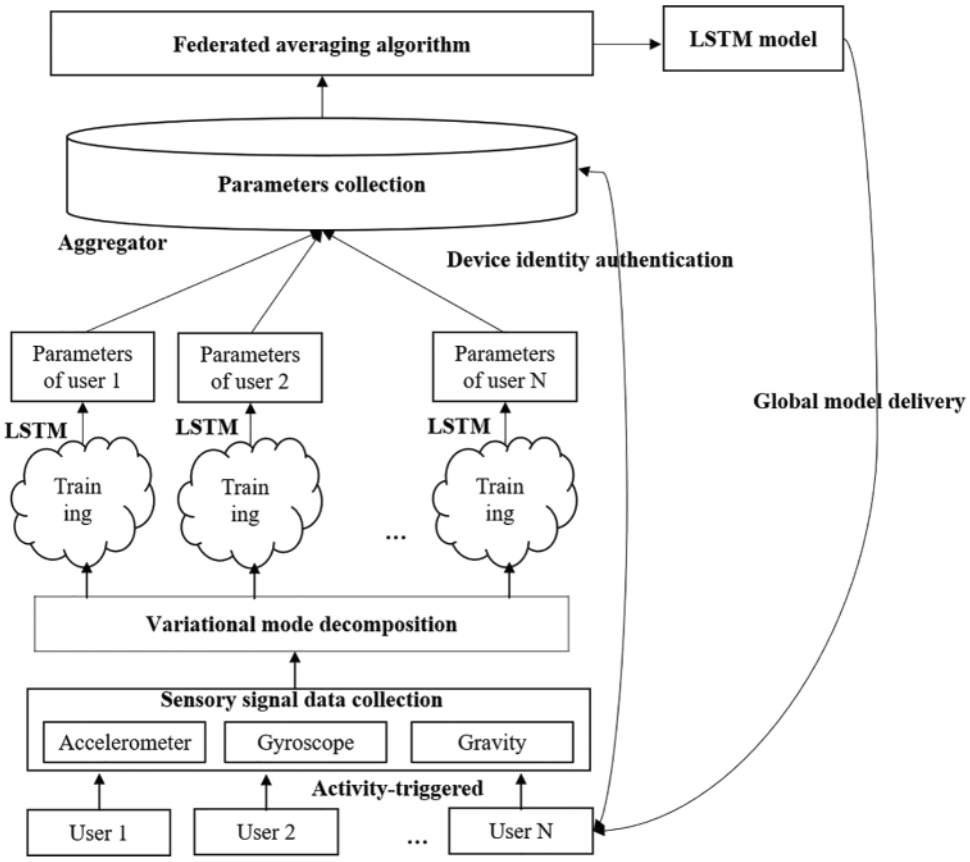


Figure 1: Overview of the architecture of our method. User authentication process based on blockchain and federated learning

2.2 Data Collection

In the process of data collection for mobile user authentication based on motion sensors, the key to subsequent authentication is when and in what state to collect data. This paper continues the data collection method in the famous work [20,21].

(1) Trigger point for data collection. When the following two requirements are met, data collection will be started: first, the screen of the smartphone is on; second, the application in the foreground of the smartphone is switched. The above two conditions indicate that the user is using a smartphone to open a certain application, and the authentication in this case can determine whether the owner himself is operating the phone, which greatly improves the security of the mobile device.

(2) Effective duration of data collection. In the real-world scenario, the typical time period when the user opens the mobile phone application generally lasts 2–4 s. The collection of effective user behaviors is performed by presetting the collection duration t ($2\,s \leq t \leq 4\,s$). If the user uses the mobile

phone for less than t (s) (e.g., the mobile phone screen is turned off during the acquisition process), the acquisition is terminated, and $t = 3$ s is set in this paper.

(3) The sampling frequency of data collection. Since the relevant readings of the three motion sensors, accelerometer, gyroscope, and gravity sensor, need to be obtained from the underlying interface of the mobile device, accessing the data interface at high frequency will generate unnecessary overhead. To save energy, this paper sets the background running service of the smartphone to collect and judge data at different frequencies, which can be divided into idle state and active state. Under normal circumstances (no active trigger), the data sampling frequency in the background is in an idle state, which is 15 Hz. When the two data collection conditions in (1) are satisfied at the same time, the data collection frequency will jump from the idle state to the active state, and the sampling frequency will be 50 Hz.

2.3 Smart Contract-Based Private Chain

To improve the efficiency of data collection, the coverage of the scene, and the authenticity of the data, this paper adopts the incentive and consensus mechanism of the blockchain to address the above problems. The requester Req and N nodes (participating volunteers) together form a smart contract-based private chain. Among them, the smart contract includes the specific requirements of the requester, such as environment configuration, model architecture, memory requirements, reward distribution, etc.; and the specific iterative process of federated learning, such as node selection, model download, local model update and upload, verification, etc. When the requester publishes the smart contract, the participant can choose whether to participate in the contract. After the contract is established, the requester and the participant create a private chain and carry out the follow-up process. At the same time, the contract remains transparent and cannot be unilaterally modified, and the process is automatically executed. In this private chain, Req will send a certain reward to each node that participates in training and helps optimize the model, so as to motivate the node to collect data. Through the consensus trust mechanism of the blockchain, Req can realize the training quality of the node. Therefore, the training effectiveness of the node is ensured and the poisoning attack is blocked. In addition, the private chain is extensible, and whenever a new node wants to join, it can participate in the private chain as long as it is legally registered and authorized. During this period, device authentication without certificate encryption can ensure the trustworthiness of nodes in the interaction process.

In the private chain, if Req wants to obtain a fully trained model, it needs multiple communication iterations. The specific steps of each communication iteration are as follows:

(1) Node selection. Before model training, all nodes in the private chain need to be screened once, and the nodes that help to train the model in this iteration are selected. The first is that the selected node (mobile device) needs to be connected to the power supply and connected to unmetered wifi, so as to ensure that this round of iteration will not affect the normal use of the mobile device. Secondly, for the nodes that meet the above conditions, these nodes are sorted according to the trust degree, and the contract will give priority to the nodes with a high trust degree. Assuming that there are a total of N nodes in this private chain, each iteration will select a sub-node set P for federated learning ($|P| \leq N$).

(2) Model download. For each selected node P_t, the global model m_i provided by Req needs to be downloaded through the network, where $P_t \subset P$, t represents the t-th node in the selected nodes, and i represents the number of communication iterations.

(3) Local model update. For each selected node P_t, after it receives the global model, it needs to preprocess the local data, use the local data to continue training the model, and update the parameters,

until the model converges again after several local iterations, during which the GPU of the node itself and other hardware acceleration devices can be used to reduce the training time. After that, the node uploads the local model, and the local client generates the authentication public key based on the unlicensed encryption and the ID of the device itself (see Section 2.5 for details), as well as the timestamp Timestamp to the requester Req.

(4) Verify. After receiving the local model uploaded by the node P_t, the requester first checks whether the corresponding client is trustworthy (note: this can only prevent attacks initiated by attackers using unknown devices). The model is validated using a consensus-based proof of training quality (PoQ) that uses prediction accuracy to quantify the performance of the trained local model (for client-side poisoning attacks, corresponding to poisoning attacks evaluation will be described in detail in Section 2.5). Specifically, in classification during training, the accuracy is represented by the score of correctly classified records, which is measured by the mean absolute error (MAE):

$$MAE\,(m_i) = \frac{1}{n}\sum_{i=1}^{n} |y_i - f\,(x_i)| \tag{1}$$

where $f(x_i)$ is the predicted value of the model m_i, and y_i is the recorded true value. The lower the MAE of the model m_i, the higher the accuracy of m_i. The requester Req will mark the local model whose MAE is higher than a certain set threshold as an invalid model, otherwise, mark it as a valid model, and reset the trust degree b of the node according to the proportion of the historical valid model of the node. It should be noted that, regardless of the MAE result, Req will record the models and timestamps uploaded by all nodes in this iteration, and as the number of iterations increases, the MAE will dynamically decrease.

(5) Block generation and broadcasting. When all the selected nodes have uploaded the local model or after a certain period of time, the requester Req will execute the incentive mechanism, that is, provide a reward (Ether) to the node that generates a valid local model, and then package a block. The block's size is (h+δND), where h is the block header, including the hash value of the previous block, the timestamp generated by the block, etc. ND represents the number of nodes providing the local model, and δ is an eight Tuple (i.e., communication iteration number i, node identifier id, hash value h of model parameters provided by this node, MAE corresponding to the model provided by this node, identifier m of a valid model, reward p of this node, The trust degree b of the node, the upload timestamp of the current model of the node), after the package is completed, the block is incorporated into the chain, and all nodes are broadcasted to notify.

(6) Global model update. The global model is updated using the FedAvg algorithm of federated learning [31], the specific algorithm is shown in Section 2.4.

For the requester Req, it is only necessary to perform the above 6 steps iteratively until the global model converges or the requirements are met, and the purpose of data collection and model training is finally achieved. Among them, the incentive mechanism of the blockchain ensures the data collection efficiency of nodes (mobile devices), and the consensus mechanism ensures the authenticity of the collected data.

2.4 Distributed Training Based on Federated Learning

Federated learning can not only allow users to obtain relevant training information from the rich sensor data of different devices but also does not need to consume additional central cloud storage. In other words, the user's data will not leave the local mobile device, which can protect the user's privacy to the greatest extent. The training task is solved by a loose federation of participating mobile

devices (i.e., clients) coordinated by a central server. Each client has a local training dataset that is never uploaded to the server, and each client computes an update to the current global model maintained by the server and only communicates this update. In addition, this method is different from the traditional distributed method, in the case that the user data is not independent and identically distributed, high-level model training can still be achieved.

The specific process of federated learning has been described in detail in Section 2.3. In the global model update step, this paper uses the federated learning averaging (FedAvg) algorithm to achieve it, which is as follows:

The federated averaging algorithm integrates multiple deep learning models using stochastic gradient descent into a single global model. Similar to stand-alone machine learning, the goal of federated learning is to minimize empirical risk, i.e.,

$$\min_{x\in R^d}\left[F(x)=\frac{1}{n}\sum_{i=1}^{n}f(x;s_i)\right] \tag{2}$$

Among them, n is the sample size, s_i represents the i-th sample individual, and f(x;si) represents the loss function of the model. Assuming that there are K local models, P_k represents the sequence number set of sample individuals owned by the k-th model. $n_k = |P_k|$, we can rewrite the objective function as:

$$F(x)=\sum_{k=1}^{K}\frac{n_k}{n}F_k(x) \tag{3}$$

$$F_k(x)=\frac{1}{n_k}\sum_{i\in P_k}f(x;s_i) \tag{4}$$

It is worth noting that since the data of each mobile device cannot represent the global data, it cannot be considered $E_{P_k}[F_k(x)]$ the same as f(x), that is, any local model cannot be used as the global model.

In this paper, a parameter update of the local model is called an iteration. Let b represent a batch, then the k-th local model iteration formula is:

$$x_k \leftarrow x_k-\frac{\eta}{|b|}\sum_{i\in b}\nabla f(x_k;s_i) \tag{5}$$

The overall method of distributed training based on federated learning is summarized as follows: The training process is divided into multiple rounds, and $C\times K$ ($0\le C\le 1$) local models are selected to learn the data in each round. The number of epochs in one round for the k-th local model is E, the batch size is B, and thus the number of iterations is E/B. After one round, the parameters of all local models participating in the learning are weighted and averaged to obtain the global model. Notably, blockchain-based authentication and training are performed by the requester and the participant, respectively. The speed of model training depends on the computing power of the participants. During this period, the participants can use the GPU of the node itself and other hardware acceleration devices to reduce the training time. To a certain extent, because in the distributed training based on blockchain, the data used by each participant is less than that in the centralized cloud, so in the case of the same configuration, the training speed of machine learning based on blockchain will increase faster than centralized training in the cloud. The specific architecture and deployment environment of the model will be embedded in the smart contract initiated by the requester. If the participant has a suitable

environment and agrees to participate in the application, the model can be constructed through the specific model architecture.

2.5 Device Authentication Based on Certificateless Encryption

In certificateless cryptography, each client's unique ID can be used to create a public key, and all other users can verify that the public key belongs to this ID. Additionally, if the client revokes the old key, it will create a new public-private key pair using its unique ID. The new key pair is different from the old one, but still generated with a unique ID. Certificateless cryptography is derived from Identity Based Encryption (IBE) to solve the key escrow problem in IBE. In certificateless cryptography, the key generation center creates a partial private key PSK according to the identity of the client, and the client uses the partial private key PSK and its secret value X to establish the private key SK. Since the secret value X only exists in the client, the Key Generation Center (KGC) will not be able to calculate the private key SK, which effectively avoids the key escrow problem in IBE. In addition, the user also creates a public key based on the secret value and discloses it. However, the certificateless encryption scheme only ensures the security and reliability of the participating computing devices but does not guarantee that the mobile device will not be lent to others during the training phase. Therefore, in practical applications, it can be required that users must personally use the mobile device during the training phase.

In specific use, device A issues a transaction signed with its private key SK_A, and attaches its public key PK_A and unique identity ID_A to the transaction. Other devices can check that: 1) this transaction is indeed signed with the private key associated with PK_A, and 2) PK_A belongs to ID_A. In this way, it is easy to verify that the transaction was created by the client device with ID_A. The steps to create keys PK_A and SK_A for client A are as follows:

(1) The algorithm takes the security parameter lambda λ, returns the system parameter K and a secret master key MSK. The algorithm is run by KGC, only KGC knows the value of MSK.

$$Setup(1^{\lambda}) \rightarrow (K, MSK) \tag{6}$$

(2) The partial private key generation algorithm adopts the system parameter K, the identity of client A $ID_A \in \{0,1\}^*$, the master key MSK, and then outputs the partial private key PSK_A. The algorithm is run by KGC and the output PSK_A will be transmitted to client A.

$$PSkeyGen(K, ID_A, MSK) \rightarrow (PSK_A) \tag{7}$$

(3) The secret value generation algorithm adopts the system parameter K and the identity ID_A of the client A, and outputs the secret value X_A. The algorithm is run by the user and X_A will be used to convert part of the private key into a private key. The algorithm is run by the client.

$$SValGen(K, ID_A) \rightarrow (X_A) \tag{8}$$

(4) The private key generation algorithm takes the system parameter K, part of the private key PSK_A and the secret value X_A as input, and returns the private key SK_A. The algorithm is run by the client, and only the client itself has the private key.

$$SKeyGen\,(K, PSK_A, X_A) \;\rightarrow\; (SK_A) \tag{9}$$

(5) The public key generation algorithm constructs the public key PK_A with the system parameter K and the secret value X_A. The algorithm is run by the client and PK_A will be broadcast to the public.

$$PKeyGen\,(K, X_A) \;\rightarrow\; (PK_A) \tag{10}$$

During the device-trusted authentication phase:

(1) Algorithm uses the system parameter K, the message M to be signed, and the client's private key SKA, and outputs a signature.

$$Sign\,(K, \mathrm{M} \in M, SK_A) \;\rightarrow\; Sig \tag{11}$$

(2) Algorithm uses message M, signature Sig, user ID_A and user private key SK_A, and outputs whether the signature is valid, that is, it is judged that the transaction is indeed signed with the private key associated with PK_A.

$$Ver\,(\mathrm{M} \in M, Sig, ID_A, PK_A)\;\; Valid \vee Invalid \tag{12}$$

(3) Judging that PK_A belongs to ID_A and complete the authentication.

$$VerID\,(ID_A, PK_A, K)\;\; Valid \vee Invalid \tag{13}$$

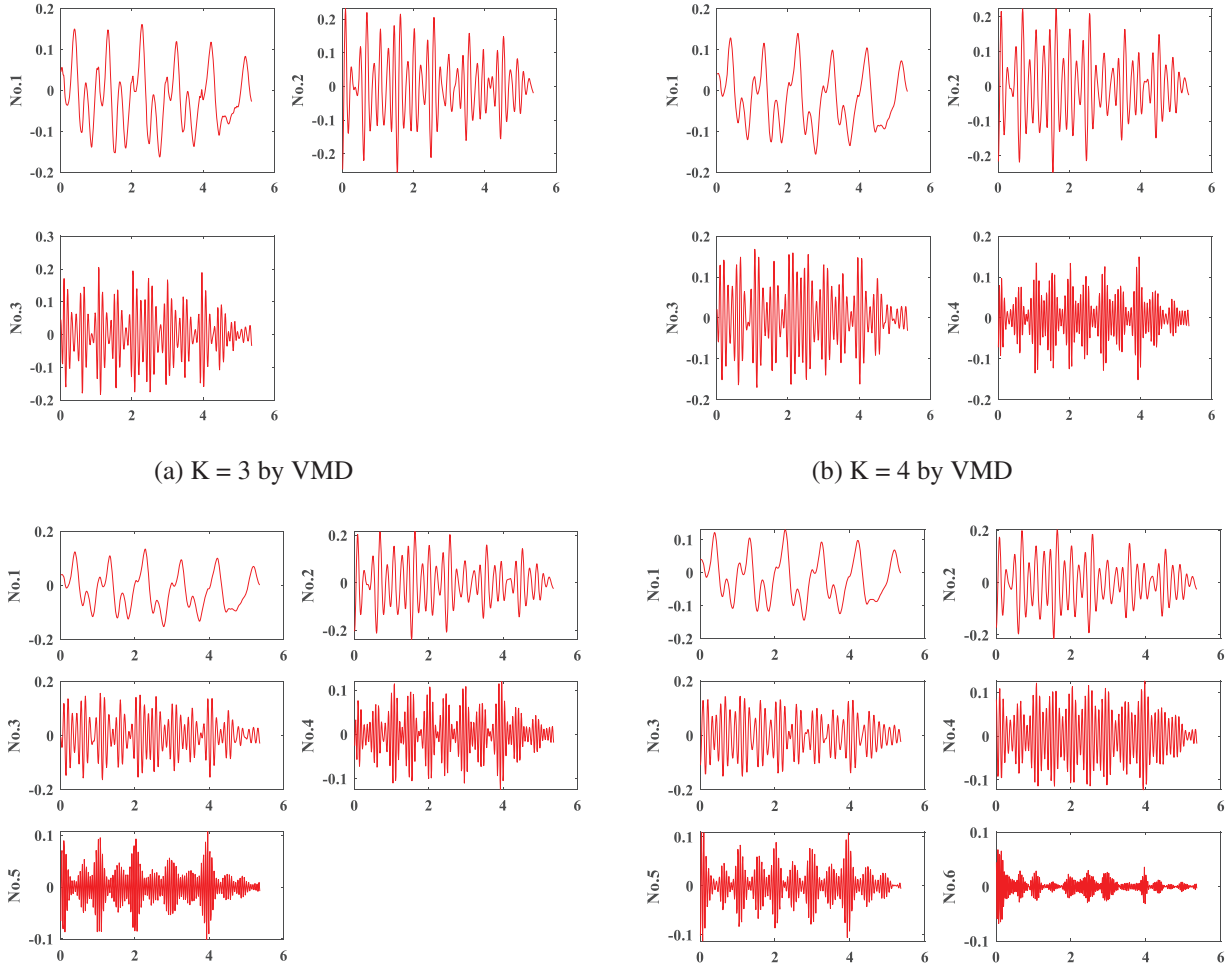
2.6 Model Training with VMD and LSTM

Since the sensor data of mobile devices is time series data, to train a robust model, two main points need to be considered: first, the removal of sequence noise and data enhancement; second, the effective identification of sequence context information.

For the first point, this paper adopts the VMD algorithm for noise filtering and data enhancement. VMD is an adaptive, non-recursive method that can simultaneously analyze non-stationary and nonlinear signals [32]. The essence of the VMD algorithm is the process of solving variational problems. This process includes the construction and solution of variational problems. The variational problem is formulated as: decompose the original signal f into multiple Instinct Mode Functions (IMF), and assume that each IMF has a limited bandwidth and a different center frequency.

During the decomposition calculation process of the VMD, the penalty factor α will affect the decomposition result of VMD together with the number of modal decomposition K. Among them, the number of modes K needs to be given before the VMD decomposition. When the value of K is too small and the decomposition of the original signal is insufficient, the original multiple modes of the signal will be aliased in one mode component, and even cause one of the modes states not to be estimated. On the contrary, when the value of K is too large and the signal is excessively decomposed, one of the modal components of the decomposed signal will appear in multiple modal components, so that the decomposed modal center frequencies overlap. The smaller the value of penalty factor α, the larger the bandwidth of each IMF obtained after decomposition; the larger the value of penalty factor α, the smaller the bandwidth of each IMF obtained after decomposition. Noise filtering and data enhancement can be effectively performed by choosing appropriate α and K. Based on the experimental test, the parameter α is selected as 1000. Fig. 2 shows the decomposition of the accelerometer signal data for K = 3, 4, 5, and 6. It can be seen that when K = 3, 4, and 5, there is a problem of insufficient decomposition; When K = 6, it can be seen from Fig. 2 that the fifth IMF in (c) is optimized in (d), the first five IMFs are valid for our classification, while the sixth IMF can

be considered as noise, which meets the requirements for data de-noising and signal enhancement. Therefore, for each sensor signal, this paper takes $K = 6$ and selects the first five IMFs as valid ones.



(a) K = 3 by VMD

(b) K = 4 by VMD

(c) K = 5 by VMD

(d) K = 6 by VMD

Figure 2: Sensor signal decomposition results by variational mode decomposition (VMD)

For the second point, to obtain the contextual relationship in the time series, this paper uses an LSTM network for model training [33]. The structure of the entire VMD+LSTM model is shown in Fig. 3, which contains an input layer (IMF signals), an LSTM layer, a classification layer, and an output layer. In the training phase, since the final output is whether the user of the mobile device is the owner, the input of the LSTM is the data of the user using the mobile device. Among them, K represents the number of modes (IMFs) determined by VMD, N represents the number of input signals, and L represents the length of each input (the duration of the signal). The parameters of hidden layers are obtained by tuning, including 2 hidden layers, each with 32 nodes. This refers to the experimental configuration in the well-known activity recognition literature [34]. The classification layer uses the softmax function to obtain the final results in the training phase. During the generation process of

the classification model at each device, an approach based on anomaly detection is employed to train individualized model parameters for each user.

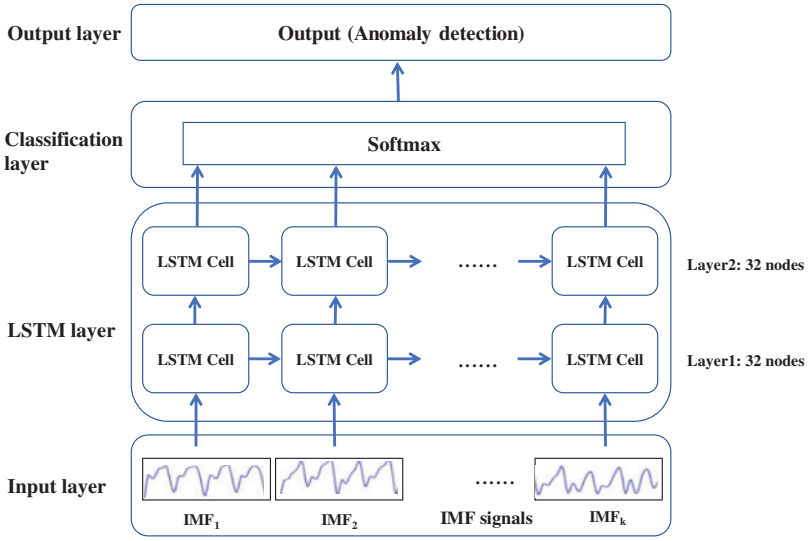


Figure 3: Structure of VMD + LSTM

3 Experimental Results and Analysis

3.1 Dataset

The dataset in this paper comes from a well-known Internet company, and the data is collected from 1513 volunteers in the age group of 20–60. To adapt to the dynamic user authentication of energy enterprise employees, this paper adopts the dynamic data when using the mobile phone as the authentication data. Relying on researchers' analysis and research on a large number of mobile users in the early stage [20,21], data collection will be activated when the following two requirements are met: first, the screen of the smartphone is on; second, the application in the foreground is switched. The above two conditions indicate that the user is opening a specific application and using the smartphone. During the data collection process, the accelerometer sensor, gyroscope sensor, and gravity sensor embedded in the phone captured 3-axis linear acceleration, 3-axis gyroscope velocity, and 3-axis gravity at a rate of 50 Hz. In this paper, a total of 283,133,354 pieces of raw sensor data were collected. After data preprocessing, 283,006,659 pieces were found to be valid, and the average number of effective records per user was 187,050. The obtained dataset was randomly divided into two groups, in which 80% of each volunteer's data was selected for training and 20% for testing.

3.2 Experimental Setup

We list all parameters and values used in this work in Table 1. In the experiment, the number of nodes belonging to volunteers in the private chain is $N = 1513$. In the process of node selection, this paper uses a 20% probability to simulate the situation where the power supply and wifi are connected, that is to say, about $|P| = 1513 \times 20\% = 302$ nodes are randomly selected for federated learning each time. The invalid model threshold for MAE was set to 0.05. In the process of generating the classification model of each node, the idea of anomaly detection is adopted to train its model parameters for each user. Meanwhile, in the federated learning global model update, we set $C = 0.5$, $E = 10$, $B = 32$.

Table 1: All parameters and values used in this work

N	1513
|P|	302
Threshold for MAE	0.05
C	0.5
E	10
B	32
K	6
Hidden layers of LSTM	2
Number of neurons in each layer of LSTM	32
Cost in SVM	100
Gamma in SVM	0.1
Kernel of SVM	RFB
Threshold for authentication	0.5

In VMD, the number of modalities (IMFs) K = 6, we pick up the first five IMFs.

In LSTM, the number of hidden layers is set to 2, and the number of neurons in each layer is 32, which refers to the experimental configuration in the famous activity recognition literature [34].

In the SVM setting, the parameters cost = 100, gamma = 0.1, and RBF is selected as the kernel function. The threshold for authentication in all trained models was set to 0.5.

It is worth noting that the above parameters are obtained through tuning, and in specific tasks, analysts can adjust them according to actual needs.

3.3 Evaluation Index

In terms of accuracy, we define the following evaluation matrix. True positive (TP), the owner is accurately marked. False positive (FP), others are marked as owner. True negative (TN), others are accurately marked. False negative (FN), the owner is marked as others. In terms of classification accuracy, the following indicators are used in this chapter:

The authentication rate of the user:

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

The authentication rate of others:

$$TNR = \frac{TN}{FP + TN} \tag{15}$$

Total accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{16}$$

3.4 Efficiency of Smart Contract-Based Private Chains

To evaluate the impact of the smart contract-based private chain on the efficiency of data collection, referring to the change in the number of mobile device sensor data collection over time in related research [24]. Assuming that the user's enthusiasm for data collection (the efficiency of data collection) is positively related to the incentive p, this paper simulates the change of the number of 1513 users' data collection over time when the user uses a mobile phone. As shown in Fig. 4.

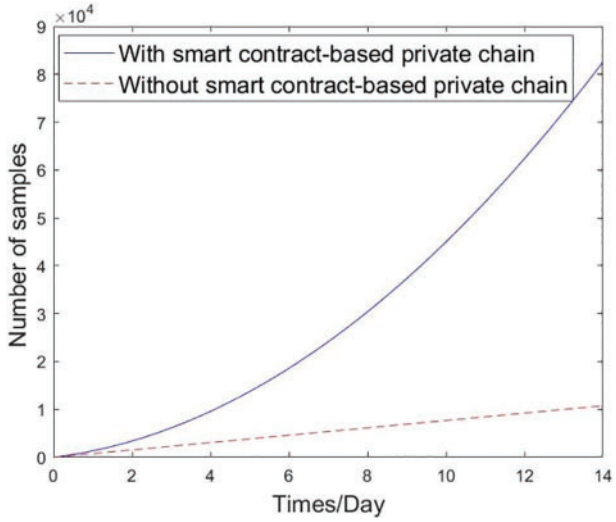


Figure 4: The impact of smart contract-based private chain on the efficiency of authentication data collection

Simulation results show that by deploying a smart contract-based private chain, the system can collect much more data at the same time. In the first week, the number of data samples can be about 25,000, which is 5 times that of the un-deployed case. After 2 weeks, the data samples of deploying the smart contract-based private chain can exceed 80,000, while the data samples without deployment are only about 10,000. It can be seen that the incentive mechanism in the smart contract-based private chain can greatly improve the efficiency of data collection.

3.5 Overall Accuracy

To evaluate the robustness of the proposed model based on VMD and LSTM, and the impact of federated learning on the final accuracy. This paper compares different kinds of methods according to the evaluation metrics proposed in Section 3.3. To ensure the generality of the experiment, all results are the average of 10 independent repeated experiments, and the accuracy statistics of the model are shown in Table 2.

We can summarize Table 2 as follows: First, the accuracy of the LSTM model is better than that of the SVM model, because the sensor data of mobile users is a time series, and LSTM can better capture the context information in the time series; Second, adding VMD before the training of the SVM and LSTM models (i.e., performing discrete wavelet decomposition and signal semantic enhancement on the time series signal in advance), can further improve the accuracy of the respective models. The improvement values for SVM and LSTM are 0.49% and 0.23% in cloud centralized training, 0.29% and 0.54% in federated learning distributed training; third, the use of federated learning distributed training has little impact on the final accuracy. Since the parameter training of the model is carried out

separately on the nodes of the respective users, there will be a certain degree of accuracy loss, but the overall accuracy loss (average) of the four methods is not more than 2%, which is within an acceptable range. To improve the practicability, some desensitized data of others for training can be stored in advance on each local user side to clarify the classification boundary and improve the accuracy.

Table 2: Index comparison of different training methods under cloud centralized training and federated learning distributed training

Model	Cloud centralized training			Federated learning distributed training		
Method/index	TPR	TNR	Accuracy	TPR	TNR	Accuracy
LSTM [34]	87.00%	97.93%	91.59%	85.93%	96.43%	90.75%
VMD + LSTM	87.28%	98.01%	91.82%	86.82%	97.70%	91.29%
SVM [20]	73.28%	98.43%	87.00%	72.12%	97.68%	85.52%
VMD + SVM	74.17%	98.55%	87.49%	73.18%	97.79%	85.81%

In order to further verify the robustness of the model, this paper collected the total accuracy average including average accuracy, maximum accuracy and minimum accuracy of 10 independent repeated experiments with different training methods under different model training methods, as shown in Fig. 5. It can be seen from Fig. 5 that the fluctuation of the accuracy of the model combining VMD and LSTM is significantly smaller than that of using only LSTM, which shows that using time series of different modalities (IMFs) can better train the model and improve the robustness of the model. Similarly, using VMD for SVM training can also improve the robustness of the model to a certain extent (compared to using SVM alone).

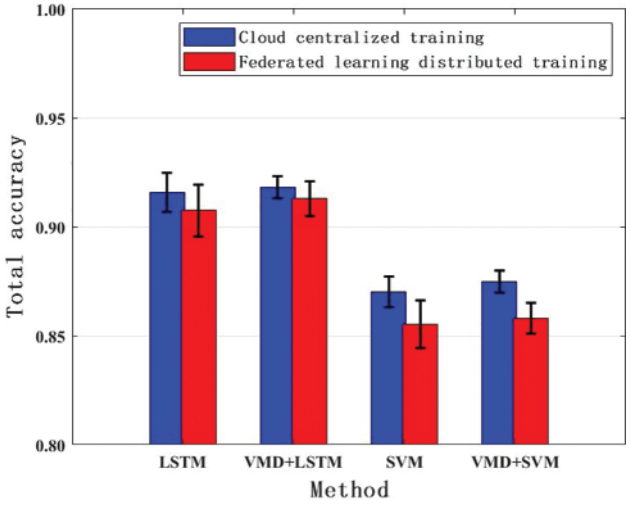


Figure 5: Total accuracy comparison of different training methods under different model training methods

3.6 Overhead

In this paper, we use the open-source performance testing tool Jmeter to conduct stress testing of the private chain. For each communication iteration process, there is time cost, memory overhead, and CPU overhead at each stage in the test. All stages include: node selection, model download, device authentication, local update upload, verification, block generation broadcast, and global model update. Among them, the memory overhead and CPU overhead in an iteration process are calculated for the Req node and the client participating in the operation (other nodes except Req). The results are shown in Table 3, and all results are the average of 10 independent experiments.

Table 3: Average time consumption, memory overhead and CPU overhead of different communication iteration stages of the private chain

Stage	Time consuming (s)	Memory (MB)		CPU (%)	
		Req	Other nodes	Req	Other nodes
Node selection	0.01	12.28	7.13	1.98	1.56
Model download	0.02	14.35	7.48	1.75	1.47
Device authentication	0.15	15.38	7.82	1.82	1.55
Local update upload	39.06	19.85	16.62	34.92	25.71
Verification	0.23	17.45	13.62	15.77	2.47
Block generation broadcast	0.06	16.25	11.80	2.61	2.53
Global model update	0.12	31.50	14.81	2.52	2.03

It can be seen from Table 3 that in the communication iteration process of the entire smart contract-based private chain, the most time-consuming operation is local update and upload, with an average time of 39.06 s, because the construction of the model requires data loading and operation operations such as VMD. The time for each node to accept the global model is short, and it can be completed in about 0.12 s with less communication. On the whole, the overhead of the smart contract-based private chain can meet the needs of the real-world environment. In addition, the consensus mechanism mainly depends on the number of clients (nodes) on the private chain. The greater the number of clients, the greater the time and resource overhead required. Therefore, choosing an appropriate number of clients will help improve system performance. On the other hand, the improvement of device configuration also helps to reduce the delay in the running process.

3.7 Anti-Poisoning Ability

To evaluate the resistance to poisoning attacks (i.e., the data of the trusted node is tampered with by the attacker) after adopting the smart contract-based private chain, this paper randomly poisons 70% of the training users (i.e., node P). The poisoning ratio is selected in node P according to 0%, 10%, 20%, 30%, 40%, 50%, 60%, and 70%. After the poisoned users are randomly selected, the training data is continuously randomly perturbed up and down, and the value after the perturbation is guaranteed to be still within the legal data value range of the sensor. In this experiment, the VMD+LSTM model with the best performance in Section 3.5 is selected as the training model. Table 4 shows the anti-poisoning ability of the model under different training methods.

Table 4: Anti-poisoning ability under different training methods (VMD + LSTM model)

Poisoning ratio	Cloud centralized training	Federated learning distributed training	Smart contract-based private chain + federated learning distributed training
0%	91.59%	90.75%	90.75%
10%	89.65%	89.18%	90.68%
20%	85.38%	85.04%	90.35%
30%	78.52%	77.30%	90.08%
40%	71.40%	70.10%	89.40%
50%	62.06%	61.05%	87.25%
60%	55.38%	50.77%	85.19%
70%	50.42%	50.10%	82.60%

Three key points of information can be observed from the experimental results in Table 4: Firstly, when the samples are poisoned, the accuracy of cloud-based centralized training and federated learning distributed training decreases with the increase of poisoned users. Because the training data is tampered with, the model obtained by training cannot effectively authenticate the test data. Secondly, when the sample is poisoned, the smart contract-based private chain with federated learning distributed training can effectively judge the training quality of the node due to its trust mechanism. It mainly includes two aspects: (1) Req will mark the local model whose MAE is higher than a certain threshold as an invalid model, and reset the trust degree of the node according to the historical valid model supply ratio of the node. (2) Select nodes with high trust for distributed training of federated learning. Therefore, it is not obvious that the test accuracy decreases with the increase of poisoned users. Thirdly, when the samples are poisoned, the accuracy of cloud-based centralized training and federated learning distributed training will remain after a certain level of decrease (about 50%). Stability is also in line with the logic in daily life: that is, the judgment of the user is either 0 or 1 in an unknown situation, such as random guessing whether it is the owner or others.

To sum up, after adopting a smart contract-based private chain, distributed training of federated learning can not only ensure the privacy of user data but also effectively resist poisoning attacks and maintain the robustness of the model.

4 Related Work

Data collection efficiency of mobile user authentication. The existing method of collecting motion sensor data is to invite specific participants to collect data in certain fixed scenarios, such as activities such as going up and down stairs [18,19,25–27], picking up mobile devices [22,23], and touching the screen of the mobile device [24]. In the above motion sensor-based user dynamic authentication, once a new user joins, the complex collection process will be repeated, which is unrealistic in real-world authentication scenarios. Zhu et al. [20,21] studied the usage patterns of different groups of people and proposed an implicit real-world data collection mechanism to provide large-scale data. However, users have different habits/frequencies in using mobile devices (i.e., some users may use the phone for several hours a day, while some users use it for less than an hour a week). For the latter, it takes a long time to obtain a sufficient amount of valid training data, so there is a large delay in the generation

of the certified model. Therefore, in the real complex environment, the current data collection scheme lacks an effective incentive mechanism, and there are problems such as a single authentication scenario that requires the user to cooperate in advance, low data collection efficiency, etc. Different from the above methods, this paper adopts a private chain method based on smart contracts to send certain rewards to employees corresponding to each node that participates in training and helps optimize the model, so as to motivate nodes to collect data and improve the efficiency of data collection in real-world environment.

Data privacy protection and device security of mobile user authentication. The existing mobile device user authentication work [18–28] needs to generate a model for each user in the central cloud, which requires users to upload their authentication data to the cloud. Moreover, this method requires users to upload human-computer interaction sensor data for a period of time for model generation. However, uploading their "identity" data will cause users to worry about their own privacy leakage. In addition, if the user's node is poisoned (the authentication data is tampered with by the attacker), it will also affect the final model. Different from existing methods, inspired by existing studies [35,36] that utilize blockchain for trusted transactions, this paper proposes a modeling scheme based on blockchain and federated learning to protect data privacy, which can perform effective training without data leaving the local node. At the same time, this paper proposes a device-trusted authentication method to prevent untrusted device access. In addition, this paper uses a consensus mechanism-based training quality proof to validate the model to effectively resist model poisoning attacks and thus prevent unreliable model updates.

Intelligent modeling in complex environments of mobile user authentication. In terms of data de-noising: At present, most dynamic authentication methods utilizing motion sensors [19,22–28] do not consider the noise impact of hardware and cannot handle unlabeled data (noisy data) in real environments, resulting in an over-fitting problem. To address the noise problem, some researchers [18] proposed noise removal algorithms to obtain effective datasets in the data preprocessing stage, but they often faced the challenge of mislabeling and ambiguous distinctions between training samples. To overcome this difficulty, researchers employ semi-supervised methods that combine noisy data with a set of clean labels [20,21]. Zhu et al. [20,21] observed that the flat data could not reflect the differences between different user modes in the data collection stage, but their usability analysis of the collected data was omitted. In terms of model building: existing research methods for mobile user authentication using motion sensors [18,22–28] verify users by continuously collecting sensor data and building corresponding models, Lu et al. [19] adopted an unsupervised learning algorithm to deal with the labelled data, but their method introduced high latency issues. In addition, unsupervised clustering algorithms with parameter tuning are expensive, and the generalization ability of the parameters needs to be verified. Zhu et al. [20] designed a semi-supervised online learning algorithm with high accuracy and low latency in processing unlabeled data in a relatively complex environment. However, the classification method (binary SVM-like) is not suitable for time series data in complex scenarios and does not consider the context of user behavior. In addition, most of the existing works [18–28] assumed that the data input to the model is sufficient, and do not consider the lack of data in real complex environments. Different from the above methods, this paper proposes a mobile user authentication model based on VMD and LSTM to perform efficient de-noising, data augmentation, and contextual semantic analysis of time series signals to make up for the lack of coverage and accuracy when using traditional statistical feature modeling.

5 Conclusion

This paper proposes a privacy-preserving mobile user authentication method, which guides a new type of mobile user dynamic authentication. The advantages of this paper are as follows: First, use the smart contracts-based private chain and federated learning to improve the data collection efficiency of mobile user authentication, reduce the probability of the model being bypassed by attackers, and avoid high overhead and privacy leakage caused by centralized data processing. Second, the authentication of the device is realized by using certificateless encryption to ensure the trustworthiness of the client nodes participating in the calculation. Finally, combined with VMD and LSTM, the motion sensor data of mobile devices are analyzed and modeled to reduce the data noise and improve the accuracy of model authentication. The experimental results on the real environment dataset of 1513 people show that the proposed method achieves satisfactory accuracy and is superior to the existing work. In the future, we will research how to efficiently train user authentication models with a small number of samples, and conduct experiments with more authentication scenarios.

Acknowledgement: We would like to thank all editors and reviewers for the review efforts.

Funding Statement: This paper is sponsored by Wenzhou Key Scientific and Technological Projects (No. ZG2020031), supported by Wenzhou Polytechnic Research Projects (No. WZY2021002), supported by Key R&D Projects in Zhejiang Province (No. 2021C01117), sponsored by Major Program of Natural Science Foundation of Zhejiang Province (LD22F020002), supported by the Cloud Security Key Technology Research Laboratory. This work was funded by the Researchers Supporting Project Number (RSP2023R509), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: Each author contributed extensively to the preparation of this manuscript. Chunlin Xiong, Zhengqiu Weng, Jia Liu, and Fayez Alqahtani designed the experiment; Chunlin Xiong, Amr Gafar, and Fayez Alqahtani performed the experiments; Liang Gu, Fayez Alqahtani and Amr Gafar collected the data; Zhengqiu Weng, Amr Gafar, Liang Gu and Fayez Alqahtani analyzed the data; Fayez Alqahtani, Amr Gafar, Jia Liu and Liang Gu designed the software; Chunlin Xiong, Fayez Alqahtani, Amr Gafar and Liang Gu wrote the paper, Fayez Alqahtani, Pradip Kumar Sharma and Zhengqiu Weng provided the funding acquisition and supervision.

Availability of Data and Materials: Due to privacy concerns, our mobile authentication user dataset is temporarily not open-sourced.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. CCS insight forecast: Wearables Momentum Continues (2022). https://www.ccsinsight.com/press/company-news/2516-wearables-momentum-continues/ (accessed on 29/03/2022)
2. Sepideh, F. (2019). Providing a secure hybrid method for graphical password authentication to prevent shoulder surfing. *Smudge and Brute Force Attack. International Journal of Computer and Information Engineering, 13(12),* 624–628. https://doi.org/10.5281/zenodo.3593252
3. Shin, H., Sim, S., Kwon, H., Hwang, S., Lee, Y. (2022). A new smart smudge attack using CNN. *International Journal of Information Security, 12,* 1–12.

4. Aviv, A. J., Davin, J. T., Wolf, F., Kuber, R. (2017). Towards baselines for shoulder surfing on mobile authentication. *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 486–498. Orlando, FL, USA.
5. Xu, Z., Bai, K., Zhu, S. (2012). Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 113–124. Tucson, Arizona, USA.
6. Wang, S., Yuan, J., Chen, S. (2020). Quality-based score level fusion for continuous authentication with motion sensor and face. *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, pp. 58–62. Nanjing, China.
7. Arasu, D. B. L., AzlanMohamed, A. S., Ruhaiyem, N. I. R., Annamalai, N., Lutfi, S. L. et al. (2022). Human stress recognition from facial thermal-based signature: A literature survey. *Computer Modeling in Engineering & Sciences, 130(2),* pp. 633–652. https://doi.org/10.32604/cmes.2021.016985
8. Iqbal, S., Irfan, M., Ahsan, K., Hussain, M. A., Awais, M. et al. (2020). A novel mobile wallet model for elderly using fingerprint as authentication factor. *IEEE Access, 8,* 177405–177423.
9. Yang, C., Zhang, J., Guo, J., Zheng, Y., Yang, L. et al. (2019). Fingerprint protected password authentication protocol. *Security and Communication Networks, 2019,* 1694702.
10. Johnson, R. C., Scheirer, W. J., Boult, T. E. (2013). Secure voice-based authentication for mobile devices: Vaulted voice verification. In: *Biometric and surveillance technology for human and activity identification X,* vol. 8712, pp. 164–176. Baltimore, Maryland, USA.
11. Yan, Z., Zhao, S. (2019). A usable authentication system based on personal voice challenge. *2016 International Conference on Advanced Cloud and Big Data (CBD)*, pp. 194–199. Chengdu, China, IEEE.
12. Rexha, B., Shala, G., Xhafa, V. (2018). Increasing trustworthiness of face authentication in mobile devices by modeling gesture behavior and location using neural networks. *Future Internet, 10(2),* 17.
13. Fujio, M., Takahashi, K., Kaga, Y., Nakamura, W., Yasumura, Y. et al. (2021). Template protected authentication based on location history and b-Bit MinHash. *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pp. 1–8. Vienna Austria.
14. de-Marcos, L., Martínez-Herráiz, J. J., Junquera-Sánchez, J., Cilleruelo, C. et al. (2021). Comparing machine learning classifiers for continuous authentication on mobile devices by keystroke dynamics. *Electronics, 10(14),* 1622.
15. Cui, Z., Huang, A., Chen, J., Gao, S. (2021). Piezoelectric touch sensing-based keystroke dynamic technique for multi-user authentication. *IEEE Sensors Journal, 21(23),* 26389–26396.
16. Zaidi, A. Z., Chong, C. Y., Jin, Z., Parthiban, R., Sadiq, A. S. (2021). Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities. *Journal of Network and Computer Applications, 191,* 103162.
17. Filippov, A. I., Iuzbashev, A. V., Kurnev, A. S. (2018). User authentication via touch pattern recognition based on isolation forest. *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1485–1489. Moscow and St. Petersburg, Russia, IEEE.
18. Ren, Y., Chen, Y., Chuah, M. C., Yang, J. (2014). User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Transactions on Mobile Computing, 14(9),* 1961–1974.
19. Lu, H., Huang, J., Saha, T., Nachman, L. Unobtrusive gait verification for mobile phones. *Proceedings of the 2014 ACM International Symposium on Wearable Computers*, pp. 91–98. Seattle, Washington, USA.
20. Zhu, T., Qu, Z., Xu, H., Zhang, J., Shao, Z. et al. (2019). RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Transactions on Mobile Computing, 19,* 466–483. https://doi.org/10.1109/tmc.2019.2892440
21. Zhu, T., Weng, Z., Song, Q., Chen, Y., Liu, Q. et al. (2020). Espialcog: General, efficient and robust mobile user implicit authentication in noisy environment. *IEEE Transactions on Mobile Computing, 21(2),* 555–572.

22. Jiang, Z., Pang, W., Xiao, W., Zhang, J. (2013). SenSec: Mobile security through passive sensing. *Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC)*, pp. 28–31, San Diego, CA, USA, IEEE. https://doi.org/10.1109/iccnc.2013.6504251
23. Buriro, A., Crispo, B., Zhauniarovich, Y. (2017). Please hold on: Unobtrusive user authentication using smartphone's built-in sensors. *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pp. 1–8. New Delhi, India, IEEE.
24. Shen, C., Li, Y., Chen, Y., Guan, X., Maxion, R. A. (2017). Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security, 13(1),* 48–62.
25. Ehatisham-ul-Haq, M., Awais Azam, M., Naeem, U., Amin, Y., Loo, J. (2018). Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing. *Journal of Network and Computer Applications, 109,* 24–35. https://doi.org/10.1016/j.jnca.2018.02.020
26. Volaka, H. C., Alptekin, G., Basar, O. E., Isbilen, M., Incel, O. D. (2019). Towards continuous authentication on mobile phones using deep learning models. *Procedia Computer Science, 155,* 177–184. https://doi.org/10.1016/j.procs.2019.08.027
27. Centeno, M. P., Guan, Y., van Moorsel, A. (2018). Mobile based continuous authentication using deep features. *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning*, Munich, Germany, ACM Press. https://doi.org/10.1145/3212725.3212732
28. Zhu, T., Weng, Z., Chen, G., Fu, L. (2020). A hybrid deep learning system for real-world mobile user authentication using motion sensors. *Sensors, 20(14),* 3876.
29. Li, T., Sahu, A. K., Talwalkar, A., Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37(3),* 50–60.
30. Anh, T. T., Luong, N. C., Niyato, D., Kim, D. I., Wang, L. C. (2019). Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach. *IEEE Wireless Communications Letters, 8(5),* 1345–1348.
31. Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.
32. Dragomiretskiy, K., Zosso, D. (2013). Variational mode decomposition. *IEEE Transactions on Signal Processing, 62(3),* 531–544.
33. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8),* 1735–1780.
34. LSTMs for Human Activity Recognition (2022). https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition/ (accessed on 29/03/2022).
35. Wang, J., Wei, B., Zhang, J., Yu, X., Sharma, P. K. (2021). An optimized transaction verification method for trustworthy blockchain-enabled IIoT. *Ad Hoc Networks, 119,* 102526.
36. Zhang, J., Zhong, S., Wang, J., Yu, X., Alfarraj, O. (2021). A storage optimization scheme for blockchain transaction databases. *Computer Systems Science and Engineering, 36(3),* 521–535. https://doi.org/10.32604/csse.2021.014530