ESPIALCOG: General, Efficient and Robust Mobile User Implicit Authentication in Noisy Environment

Tiantian Zhu, Zhengqiu Weng, Qijie Song, Yuan Chen, Qiang Liu, Yan Chen, *Fellow, IEEE,* Mingqi Lv, Tieming Chen

Abstract—Mobile authentication is a fundamental factor in the protection of user's private resources. In recent years, motion sensor-based biometric authentication has been widely used for privacy-preserving. However, it faces with the problems including low data collection efficiency, insufficient authentication scenario coverage rate, weak de-noising ability, and poor robustness of models, rendering existing methods difficult to meet the security, privacy, and usability requirements jointly in the real-world scenario.
To overcome these difficulties, we propose a system called ESPIALCOG, which is able to 1) collect the sensor data embedded in mobile devices self-adaptively, unobtrusively and efficiently through the evolutionary stable participation game mechanism (ESPGM) with a high scenario coverage rate, 2) minimize noise from collected data by analyzing three types of abnormalities, and 3) authenticate the ownership of mobile devices in real-time by adopting optimized LSTM model with an enhanced stochastic gradient descent (SGD) algorithm. The simulation experiment on 6000 users shows that the efficiency and coverage rates increase dramatically by deploying our ESPGM. Moreover, we conduct experiments on a large-scale real-world noisy dataset with 1513 users and two other small pure real-world datasets. The experimental results show the high accuracy and favorable robustness of ESPIALCOG in the noisy environment.

Index Terms—User Authentication, Mobile Device, Game Theory, Deep Learning.

1 INTRODUCTION

RECENT hardware advances have led to the development and consumerization of mobile devices. GPU, TPU, and other chips are increasingly integrated into mobile phones to meet the growing complex computing demand. Meanwhile, numerous sensors (acceleration, gyroscope, light, heartbeat sensors, etc.) are used to complete various practical tasks. With the coming of 5g Era, a recent global survey by the world's leading market researcher (CCS Insight) forecasts that 0.84 billion 5G-enabled mobile phones will be shipped in 2022, accounting for 42% of total global shipments [1]. In order to prevent illegal access to private information stored in mobile devices, it is urgent to design appropriate and robust authentication mode to protect users' information security according to the characteristics of hardware/software and application scenarios of those devices. At present, user authentication methods on mobile devices can be classified into three categories: credential-based, static characteristics-based, and dynamic behavior-based. Traditional credential-based authentication methods (e.g., text passwords, PIN codes and patterns, etc.) are widely used to unlock mobile devices and log in applications. During authentication, users need to explicitly enter authentication information. Such methods simply verify that the user has entered the account credentials correctly, not that the user is trusted. Also, previous studies show that this kind of authentication method is easy to be cracked by brute-force attacks [2], touchscreen smudges [3], shoulder attack [4] and sensor-based inferring [5]. Compared with credential-based authentication, static characteristics-based methods are based on user static biometric characteristics such as fingerprint and face, which can achieve relatively high authentication accuracy. However, frequent human-computer interaction may bring bad user experience, and biological content collection may also arouse users' concerns about leakage of their private information. Moreover, the latest research shows that the misuse of fingerprint API on Android will make apps vulnerable to multiple attacks [6], fingerprint recognition systems are easy to be compromised by image-level MasterPrints [7], and deep learning-based face recognition systems are also proven to be bypassed by sophisticated attackers [8, 9].

Considering the weakness of the credential technology-based and user static characteristics-based authentication, motion sensor-based dynamic user authentication is proposed by many researchers [10–22]. The motion sensor-

- *This work is supported in part by the following grants: National Natural Science Foundation of China under Grant No. U1936215 and 61772026. Ministry of Industry and Information Technology of the People's Republic of China under Grant No. TC190H3WN.*
- *Tiantian Zhu, Qijie Song, Yuan Chen, Mingqi Lv and Tieming Chen* are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China. E-mail: ttzhu@zjut.edu.cn, 2431712530@zjut.edu.cn, 2111712038@zjut.edu.cn, mingqilv@zjut.edu.cn, tmchen@zjut.edu.cn. *corresponding author
- *Zhengqiu Weng is with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou,Zhejiang 310023, China, and she is also with Department of Information Technology, Wenzhou Polytechnic, Wenzhou, Zhejiang 325035, China. E-mail: derisweng@qq.com.
- *Qiang Liu is with the College of Computer Science and Technology, Zhejiang University , Hangzhou 310027, China. E-mail: qiangliu@zju.edu.cn.
- *Yan Chen is with Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA. E-mail: ychen@northwestern.edu.

TABLE 1
Comparison with related studies on smartphone sensor-based authentication. L represents low, M represents medium, and H represents high. NA represents the information is not mentioned.

Study	Efficiency			Scenario coverage	De-noise ability	Robustness	Accuracy
	Require user movement	Device placement	Authentication Latency				
ESPIALCOG	No	Dynamic	L	H	H	H	H
RISKCOG (2019) [10]	No	Dynamic	L	M	L	M	H
Derawi et al. (2010) [11]	Yes	Fixed	NA	L	NA	L	L
Kwapisz et al. (2010) [12]	Yes	Fixed	NA	L	NA	L	M
Ho et al. (2012) [13]	Yes	Fixed	NA	L	NA	L	NA
Zhu et al. (2013) [14]	No	Fixed/Dynamic	H	L	NA	L	L
Lu et al. (2014) [15]	Yes	Dynamic	M	L	NA	L	M
Kayacik et al (2014) [16]	Yes	Fixed	H	L	NA	L	M
Ren et al. (2015) [17]	Yes	Fixed	NA	L	NA	L	M
Lee et al. (2015) [18]	Yes	Fixed	M	L	NA	L	M
Sitová et al. (2016) [19]	Yes	Fixed	M	L	NA	L	H
Lee et al. (2017) [20]	No	Fixed	L	L	NA	L	M
Buriro et al. (2017) [21]	Yes	Dynamic	NA	L	NA	L	H
Shen et al. (2018) [22]	Yes	Fixed	M	L	NA	L	H

based mechanisms can be applied to many common scenarios. For example, Alice and Bob are classmates. One day, Alice leaves her smartphone at the desk without turning off the screen. Thus it is possible for Bob to check Alice's WeChat private activities without her consent if WeChat's automatic login option is enabled. In this case, sensor-based authentication approaches which are running in the background can detect the unauthorized user access implicitly, and then invoke the follow-up self-defense actions, such as privately alerting the phone owner of the suspicious access by email, rendering an empty page or demanding for retyping the password of Alice's WeChat. Among all the existed sensor-based user authentication work, the most representative one is RISKCOG [10], which overcomes the shortcomings of previous work and exploits a dynamic and implicit real-time user authentication method. But the low data collection efficiency, insufficient authentication scenario coverage rate, weak de-noising ability, and poor robustness of models, rendering existing motion sensor-based authentication methods difficult to meet the security, privacy, and usability requirements jointly in mobile user authentication.

To fill this critical research gap, in this paper we design an effective and accurate sensor-based user authentication system, called ESPIALCOG[1]. In Table 1, we list the problems with previous sensor-based approaches and summarize the following challenges:

(1) Low data collection efficiency. Motion sensors (including accelerometer, gyroscope, and gravity) are widely embedded in smart-phones as privacy-independent sensors. However, the traditional way of data collection is to invite a small number of participants to provide sensor data, users are required to restart collecting new data of each gait pattern, which wastes human/ material resources and leads to low collection efficiency [11–13, 15–19, 21, 22]. Furthermore, in the real complex environment, users are not required to perform specific behaviors or to fix the location of mobile phone, but the collecting process is not controllable, and the time required to obtain sufficient training data will be longer [10]. It is crucial to stimulate smart-phone users to actively participate in sensing processes by contributing sensing data.

(2) Insufficient authentication scenario coverage rate. Previous studies [11–13, 16–18] have a strong assumption that the smart device placement should be fixed (e.g., attached on the user's leg). The single device location makes it hard to cover various authentication status and behavior patterns. RISKCOG [10] collects data at the start of apps of different types and has a relatively higher coverage rate of the user's behavior patterns. It is still insufficient because RISKCOG does not consider the user patterns of different applications in the same type (e.g., for the same type of chat applications, one may use DingTalk during office hours and use Instagram at the spare time). Collecting data from different apps of the same type helps us to model the behavior of users using mobile devices comprehensively. Therefore, we need to collect a large amount of sensor data in different application scenarios to improve model accuracy.

(3) Weak de-noising ability. Considering the low-cost and portability of mobile devices, the initial reading of their built-in motion sensors is often influenced by inherent factors such as materials and workmanship. In addition, the value of the motion sensor is also affected by external factors such as temperature, humidity, and age of use, which make the direct reading of mobile device motion sensor inaccurate. Then the data deviation will directly lead to the accuracy of the final authentication. Therefore, the error correction and de-noising of motion sensor are of great importance. The noise impact of the hardware is hardly considered by previous studies [11–22], while the existing de-noising technique [10] can remove the flat data, it's still thought to be one-sided.

(4) Poor robustness of models. On the one hand, the traditional sequence matching algorithm [11, 17, 20] and machine learning methods [12, 14–16, 18] are not suitable for time series data in complex scenarios, and can not take into account the contextual issues of behavior. On the other hand, the robustness of existing models is not strong, the data set might be simulated, and there are problems such

1. ESPIAL for mobile user Implicit Authentication through Evolutionary Stable Participation game and Lstm. COG for cognition.

as no labels, data loss, etc. Although a semi-supervised online learning algorithm is proposed in [10] to address the unlabeled data issue, the model should be re-trained when the new data samples are uploaded, which can introduce a large number of computing costs.

Compared with the state-of-the-art work RISKCOG [10], our system has the following advantages. 1) We propose an evolutionary stable participation game framework to collect the sensor data embedded in mobile devices self-adaptively, unobtrusively, and efficiently with a high scenario coverage rate. 2) We deploy three methods for data de-noising to further eliminate the noise impact of the hardware. 3) We present an optimized LSTM network for data training, which greatly improved the accuracy and robustness of the model. Finally, our system can authenticate the user owner unobtrusively utilizing the well-trained LSTM model. In summary, we make the following contributions:

- We design a heuristic data collection mechanism based on participation game theory to reach high data collection efficiency. Moreover, we propose an evolutionary stable mechanism to improve the coverage rate of various application scenarios for authentication by collecting 1) data of different types of applications and 2) data from multiple applications in the same type.
- Through the study of a large number of motion sensor data in the real-world scenario, we propose a data de-noising technique to recognize and remove data abnormalities such as Equal-Value abnormalities, Jump-Point abnormalities and Zero-Value abnormalities .
- We propose an accurate and robust LSTM network architecture which considers the time series based contextual issues of user behavior. Moreover, we implement an enhanced SGD algorithm to minimize the impact of noisy labels during the training phase to improve the robustness of the model.
- We achieve high accuracy for implicit user authentication under the noisy environment. The experimental results show that ESPIALCOG gains the classification accuracy values of 87.00% and 97.93% for the user owner and others, respectively. Our system can surpass the existing methods in the efficiency of data collection, the sufficiency of scenario coverage rate, the comprehensiveness of de-noising ability and the robustness of the model.

In practice, ESPIALCOG could be used as a third-party service to perform implicit authentication firstly. If it fails, other explicit authentications or effective countermeasures would be then leveraged. The remainder of this article is organized as follows: In Section 2, we cover ESPIALCOG design in detail. Section 3 presents the overall evaluation of our system. Section 4 discusses the shortcomings of our work and proposes countermeasures. Section 5 surveys the relevant work. Section 6 concludes our work.

2 SYSTEM DESIGN

In this section, we first introduce the usage scenario and overall architecture of our proposed user authentication mechanism for mobile devices. We then discuss several important topics in its design, including evolutionary stable participation mechanism, data de-noising, optimized LSTM, and user authentication.

2.1 Usage Scenario and System Overview

The usage scenario of ESPIALCOG is as follows.

In the training phase: Alice has deployed ESPIALCOG on her smartphone. Each time she is opening some applications (e.g., WeChat, Instagram), motion sensors (including acceleration sensor, gyroscope sensor, and gravity sensor) embedded in smartphones will sense and collect her behavior related data, and then upload it to cloud servers. At the same time, she will receive the payoff which is calculated by the payoff function (in Algorithm 1). The payoff information obtained from both population state and system average payoff will guide her to decide whether to change another sensing process (e.g., use game or news related applications) or keep the current sensing process (e.g., use chat related applications such as WeChat) by comparison. Then the uploaded data will be de-noised and normalized for LSTM model training. Finally, the model on the server will be pushed to the smartphone when a WiFi connection is available.

In the authentication phase: Alice has the trained LSTM model on the smartphone. One day, Alice leaves her smartphone at the desk without turning off the screen. Bob finds the smartphone and tries to check Alice's WeChat private activities without her consent. When Bob is opening the WeChat application, the data of motion sensors will be collected, pre-processed (data de-noising and normalization), and then fed into the LSTM model to authenticate the owner. Finally, Bob will be judged as a suspicious user and ESPIALCOG will invoke the follow-up self-defense actions.

The architecture of our system is illustrated in Figure 1. We develop a mobile application for the purpose of data collection, data de-noising, and user authentication. The application needs to detect the duration that the device is being actively used because only the sensor readings during such a duration are effective to represent user's manner [10]. On the contrary, the computation-intensive tasks such as ESPGM and optimized LSTM training with enhanced SGD algorithm are offloaded to cloud servers to conserve the on-device battery energy and computing power (data de-noising is also used in the training phase on the server). Noting that ESPGM is only used in the training phase, while the authentication is unobtrusive. Moreover, to resolve the issue of practicality (e.g., network is unreachable), we decouple the authentication from the server side. The generated model will be pushed to the client for real-time authentication. We will describe the key components in the following article.

2.2 Data Collection

Mobile devices are equipped with a list of various sensors. The more sensors and sensory data being used by the system, the harder the system to be circumvented. However, this will also increase the chance of genuine device users being rejected. In addition, the private and sensitive data collected by sensors and used for authentication may

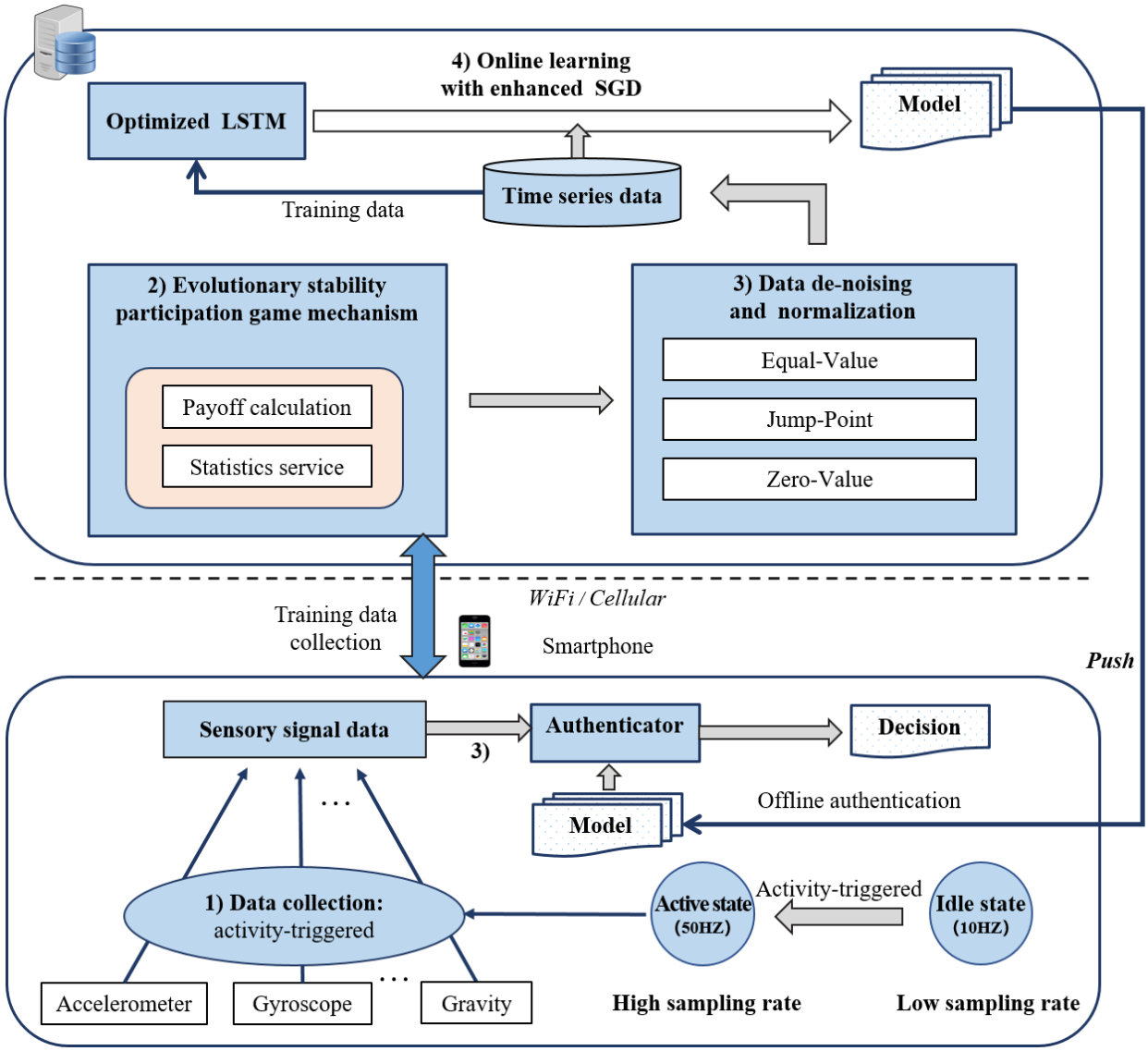


Fig. 1. System architecture; including activity-triggered data collection, data de-noising and offline real-time authentication in the client side, evolutionary stable participation game mechanism, data de-noising and online learning with enhanced SGD algorithm in the server side.

suffer from data disclosure and privacy breach and hence raise privacy concerns. Therefore, it is considered important to strike a balance among security, privacy, and usability. Firstly, the sensors required for authentication should be widely used in existing mobile devices. The type of sensor built into a mobile device is affected by factors such as device manufacturer, price, and population orientation. For example, the heart rate sensor is more common on personally worn devices such as smartwatches, but is not equipped on general smartphones. Although kinetic energy collectors have proven to be a high-precision sensor [23], which can effectively capture the moment of user's movement momentum and thus used for user authentication. However, it is relatively rare in current mobile devices. Common motion sensors, such as acceleration sensors, gyroscope sensors, and gravity sensors, have become an integral part of mobile devices. Due to their high sampling rate and low energy consumption, these two motion sensors are widely used to collect daily activity data of users. Secondly, the sensors for authentication should be independent of user privacy. Mobile devices are generally integrated with sensors such as cameras, microphones, and global positioning systems (GPS). On Android, privacy-related permissions will be invoked if the application wants to get the camera/voice/location information, which undoubtedly renders users worry about their privacy being leaked. In addition, face information is easy to be stolen, sound information is easy to be recorded, and geographic information is easy to be simulated. These emerging phenomenons make the above-mentioned sensors also have certain security risks, so this type of sensor is not considered in this article. Thirdly, the sensors demanded for authentication should be insensitive to changes in the external environment. The identity authentication system expected in this article can work normally in any scenario, which can meet the new requirements on the selection of sensors. Those sensors that are easily affected by the external environment will not be considered in this article. For example, the sound sensor (microphone) will not be able to obtain sound information from the user in a noisy environment. The camera sensor cannot accurately identify the information of the object that needs to be photographed in a poor light environment. The capacitive sensor (such as

some fingerprint sensors) cannot work normally in a humid environment, and the temperature sensors and magnetic sensors are greatly affected by external air temperature and the earth's magnetic field. Therefore, ESPIALCOG choose acceleration sensor, gyroscope sensor, and gravity sensor for data collection and final authentication.

An ideal data collection should be able to collect sufficient data for authentication while consuming minimal battery power, which calls for a smart data collection design. Typically, the applications of mobile devices always stay one of the two states: idle state and active state. Idle state refers to the situation that the application is suspending in the background or the user is not performing actions on the device. An active state refers to the situation that the user is performing an activity on the device and the application is running in the foreground. Based on the observation that the embedded sensors nearly cannot collect data specifically meaningful for authentication while the application of the device is in idle states, like the previous work [10], we propose a smart activity-triggered data collection approach to achieve the goal of collect effective data with minimal power consumption. As shown in Figure 1, the core idea is that sensors will collect data in a low sampling rate for energy conservation until some events specifically meaningful for authentication happen. Specifically, when the application of the device is in the idle state (The screen of the phone is off or no new applications are running in the foreground), the data collection process should be suspended and only periodically query specific sensors for the possible state change at a quite low rate. Once the application device becomes active from the idle state (The screen of the phone is on and a new application is running in the foreground), the data collection process resumes and collects data with a relatively high sampling rate. Take the smartphone as an example. Every time the user has an interaction with the phone by opening an application, the package name of the application running in the foreground will be changed, which is the time for the data collection to resume. Android allows developers to refresh the sensor data in customized intervals/delays. Considering the battery consumption, we empirically set the sampling frequency of 10HZ in idle state and that of 50HZ in an active state. Moreover, we set the duration of our data collection time as 3 seconds, which has been discussed in [10] to be the optimum.

2.3 Evolutionary Stable Participation Game Mechanism

To efficiently collect data that represent user behavior and to increase data coverage rate in the training phase, firstly, we present an evolutionary stable participation game mechanism, then apply evolutionary dynamics to provide optimal strategies for the participants. The system will eventually reach the evolutionary equilibrium following the evolutionary dynamics.

We consider a shared sensing system with a server and a set of smartphone users $\mathcal{U} = 1, 2, ..., U$. The size of $\mathcal{U}$ depends on the number of users participating in the authentication. The server has a limited predefined set of sensing processes $\mathcal{P} = 1, 2, ..., P$, each of which contains a series of tasks aimed at collecting sensor data from different types

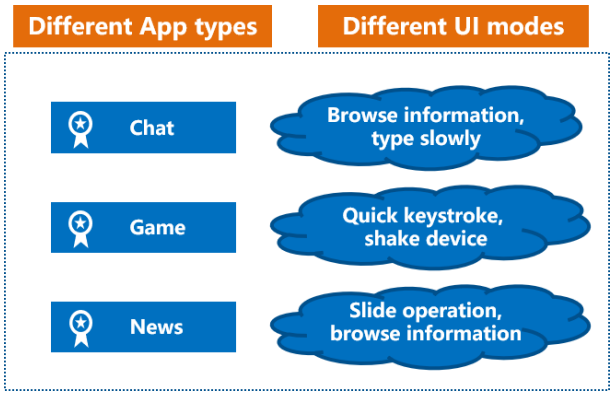


Fig. 2. Different application types will have different UI modes.

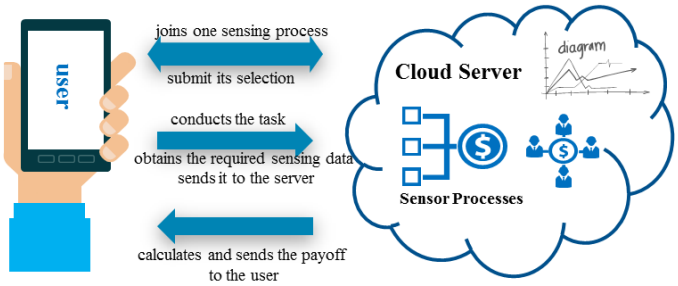


Fig. 3. Sensing process in our ESPGM, users interact with the cloud server dynamically.

of applications. Inspired by [10], we consider three different types of applications (three sensing processes): game, chat, and news, as shown in Figure 2. Specifically, one task is collecting sensor data when opening some applications of one type for n seconds ($n = 3$ seconds), and one user can complete each task in a time slot (e.g., if the sensing process represents a user interacts with chat applications, the corresponding tasks contain opening and using Instagram, Facebook, WeChat, DingTalk, etc).

The specific applications used by users may vary at a different time (use DingTalk during office hours and use Wechat in spare time), so the sensing process from an application of the same type is divided into multiple time slots to obtain different applications of the same type, which can make data coverage rate wider. The budget S_i for each type to collect the sensing process is limited [24]. All users participate in the same sensing process i share S_i in the same time slots.

The process of participatory sensing is described as follows: One user joins a sensing process and submits its selection to the server. The user then performs the task of a sensing process that belongs to a type of applications, obtains the required sensing data, and sends it to the server. The server receives sensing data, calculates and sends the payoff to the user. Noting that one user can only open one type of application at a time so that only one sensing process can be added. Each sensing process aims to collect the motion sensor data when opening the application of the same type. As shown in Figure 3, we will repeat the sensing process, including the following three steps for each time slot:

- **Step 1.** Collecting sensing data. The user collects

the sensing data for the sensing process. E.g., when a user opens WeChat, it means he triggers a chat sensing process. Meanwhile, ESPIALCOG will collect sensor data of chat type for 3 seconds.

- **Step 2.** Sending data and getting payoff. The user exchanges sensing data, payoff, and other statistics (the information of the population state and the system average payoff) with the server. E.g., the collected data of opening WeChat will be sent to the server. Then the user will receive the payoff which is calculated by the payoff functions in Algorithm 1.
- **Step 3.** Evolving statistics. The user decides whether to change his statistics or not based on the evolutionary stability participation mechanism. E.g., after sending sensing data to the server, the user will obtain the information of the population state and the system average payoff, which will advise the user whether to choose another sensing process (game, news) or stick to the current sensing process (chat). When the user's payoff is less than the system average payoff, the system will generate a random value A according to a uniform distribution on (0, 1). At the same time, each user choosing the sensing process has an evolving probability B to choose other sensing processes. If A is less then B, our system will advise the user to choose another sensing process, otherwise stick to the current sensing process. The user will learn and improve its strategy over time with the statistical information provided by the server.

Next, we give the related theorems, including the evolutionary dynamics of the evolutionary stable participation game, and the evolutionary equilibrium of the evolutionary stable participation algorithm:

Theorem 1. The evolutionary dynamics of the evolutionary stable participation game are given as

$$\dot{\theta}_i(t) = \alpha\left(\frac{S_i}{\sum_{i=1}^{P} S_i} - \theta_i(t)\right), \forall i \in \mathcal{P}.$$

where α is the strategy adaptation factor, and S_i is the budget for collecting sensing process of type i, $\theta_i(t)$ is the proportion of users choosing sensing process i at time t. For all t , we have $\sum_{i\in P}\theta_i(t) = 1$. Theorem 1 is the motivator for a user to decide its strategy profile when the current payoff is lower than the system average payoff. Details of the proof of Theorem 1 is in Appendix A.

According to Theorem 1, we then propose the evolutionary equilibrium in Theorem 2.

Theorem 2. The evolutionary stable participation algorithm converges to an evolutionary equilibrium, and it is globally asymptotically stable.

$$\theta^* = \left(\theta_i^* = \frac{S_i}{\sum_{j=1}^{P} S_j}\right), \forall i \in \mathcal{P}.$$

The proof of Theorem 2 is listed in Appendix B. The Evolutionary Stable Strategy (ESS) is an equilibrium refinement of the Nash Equilibrium, which naturally leads to the idea of evolutionary equilibrium [25, 26]. We prove that the ESS in Equation 2 is globally, asymptotically stable. It is an important characteristics since the evolutionary stable participation algorithm is thus robust to any degree of mutations of the users. Theorem 2 implies that the system eventually evolves to the evolutionary equilibrium θ_i^*.

The intuition behind Theorem 1 and Theorem 2. We discuss two factors that affect a user's choice in strategy evolvement. The first factor is the fraction of users participating in the sensing process. In the real world, a user observes the trend and follows it. The second factor is the "extra percentage of payoff" of a sensing process. It reflects the appealing of improving the utility of a sensing process. We assume that the server is responsible for collecting the statistics and computing the average. A user chooses one sensing process based on a probability distribution denoted by the product of the above two factors. At the end of each time slot, the user receives the payoff from the server, along with the statistics from the server. Then, the user decides whether to change his strategy or not. The basic idea is to let a user choose a better sensing process if his payoff is lower than the system-wide average payoff. The probability is based on two factors (i.e., the "extra percentage of payoff" of the sensing process, and the fraction of users choosing the sensing process). Theorem 1 shows the rate of strategy adaptation is governed by evolutionary dynamics. Theorem 2 implies that the system eventually evolves to the evolutionary equilibrium. The evolutionary stable participation mechanism converges to the equilibrium such that users choosing different sensing processes receive the same payoff. It is an important characteristic since the evolutionary stable participation algorithm is thus robust to any degree of mutations of the users. With the evolutionary stable participation algorithm, the system can quickly recover from the mutant states. This demonstrates that the algorithm is robust to the perturbations of the users. Considering that a user may change its choosing sensing process by following the popularity, it is important to ensure the stability of data collection in the crowdsourced sensing systems.

In reality, users are often bounded rational [24], and they do not always maximize their interests in the process of performing tasks, which means that even if the payoff of each participant is lower than the average, they may stick to the current strategy without changing the strategy. According to the above assumptions, we use a random probability parameter of ξ in the algorithm, and the participants will change the strategy under this certain probability.

The dynamics of user participation in our mechanism can be described with the evolutionary dynamics in Theorem 1. Based on it, we propose the evolutionary stable participation mechanism to guide the users in Algorithm 1. The mechanism is designed in a distributed and paralleled manner, and users can evolve their strategy simultaneously in each time slot. The server confirms the selected sensing process (i.e., the strategy) of all the users, exchanges sensing data and payoffs, and provides the necessary statistical information for the users. Combined with Algorithm 1, the user will learn and improve its strategy over time with the statistical information provided by the server.

Our system will converge into an evolutionary equilibrium in a short time, which is globally asymptotically stable. In other words, the data we collect has a high coverage rate: it not only guarantees the balance of data of different types

Algorithm 1 Evolutionary Stable Participation Algorithm

Require: The limited set of mobile device users $\mathcal{U}$,
The limited predefined set of sensing processes $\mathcal{P}$,
The payoff functions $\Phi_u\left(S_u, \theta\left(t\right)\right)$.
Initialization:
Set the global strategy adaptation factor $\alpha \in (0, 1]$.
Each user chooses a sensing process randomly.

for each user u and each time slot t **do**
  in parallel:
  collect the sensing data required by the sensing process P_u.
  send the collected sensing data to the server and receive the payoff $\Phi_u\left(S_u, \theta\left(t\right)\right)$.
  receive the information of the population state $\theta\left(t\right)$ and the system average payoff $\bar{\Phi}\left(\theta\left(t\right)\right)$.
  if $\frac{S_i}{\sum_{j=1}^{P} S_j} < \theta_i(t)$ **then**
    generate a random value ξ according to a uniform distribution on $(0, 1]$).
    if $\xi < \alpha\left(\frac{S_i}{\sum_{j=1}^{P} S_j} - \theta_i(t)\right)$ **then**
      select another sensing progress $p^{'}$ with evolving probability.
    else
      stick to the current sensing process.
    end if
  end if
end for

of applications, but also ensures that there is as much data of different applications in the same type.

2.4 Data De-noising

In the actual data collection process, there are many uncontrollable variables or factors that prevent the device from being uniformly and effectively calibrated. Therefore, there may be a certain deviation between the data collected by directly calling the sensor interface and the actual situation. It is important for us to de-noise the data with illegal semantics to improve the accuracy of the final model. Generally, we can divide illegal data semantics into two types: invalid data and abnormal data.

2.4.1 Invalid data

The first type is invalid data which can not represent the users' usage manner, and this part of data is not related to our authentication. When collecting data in an actual scenario, it is often impossible to guarantee that a user is always actually holding the phone during daily usage. For example, users can place the device on a horizontal desktop for interactive operations. In this case, even if the two collection requirements (The screen of the phone is on and a new application is running in the foreground) mentioned in Section 2.2 are met, the collected data still cannot effectively reflect the differences of usage patterns between different users. We asked 20 participants to handle a phone and put the phone on a stationary plane. Then we get the boundaries of the gravity sensor readings on three dimensions by minimizing the errors of device placement prediction: If the readings of gravity sensor meet $-1.5 < X_{gr}(k) < 1.5$, $-1.5 < Y_{gr}(k) < 1.5$ and $9 < |Z_{gr}(k)| < 10$ simultaneously, we regard it as invalid data and remove. Here, $X_{gr}(k)$, $Y_{gr}(k)$ and $Z_{gr}(k)$ represent values of the three axes of the gravity sensor x, y, and z at time k.

2.4.2 Abnormality

The second type is abnormality caused by abnormal mobile device sensors. Considering the low cost and portable nature of mobile devices, the readings of its built-in motion sensors are often affected by inherent factors such as materials and workmanship. In addition, the value of the motion sensor is also affected by external factors such as temperature, humidity, and age of use. These characteristics cause a certain deviation between the direct reading of mobile device motion sensors and the accurate value, we call them abnormalities. The abnormality will directly lead to the failure of the final authentication. There are three main types of abnormalities: Equal-Value abnormalities, Jump-Point abnormalities, and Zero-Value abnormalities. In the rest of this section, we will give the definition of these three types of abnormalities and the solution of how to handle them.

Equal-Value abnormalities. It is based on our observation that even if the device is not moving, the value collected from the motion sensor will change slightly between two adjacent samples when the sampling rate is 50HZ. But in some cases, low-performance motion sensors will cause Equal-Value abnormalities. We define it as an Equal-Value abnormality if two or more consecutive values are equal in the time series data. The ratio that the sum of the points with Equal-Value abnormalities (at least 2) E to the total number of points in the time series data N, called the Equal-Value abnormalities rate R_E. Here, $N = 150$ because the duration of data collecting is 3 seconds and the sampling rate is 50HZ.

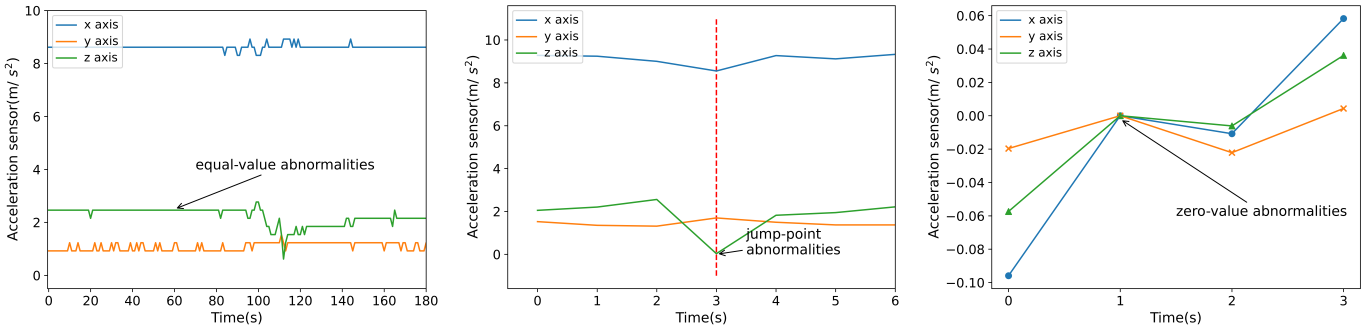
$$R_E = \frac{E}{N} \times 100\%.$$

Equal-Value abnormalities belong to machine anomalies and it should be removed. As shown in Figure 4(a), there are continuous Equal-Value abnormalities on z-axis. Continuous and constant values will undoubtedly interfere with the construction of user authentication models. In our experiment, if the R_E is greater than 0.7, we consider the corresponding time series data as abnormal and remove it. In Table 2, we list the distribution of 1513 users at different abnormal percentages. (We will introduce the dataset in Table 3 in Section 3). For example, the number '589' represents that among the 589 people, the percentage of abnormal data for each person is between 0 to 10%, which should be removed.

Jump-Point abnormalities. In the time series data, for any consecutive 3 points x_1, x_2 and x_3, if:

$$\begin{cases} (x_2 - x_1)(x_3 - x_2) < 0 \\ |x_2 - x_1| \geq a \cdot g \\ |x_3 - x_2| \geq a \cdot g \end{cases}, a \geq 10$$

then we call x_2 as a jumping point and this phenomenon as a Jump-Point abnormality. Here, g represents earth gravity. The ratio that the sum of the points with Jump-Point abnormalities J to the total number of points in the time series data N, called the Jump-Point abnormalities rate R_J.



(a) Equal-Value abnormalities (b) Jump-Point abnormalities (c) Zero-Value abnormalities

Fig. 4. Three types of abnormalities: an example.

$$R_J = \frac{J}{N} \times 100\%.$$

Jump-Point abnormalities describe the behavior of mobile device users and it should be retained. Through a lot of observations, we find that when the screen is tapped or shook, Jump-Point abnormalities will occur. As shown in Figure 4(b), the Jump-Point abnormalities can effectively reflect human participation and have a positive effect. However, there are still many values that do not conform to common sense. For example, the x-axis of the acceleration sensor might have reached more than 1 million. For the statistical characteristics of such data (when $a \geq 10$), this paper normalizes the data set into a fixed space to ensure consistency of data distribution.

Zero-Value abnormalities. The record of each axis of the sensor is 0, we call this phenomenon as Zero-Value abnormalities, as shown in Figure 4(c). Theoretically, this phenomenon can only happen when the mobile device is freely falling. From Figure 4(c), we find that the behavioral semantics around the Zero-Value abnormality is relatively continuous. Therefore, we remove the Zero-Value abnormality and stitch the data around it. From Table 2, we can see that it is similar to the Equal-Value abnormalities, Zero-Value abnormalities account for a small proportion in the entire data set. Removing the Zero-Value abnormalities helps to physically distinguish the states of users and is beneficial to improve the quality of the data.

2.5 Optimized LSTM with enhanced SGD algorithm

Unlike traditional machine learning methods, the LSTM network is well-suited to learn from time series for the potential contextual contents. This is one of the main reasons why LSTM outperforms alternative RNNs and Hidden Markov Models and other sequence learning methods in numerous applications [27]. Few works used LSTM for motion sensor-based user authentication, others such as [28, 29] utilized LSTM for wearable activity recognition, but the noisy labels in the complex environment and a large amount of real-world data were not considered by them. In this section, we will first introduce the framework of our optimized LSTM for mobile user authentication, and then we will discuss several important topics such as training set construction and enhanced SGD algorithm.

2.5.1 *The framework of optimized LSTM*

In Figure 5, we present an optimized LSTM framework for mobile device user authentication. The left side is a standard LSTM network structure which contains input layer (time series data), LSTM layer, classification layer, and output layer. The right side represents our enhanced stochastic gradient descent for our optimized LSTM training, including clipping, group, robustness factor, and tuning.

2.5.2 *Training set*

Binary classification is widely used in previous mobile device user authentication work [10, 21] because it can recognize the occasional overlapping among different users, while it can not be well addressed through one-class classification tasks. We can learn a universal network structure for all users [30, 31], but the network complexity and computational overhead will increase rapidly, we will discuss it in Section 4 later. Here, we choose a binary classification, which can better represent the different mobile device usage patterns of the device owner and others.

In the training phase, assuming that each user is profiled by n time series data. For p mobile device users, $n \times p$ samples are used to train the classifier in total. Treat the dataset of the authorized user as Class 1 and that of other users as Class 0. If p is large, our training set will be imbalanced. We adopt the stratified sampling heuristically to address this problem [32]. Firstly, we need to find a sort factor S which can mostly represent the characteristics of a user. With the idea of feature extraction in machine learning, magnitude is the most important feature for user authentication [10, 12, 13, 33], we define the sort factor S as follows:

$$S = \sqrt{X_a^2(k) + Y_a^2(k) + Z_a^2(k)}.$$

where $X_a(k)$, $Y_a(k)$ and $Z_a(k)$ represent values of the three axes of the acceleration sensor x, y, and z at time k. Secondly, we need to construct the training set. For each user, we sort all $n \times (p-1)$ samples and divide them into 5 equal size strata. Then, an equal amount of continuous samples are

TABLE 2
The distribution of 1513 users at different abnormal percentages.

The percentage of abnormal samples (3 seconds) for each user	0-10%	10%-20%	20%-30%	30%-40%	40%-50%	50%-60%	60%-70%	70%-80%	80%-90%	90%-100%
Equal-Value abnormalities	589	647	174	36	15	25	10	8	0	9
Jump-Point abnormalities	30	104	231	353	315	220	133	68	37	22
Zero-Value abnormalities	1323	37	25	19	12	14	17	22	22	22

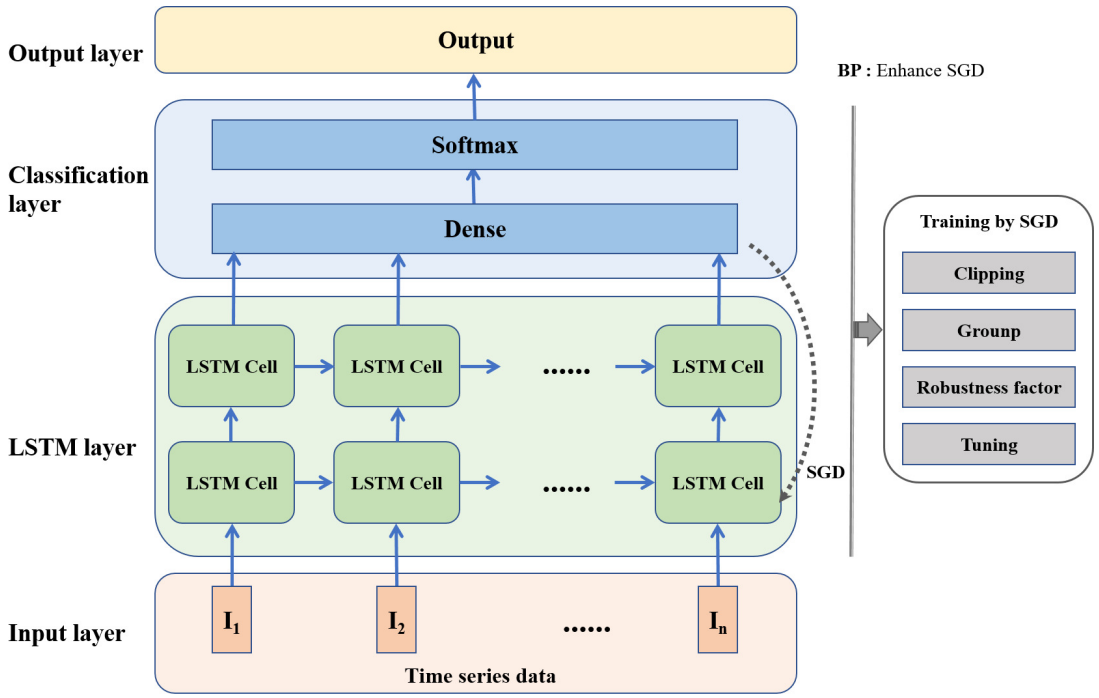


Fig. 5. Framework of optimized LSTM for mobile user authentication, and equipped with an enhanced stochastic gradient descent algorithm.

randomly drawn from each stratum. Noticing that the data samples from one stratum must be continuous because the temporal continuity of sensor reading is actually helpful to depict the authorized owner's pattern of handling the device. By doing so, negative samples including in the training set have better representativeness of the $p-1$ users. And the final model becomes more robust than simple random sampling. Actually, the ratio of the number of samples by the owner to that of other users is 1:5, which can be properly handled by deep learning algorithms. Moreover, for effective training, the number of positive instances usually needs to exceed 4,000. The above optimal point is chosen by conducting an experiment which has been proved in previous work [10].

2.5.3 Enhanced SGD Algorithm

In a noisy environment, the training set is not always pure. That means one may get incorrect labels from training set during the supervised learning. For example, The authorized device owner may share her/his phone to others, such as friends and family members. We have no idea of the label ground truth. Incorporating the noisy data into the LSTM model would affect the classification accuracy. To deal with this problem, one possible way is to check the training set and adjust it manually, but this is obviously unrealistic because the data collection is unobtrusive and automatically. Therefore, we inspire from previous work [34] and propose a more sophisticated and robust approach which aims to reduce the influence of noisy labels during the training phase, especially in the stochastic gradient descent (SGD) computation.

Algorithm 2 describes the details of our enhanced stochastic gradient descent method. We begin the algorithm with random parameters θ_0 and end with outputting the optimal parameters θ_T which can minimize the loss function $\mathcal{L}(\theta)$. Meanwhile, at each step of the SGD, we firstly calculate the gradient $\nabla_{\theta_t}\mathcal{L}\left(\theta_t, x_i\right)$ for a random subset which contains G samples, we then clip ℓ_2 norm after getting the single gradient, add robustness factors and calculate the average. Finally, we take a step in the opposite direction of this average robust gradient. Next, we will describe the details of the components used in our algorithm.

Noise clipping: To reduce the influence of noisy labels, Algorithm 2 needs to bound the influence of each individual samples. To that end, we clip each gradient in ℓ_2 norm, that is, we replace the original gradient vector $\mathbf{g}_t\left(x_i\right)$ with $\mathbf{g}_t\left(x_i\right) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$, and C is a clipping threshold which is used to control the gradient norm bound. After that, we get a new gradient vector as follows:

$$\overline{\mathbf{g}}_t\left(x_i\right) \leftarrow \mathbf{g}_t\left(x_i\right) / \max\left(1, \frac{\|\mathbf{g}_t\left(x_i\right)\|_2}{C}\right).$$

From the above equation, we can easily get that if $\|\mathbf{g}\|_2 \leq C$, $\overline{\mathbf{g}}_t\left(x_i\right) \leftarrow \mathbf{g}_t\left(x_i\right)$. But if $\|\mathbf{g}\|_2 > C$, the values of gradient will be scaled down. As a matter of fact, one can change

Algorithm 2 Enhanced Stochastic Gradient Descent

Input:
(1) Traning samples $\{x_1, \ldots, x_N\}$;
(2) Loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_i) = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log\left(f\left(x_i, \theta\right)\right)$;
(3) Learning rate η_t;
(4) Noise scale σ;
(5) Group size G;
(6) Gradient norm bound C;
Initialize θ_0 randomly
  for $t \in [T]$ **do**
    Select random samples into a group G_t with sampling probability G/N
    Calculate gradient
    For each $i \in G_t$, calculate $\mathbf{g}_t\left(x_i\right) \leftarrow \nabla_{\theta_t}\mathcal{L}\left(\theta_t, x_i\right)$
    Norm clipping
    $\overline{\mathbf{g}}_t\left(x_i\right) \leftarrow \mathbf{g}_t\left(x_i\right) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
    Add Robustness factors (Gaussian noise)
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{G}\left(\sum_i \left(\overline{\mathbf{g}}_t\left(x_i\right) + \mathcal{N}\left(0, \sigma_t^2\right)\right)\right)$
    Gradient Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
  end for
Output:
The final parameters θ_T.

the hyper-parameter C intelligently to correct the biased gradient direction.

Multi-layer parameters: In Algorithm 2, all the parameters in the deep neuron network are grouped into a single input θ, where $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$, w represents the weight and b represents the bias. Considering multi-layer neural networks, we deal with each layer separately. That is to say, one can customize the gradient norm bound C and noise scale σ in each layer. Also, in a more sophisticated way, the gradient norm bound and noise scale will dynamically change as the number of epochs increases. In our experiment, we use a constant setting for these two parameters because the performance and robustness of the deep learning model are good enough by using our proposed method.

Groups: Recall that in our noisy data clipping, Algorithm 2 calculates the gradient of the loss function $\mathcal{L}(\theta)$ by computing the gradient of the loss on a group of samples and taking the average. To further eliminate the effects of noisy labels, we introduce a new concept called group. We know that in deep learning, batch is a common way to prevent local minimum, while group consists of several batches. To reduce memory consumption, we set the batch size much smaller than the group size G, which is a hyper-parameter in our algorithm. We perform the computation in batches, then put several batches into a group for revising the gradient. As a matter of fact, the construction of batches and group is done by randomly shuffling the samples for efficiency. For ease of analysis, however, we assume that each group is formed by independently picking each sample with probability $q = G/N$, where N is the size of the input dataset.

Robustness factors: There is a long tradition of adding random weight noise in classical neural networks, and it has been under-explored in the optimization of modern deep architectures. We inspire from previous work [35] and consider a simple technique of adding time-dependent Gaussian noise to the gradient at each training step t:

$$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \left(\overline{\mathbf{g}}_t\left(x_i\right) + \mathcal{N}\left(0, \sigma_t^2\right)\right)\right).$$

Existing work [36] has indicated that adding annealed Gaussian noise by decaying the variance works better than using fixed Gaussian noise. We use the following schedule for most of our experiments:

$$\sigma_t^2 = \frac{\alpha}{(1+t)^{\gamma}},$$

where α is selected from $\{0.01, 0.3, 1.0\}$ and $\gamma = 0.55$. If the gradient noise at the beginning of training is high, the gradient can be away from 0 in the early stages.

2.5.4 Tuning in Noisy Environment

In order to balance the robustness, accuracy, and overall performance of our approach for multiple complex tasks, we tune the hyper-parameters in a noisy environment. Especially, in our experiments, we find that the accuracy of the deep learning model is more sensitive to the training hyper-parameters such as group size, learning rate, and dropout ratio. The experimental results will be discussed in Sec. 3.

2.5.5 Decision

Given a testing sample (time series data for 3 seconds), the trained classifier outputs the probability whether the owner is using the phone p, which is used to get the binary decision d as

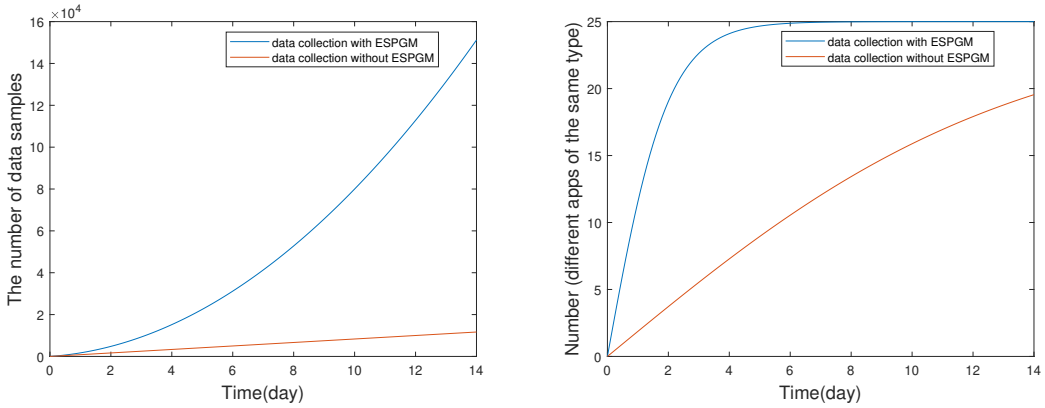
$$d = \begin{cases} 1, & \text{if } p > \theta \\ 0, & \text{else.} \end{cases}$$

Here, θ is the decision threshold from 0 to 1.

3 Evaluation

In our evaluation, we define the data collected in the experimental environment as pure and the data collected from the sophisticated environment will have noisy labels. We totally have three datasets, as shown in Table 3.

The composition and distribution of our dataset are the same as our previous work [10]. For the experimental data without noisy labels, we asked 20 participants to use the same phone for two weeks. Each participant generates 9240 effective samples to represent his/her usage manner (**dataset I**). For each user, we split the data samples into the training set and the test set. The ratio of the number of samples in the training set to that in the test set is 4:1 (the training data come from an earlier time than the testing data). In the training set, the ratio of the number of samples from the owner to that of other users is 1:5. The test set follows the same distribution. Moreover, we were able to obtain a labeled dataset provided by a big Internet company for our benchmarking and enhanced SGD algorithm test. That dataset was generated by 34 participants from the iOS platform (iPhone 7) (**dataset II**). Our third dataset is a large-scale raw dataset without ground truth, which was directly collected from the product by another Internet company with millions of users. All volunteers are employees within



(a) The efficiency of ESPGM (b) The coverage rate of ESPGM

Fig. 6. Performance of ESPGM on 6000 users, the efficiency of data collection and the coverage rate of application scenario improve greatly.

TABLE 3
The details of our datasets; all participants were skilled smartphone users with at least two years' experience; NL for Noisy Label.

Dataset	Participants	Age	Provider	Devices and Vendors	Duration	NL
Dataset I	20 individuals	20 - 60	Our laboratory	Samsung N9100 from Samsung Inc.	14 days	No
Dataset II	34 individuals	20 - 60	Internet company I	iPhone 7 from Apple Inc.	10 days	No
Dataset III	1513 individuals	20 - 60	Internet company II	Mi 3, Mi 4 and Redmi Note2 from Xiaomi Inc.	10 days	Yes

the company. For ethical considerations, we include the purpose of data collection in the user agreement. All the participants were informed of this study and they were given the option to opt in or opt out. Finally we collect data from 1,513 different users for 10 days (**dataset III**). For all the above datasets, the collection frequency is 50Hz. Each data collection phase lasts 3 seconds. IMEI is used as the user identifier.

To evaluate the model for user authentication, we define the following metrics:

True positive (TP): The authorized owner is correctly identified.

False positive (FP): Other users are incorrectly identified as the authorized owner.

False negative (FN): The authorized owner is incorrectly identified as other users.

True negative (TN): Other users are correctly identified.

Performance: The performance contains true positive rate $TPR = TP/(TP + FN)$, true negative rate $TNR = TN/(TN + FP)$ and overall accuracy $Accuracy = (TP + TN)/(TP + FP + FN + TN)$.

Overhead: Time latency of training of each user on the server side is to show our strengths compared with the state-of-the-art solutions.

3.1 Performance of ESPGM

In this section, we conduct simulations to evaluate the performance of our evolutionary stable participation game mechanism.

Assuming that there is a participatory sensing system with N users and P sensing processes, we set $N = 6000$ and $P = 3$ in our experiment. Noting that the proposed game model and mechanism can handle a much larger number of sensing processes. In our scenario, we have 3 sensing processes which have been described in Section 2. We assume each sensing process m have a budget $B_m = 200m$ for each time slot. At the start time, each user n randomly selects a sensing process $p \in P$. In each time slot, a user collects the required sensing data, exchanges the data and the payoff with the server, and then decides whether to change its strategy or not based on the proposed mechanism. We choose the strategy adaptation factor $\alpha = 0.5$ as the previous work [24] did. Also, we assume there are 25 different applications in one sensing process.

The simulation results are shown in Figure 6. From Figrue 6(a), we can see that by deploying ESPGM, our system can collect much more data in the same amount of time. In the first week, the number of data samples can reach 40000 with ESPGM, which is seven times as many as that without ESPGM. After two weeks, the number of data samples can reach 150000 with ESPGM while that of data samples without ESPGM is only 12000. Similarly, we can find that the coverage rate of users using different applications from the same type is also greatly improved in Figure 6(b). By deploying ESPGM, the coverage rate will reach 100% after five days. While the coverage rate is only 80% after two weeks without ESPGM.

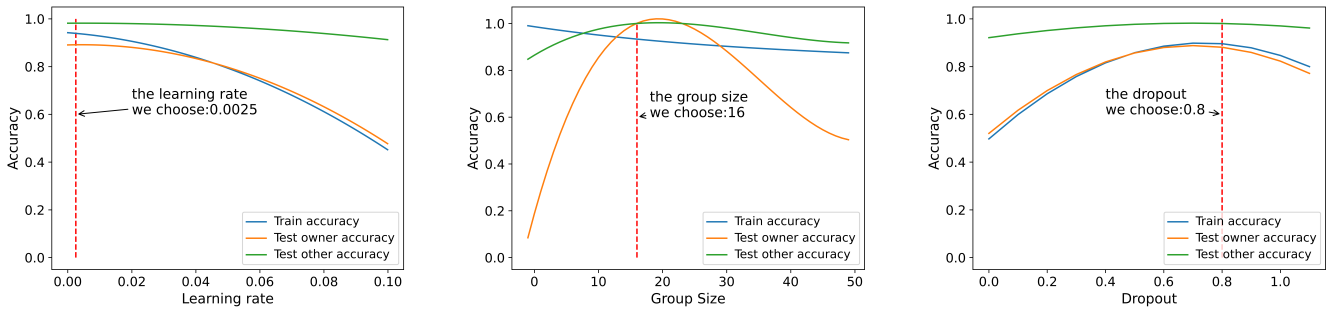
3.2 Performance on Noisy Dataset

In this section, we experiment to pick up the best parameters of our LSTM network. Moreover, we compare our method with the state-of-the-art work to show the overall performance of ESPIALCOG.

3.2.1 Parameter Optimization

Our LSTM network structure refers to the famous work in human activity recognition [37], due to the universality of the network structure in similar tasks [30], we reuse its number of layers (layer = 2) and the number of neurons per layer (neuron = 32). In order to balance the robustness,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3012491, IEEE Transactions on Mobile Computing



(a) Learning rate (b) Group size (c) Dropout ratio

Fig. 7. Model accuracy v.s. learning rate, group size and dropout ratio.

TABLE 4
The performance of EspialCog compared with RISKCOG [10] on our **Dataset III**.

Studies	State	TPR	TNR
RISKCOG	Steady	73.28%	98.43%
	Moving	81.41%	98.89%
EspialCog	Not distinguish	87.00%	97.93%

accuracy, and overall performance of our approach for multiple complex tasks, we tune the hyper-parameters in noisy environments based on the initial parameters in [37]. Especially, in our experiments, we find that the accuracy of deep learning model is more sensitive to the training hyper-parameters such as learning rate η_t, group size G, and dropout ratio ϵ. Other parameters, such as noise scale σ and gradient norm bound C, we use constant setting empirically which have been explained in Section 2.5.3.

We conduct the grid search to find the optimal configuration of the parameters η_t, G and ϵ in the 2-layer LSTM network structure, where the parameter η_t determines how fast the loss function changes, G is used to reduce the local minimum and revise the gradient, and ϵ is used to prevent the deep learning model from over-fitting. Especially, due to the group size G, we use mini-batch in our experiment.

Learning rate. If the learning rate is set too large, the model will fail to converge. Otherwise, the loss function may be trapped into a local minimum. We adjusted the learning rate η_t over a coarse range. The train accuracy, test owner accuracy (TPR), and test other accuracy (TNR) under different learning rates are shown in Figure 7(a). We change the range of the learning rate is from 0 to 1, the step size is 0.01. After comparison, we can find that the range of 0 to 0.01 performs better. We repeat the above procedure and set the step size 0.0005. Finally, we choose $\eta_t = 0.0025$.

Group size. Since we use mini-batch in our experiment, the value of group can refer to the original batch size. Generally using a power of 2 as the batch/group size can effectively use computing resources. When $\eta_t = 0.0025$, we choose the value of group size from 8, 16, 32, 64, 128, and 256. The train accuracy, TPR, and TNR under different group sizes are shown in Figure 7(b). We finally choose $G = 16$.

Dropout ratio. When $\eta_t = 0.0025$ and $G = 16$, we change the range of the dropout ratio from 0 to 1, the step size is 0.1. The train accuracy, TPR and TNR under different dropout ratio are shown in Figure 7(c). We finally choose $\epsilon = 0.8$.

3.2.2 *Accuracy in Noisy Dataset*

By deploying the best hyper-parameters, we achieve high accuracy on the large-scale dataset (**Dataset III**) from 1513 people. The final results are shown in Figure 8. In particular, the average values of training accuracy, TPR, and TNR are 88.50%, 87.00%, and 97.93%, respectively. The results show that EspialCog can identify non-owner users accurately (low false positive) with relatively stable availability (low false negative). As we know, false positive and false negative are mutually restrictive, decrease one of them may lead to the increment of another. In mobile device user authentication, we always pay attention to the sensitive information of the device owner will not be stolen by others, the low false positive rate can meet this kind of demand.

We also summarize the performance of EspialCog compared with the state-of-the-art work RISKCOG [10] on our **Dataset III**. As shown in Table 4, we can see that the TPR of our system is much higher than RISKCOG while the TNR is basically unchanged compared with RISKCOG. Note that RISKCOG used the SVM classifier and divided the motion state into steady state and moving state, but the TPR of both two motion states is worse than ours. The main reasons are that our context-based deep learning model is more suitable for this type of time series data collected from motion sensors, and our data de-noising method brings in a more clear dataset. Another reason is that we use an enhanced SGD algorithm which will be evaluated and discussed in Section 3.3.

3.3 Performance of Enhanced SGD Algorithm

In this section, we conduct the experiment by adding noisy labels to the pure dataset to prove that our enhanced SGD algorithm is highly robust in the noisy environment.

We test the accuracy of our enhanced SGD algorithm on **Dataset I** and **Dataset II** which have been described in Table 3. In the experiment, the network structure is a 2-layer LSTM network structure. The non-linear activation we used was ReLU and we used dropout with parameter 0.8. We trained the network using the Adam optimizer

TABLE 5
The overall accuracy of enhanced SGD algorithm compared with other related work on **Dataset I** and **Dataset II**. There are three different status: Original training data without any noise, training data with 10% noise and training data with 20% noise. The test data did not contain any noisy labels.

Studies	Dataset I			Dataset II		
	Original data	10% noise	20% noise	Original data	10% noise	20% noise
Baseline [37]	91.37%	88.74%	86.33%	94.39%	91.02%	90.35%
Semi-supervised learning [10]	90.23%	90.28%	88.78%	94.02%	93.35%	91.61%
Enhanced SGD algorithm	91.59%	91.07%	89.93%	94.51%	93.78%	92.14%

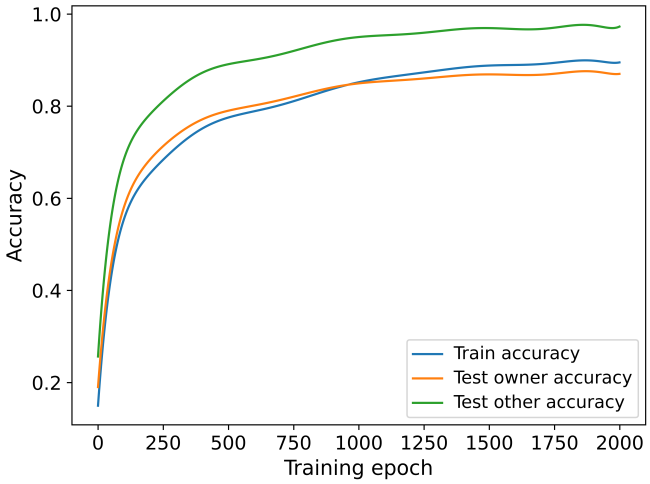


Fig. 8. The overall performamce of ESPIALCOG on our **Dataset III**.

[38] with default parameters. Regarding the configuration of ESPIALCOG, we set the learning rate η_t as 0.0025, noise scale σ as 0.01, group size G as 16 and gradient norm bound C as 100. These settings were kept fixed for all the experiments described below (except the semi-supervised learning algorithm [10]). We generated noisy data from clean data by randomly changing some of the labels followed by the existing work [34]. We converted each label with probability p to a different label according to a predefined permutation. The labels of the test data will not change anymore in order to validate and compare our method to the regular approach.

In our evaluations, we totally use three different methods (include our enhanced SGD algorithm). The baseline method is the 2-layer LSTM network structure described above [37]. We also compare our enhanced SGD algorithm with state-of-the-art solutions in the area of dealing with noisy labels, RISKCOG [10]. The results of totally test accuracy on **Dataset I** and **Dataset II** are shown in Table 5.

From Table 5 we can see that the LSTM-based methods except semi-supervised learning perform well by deploying the original training data. The reason is that the traditional machine learning-based classifier (SVM) can hardly handle the time series data, the contextual relationships contained in the data are not well extracted. In the 10% and 20% noise case, the weakness of the baseline LSTM method appears because there is no additional way for it to fight noisy labels. All in all, in the 10% noise and 20% noise case, our enhanced SGD algorithm works better and better along with the training epochs though it fluctuates and finally achieves the higher accuracy over other state-of-the-art methods.

TABLE 6
Average training latency; Iterations represent the number of average times needed to reach convergence; results are based on the data from 1,513 users (**Dataset III**).

Studies	Training for once	Iterations	Training until convergence
Semi-supervised learning [10]	102.74s	100	10274s
Enhanced SGD algorithm	3.13s	1500	4695s

3.4 Overhead

In this section, we first calculate the latency of the training phase by deploying different algorithms on the server. Then we measure the overhead on the client side.

3.4.1 *Overhead on The Server*

We set up the experiment on a server with an Intel Xeon E5 CPU and 64G memory running on Ubuntu 16.04 and the latency results of ESPIALCOG compared with the semi-supervised learning presented in [10] are listed in Table 6. On the one hand, for each training procedure, the average training latency for our enhanced SGD algorithm is 3.13s with CPU acceleration. If we use a GPU, we are optimistic that the training time for each training epoch will be within 1 second. While for semi-supervised learning proposed in [10], the average latency of each training procedure is 102.74s. On the other hand, the number of average times needed to reach convergence for these two methods is different, the enhanced SGD algorithm requires more average iterations (about 1500), but it has better performance on the average time for each user's model to reach convergence.

3.4.2 *Overhead on The Client*

On the client side, Tensorflow[2] is deployed on Android to finish the authentication task. We utilize the famous Android performance test tool Emmagee[3] to monitor battery consumption, CPU, and memory usage. Table 7 shows the results. For the battery consumption, we let one participant use the client app for three hours, which includes both data collection and offline authentication. The battery required by our application in one hour is less than 0.4%. The CPU and memory usage during offline authentication on three different smartphones is a bit more than that during offline authentication. This is logical because additional

2. Tensorflow. https://www.tensorflow.org/guide/saved_model
3. Emmagee. https://github.com/NetEase/Emmagee

TABLE 7
The overhead results on three different smartphones; the measurement of battery consumption lasts three hours.

		Data Collection		Offline Authentication	
Phone Type	Battery Consumption (mAh)	CPU (%)	Memory (MB)	CPU (%)	Memory (MB)
Samsung S20	105.3/4000	1.4	13.9	7.9	65.0
Vivo Xplay6	102.0/4080	2.3	14.5	6.5	50.2
MI8	114.1/3400	1.6	14.4	7.0	87.1

TABLE 8
Latency of offline authentication.

Procedure	Time (ms)
Data collection	3124.6
Data preprocessing	13.8
Decision	17.3
Overall	3155.7

computational tasks are generated when performing real-time authentication.

We also check the latency of offline authentication. We execute the procedures: data collection, data preprocessing (data de-noising and normalization), and decision for 1000 times on Vivo Xplay6, and record the average time for each step. The results are listed in Table 8. We can see the whole process can be finished within 3155.7 ms. The latency introduced by steps other than data collection is negligible. All in all, the client's overhead can fully meet the needs of real-world usage.

3.5 Resistance to Mimicry Attack

The safety of the sensor-based gait authentication has always been concerned by the industry community and the academic community. Some researches [39–41] have analyzed its ability to resist attacks, in which the majority is mimicry attack. For the mimicry attack, the attackers observe the user's usage manner and mimic the authentic user's gesture and operation.

To launch the mimicry attack, for the 20 individuals (the same individuals in Dataset I in our laboratory), we select the victim in turn: we first select one individual as the victim and one classifier is trained for the authorized device owner by fingerprinting the usage manner (Dataset I, each victim has 9240 samples for model training), then we asked the remaining 19 individuals to imitate the victim's pattern one by one (100 times for each person, and a participant generates 30 samples each time). Those samples are checked against the classifier, and we identify the percentage of samples which are correctly labeled as other users. Finally, ESPIALCOG blocks the attacks by imitation with probability over 98.40%, and this number is even higher than the TNR (97.93% in Table 4) of ESPIALCOG on the large dataset. One possible reason is that the contextual contents (e.g., the inherent usage pattern for the user owner) of the time series data were considered via the LSTM model, which is hard to be bypassed via imitation. Previous works [39–41] also raised the similar results.

4 Discussion

In this section, we discuss the strengths and weaknesses of EspialCog, and how to improve in future work.

4.1 The Strengths of EspialCog

Here, we discuss the strengths of our system from the following three aspects:

High data collection efficiency and sufficient scenario coverage rate. The proposed participation game theory can stimulate smart-phone users to actively participate in sensing processes by contributing sensing data. The data collecting process is controllable and the time required to obtain sufficient training data is much less than RISKCOG. Moreover, we propose an evolutionary stable mechanism to improve the coverage rate of various application scenarios for authentication, which is not well addressed in [10]. As shown in Figure 6, the efficiency of data collection and the coverage rate of application scenario improve greatly by deploying ESPGM.

Strong de-noise ability. We deploy three methods for data de-noising to further eliminate the noise impact of the hardware. The noise impact of the hardware is hardly considered by previous studies [11–22], while the existing de-noising technique [10] can remove the flat data, it's still thought to be one-sided. Table 2 shows all three abnormalities which were not well addressed by [10].

Good robustness and high accuracy. We propose an optimized LSTM method to solve the contextual issues of user behavior based on time series. Also, by deploying our enhanced SGD algorithm, the optimized LSTM can learn from data that has been diluted by a large amount of label noise. Table 4 and Table 5 show the overall performance of EspialCog and the performance of Enhanced SGD Algorithm compared with [10], respectively. Although the accuracy results seem only marginally better than other solutions, the potential improvements are obvious. Firstly, a semi-supervised online learning algorithm is proposed in [10] to address the noisy label issue, the model should be re-trained when the new data samples are uploaded, which can introduce a large number of computing costs. The LSTM in our system can learn incrementally with new group (Section 3.2.1). Secondly, we can see that the TPR of our system is much higher than RISKCOG while the TNR is basically unchanged compared with RISKCOG, which greatly improve the usability of our system.

4.2 The Weaknesses of EspialCog and Future Work

Firstly, we only theoretically prove that the evolutionary stable game mechanism stimulates user participation and thus greatly improves the efficiency of sensor data collection. At the same time, the coverage rate of different apps is increased by setting incentives for different apps in advance. Because of the marginal effect, users' decisions also depend on other smartphone users. Although the user's decision theoretically comes from the algorithm calculation of the

server, the user's decision will change with the influence of the environment due to the randomness in the wild. In the future, we will optimize the game method and consider the mutation factors in practical application.

Secondly, the training phase of our system has redundancy. That is to say, once a new user participates in, a new model must be retrained from a set of random parameters like the previous work [10], and it is a waste of time. In the future, we would like to utilize transfer learning to improve training efficiency. E.g., train a transferable deep neural network structure similar to ImageNet [31] on an existing large-scale data set.

5 Related Work

In this section, we review notable efforts done to mobile user authentication and compare with them in data collection efficiency, scenario coverage rate, de-noising ability, robustness of models to highlight the novelty of our approaches.

We design a heuristic data collection mechanism based on participation game theory to reach high data collection efficiency. To the best of our knowledge, the traditional way for collecting data is to invite fixed participants to provide sensor data under some specific scenarios, such as walking [11–13, 15, 17, 18], going up/down stairs [12], picking up mobile devices [14, 20] and touching the screen of mobile devices [19, 21, 22]. In their experiments, the complicated collection process will be repeated once a new user joins, which makes it unrealistic in the real-world scenario. Zhu et al. [10] investigated the usage manner of different people and proposed an unobtrusive and passive data collection mechanism in the wild to provide large-scale data. However, users have different habits/frequencies of using their mobile devices, i.e., some people may play with their smartphones several hours in one day, while some only check the smartphone less than one hour in one week. For the latter, it takes a long time to get a sufficient amount of effective training data. Given the rare limitations on the efficiency of data collection, EspialCog proposed a heuristic data collection incentive mechanism to stimulate smartphone users to actively participate in sensing processes by contributing sensing data. In a word, our system can ensure that all users provide a sufficient amount of data in a short period of time.

We propose an evolutionary stable mechanism to improve the coverage rate of various application scenarios for authentication, and it finally converges to an evolutionary equilibrium. Collecting data from motion sensor for user authentication has been exploited by many researchers. Derawi et al. [11] and Kwapisz et al. [12] made use of phone-based acceleration sensor to authenticate device users. Ren et al. [17] devised a user verification system leveraging the unique gait pattern derived from acceleration sensors to detect possible user spoofing in the mobile health care system. These approaches require that the devices are placed in specified body locations (e.g., bind the device to the hip). The single device location makes it hard to cover various authentication status and behavior patterns. Zhu et al. [10] developed a system called RISKCOG, which could collect data at the start of apps of different types and had a relatively higher coverage rate of user's behavior patterns. It is still insufficient because RISKCOG does not consider the user patterns of different applications under the same type (e.g., for the same type of chat applications, one may use DingTalk during office hours and use Instagram at spare time). Consider that the user's pattern dramatically varies with different types of applications, we need to collect a large amount of sensor data under different application scenarios to increase the authentication accuracy. In this article, we propose an evolutionary stable mechanism to improve the coverage rate of various application scenarios for authentication by collecting 1) data from different types of applications and 2) data from multiple applications under the same type.

We present a data de-noising method to remove Equal-Value abnormalities, Jump-Point abnormalities and Zero-Value abnormalities in the real-world scenario. Most of the previous user authentication studies [11–22] considered motion sensors ideally error-free during data collection and they had never took the noise impact of the hardware into account, which would lead to the difficulty in fitting the model and affect the prediction accuracy. Zhu et al. [10] observed that the flat data was ineffective to reflect the difference among various users' patterns in the data collection stage and removed them, but the analysis on the availability of the collected data was missed. Dey et al. [42] and Das et al. [43] utilized the hardware differences in motion sensor to fingerprinter mobile devices and the abnormalities in sensor readings are still not discussed. We analyze a lot of real data from 1,513 users and propose the data de-noising technique to remove the invalid data abnormalities such as Equal-Value abnormalities, Jump-Point abnormalities and Zero-Value abnormalities.

We implement an optimized LSTM method for time series data and propose an enhanced stochastic gradient descent (SGD) algorithm to improve the robustness of model against the noisy labels in the sophisticated environment. To the best of our knowledge, most of the existing work [11–14, 16–22] constantly collected sensory data and established corresponding models to authenticate users, they could not deal with the unlabeled data (noisy labels) in real-world environment. Lu et al. [15] handled the unlabeled data with an unsupervised learning algorithm which introduced high time latency. Moreover, the parameter adjustment of unsupervised clustering algorithm needs to take a great cost, and the generalization ability of parameters remains to be verified. Zhu et al. [10] designed a semi-supervised online learning algorithm with high accuracy and low latency to deal with the unlabeled data in a more sophisticated environment. However, the classification method used by them (binary-class SVM) is not suitable for time series data in complex scenarios, and it also cannot take into account the context of user behavior. Compared with the above work, we propose an optimized LSTM method to solve the contextual issues of user behavior based on time series.

To address the noisy label problem, several studies have investigated the impact of noisy labels on machine learning classifiers. Approaches to learn from noisy labels can generally be categorized into two groups: In the first group, existing approaches aim to propose noise elimination algorithms to get a clean dataset in the data pre-processing phase. Methods in this group frequently face the challenge of disambiguating between mislabeled and hard training

samples. In order to overcome this difficulty, people often use semi-supervised approaches by combining noisy data with a small set of clean labels [44]. Some approaches use unsupervised learning [45] and self-supervised learning [46, 47] to throw the noisy data aside. In the second group, existing methods propose some noise-robust algorithms to learn directly from noisy labels in the training phase [48–51]. Therein, the newest and representative approach is Pumpout [48]. Pumpout is aim to squeeze out the negative effects of noisy labels actively from the model being trained, instead of passively forgetting these effects. The realization of Pumpout is to train deep neural networks by stochastic gradient descent "fiting" labels, while train deep neural networks by scaled stochastic gradient ascent on "not-fitting" labels. Since the pattern which Pumpout uses is single, this method is hard to be controlled and adjusted in the real-world scenario. In order to reduce the impact of noisy labels during training, we designed an enhanced SGD algorithm, which greatly improved the robustness of the model. Our work differs from these approaches in that we not only consider the final accuracy, but also the intrinsic mechanisms and the scalability of the network structure. We study the behavior of standard neural network training procedures in settings with different proportions of label noise. By deploying our enhanced SGD algorithm, the optimized LSTM can learn from data that has been diluted by a large amount of label noise.

All in all, ESPIALCOG is able to improve the data collection efficiency, scenario coverage rate, de-noising ability and robustness of models for motion sensor-based authentication greatly, and it outperforms all the related methods.

6 CONCLUSION

In this paper, we present the system ESPIALCOG to authenticate the mobile device owner through optimized LSTM with an enhanced stochastic gradient descent algorithm. Regarding the issue of privacy-preserving in the context of social impact, our collected data only involves the insensitive motion sensors, which are commonly available on current devices.

Unlike previous studies with the problems of low data collection efficiency, insufficient authentication scenario coverage rate, weak de-noising ability and poor robustness of models, our system can collect the sensor data embedded in mobile devices self-adaptively, unobtrusively and efficiently through the Evolutionary Stable Participation Game mechanism with high scenario coverage rate, minimize noise from collected data by analyzing and removing three types of abnormalities, authenticate the ownership of mobile devices by adopting optimized LSTM model in real-time. The experiment of adding noise labels to pure data proves that our enhanced SGD algorithm is highly robust even in a noisy environment, and the experimental results on the large-scale dataset show that our system gains the classification accuracy values of 87.00% and 97.93% for the user owner and others, respectively. It can meet the security, privacy, and usability requirements jointly in mobile user authentication and surpass the state-of-the-art work.

ACKNOWLEDGMENTS

We would like to thank IEEE Transactions on Mobile Computing editors and reviewers for the review efforts.

REFERENCES

[1] "CCS Insight forecast: Wearables Momentum Continues," http://www.ccsinsight.com/press/company-news/2516-wearables-momentum-continues.

[2] F. Zoebisch and C. Vielhauer, "A test tool to support brute-force online and offline signature forgery tests on mobile devices," in *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, vol. 3. IEEE, 2003, pp. III–225.

[3] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *Woot*, vol. 10, pp. 1–7, 2010.

[4] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proc. ACM SOUPS*, 2011, p. 6.

[5] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proc. ACM SIGSAC*, 2012, pp. 113–124.

[6] A. Bianchi, Y. Fratantonio, A. Machiry, C. Kruegel, G. Vigna, S. P. H. Chung, and W. Lee, "Broken fingers: On the usage of the fingerprint api in android." in *NDSS*, 2018.

[7] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems*. IEEE, 2018, pp. 1–9.

[8] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1528–1540.

[9] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa, "Unravelling robustness of deep learning based face recognition against adversarial attacks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] T. Zhu, Z. Qu, H. Xu, J. Zhang, Z. Shao, Y. Chen, S. Prabhakar, and J. Yang, "Riskcog: Unobtrusive real-time user authentication on mobile devices in the wild," *IEEE Transactions on Mobile Computing*, 2019.

[11] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *Proc. IEEE IIH-MSP*, 2010, pp. 306–311.

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Proc. IEEE BTAS*, 2010, pp. 1–7.

[13] C. C. Ho, C. Eswaran, K.-W. Ng, and J.-Y. Leow, "An unobtrusive android person verification using accelerometer based gait," in *Proc. ACM MoMM*, 2012, pp. 271–274.

[14] J. Zhu, P. Wu, X. Wang, and J. Zhang, "Sensec: Mobile security through passive sensing," in *Proc. IEEE ICNC*, 2013, pp. 1128–1133.

[15] H. Lu, J. Huang, T. Saha, and L. Nachman, "Unobtrusive gait verification for mobile phones," in *Proc. ACM ISWC*, 2014, pp. 91–98.

[16] H. G. Kayacik, M. Just, L. Baillie, D. Aspinall, and N. Micallef, "Data driven authentication: On the effectiveness of user behaviour modelling with mobile device sensors," in *Proc. of the 3rd Workshop Mobile Secur. Technol.*, 2014.

[17] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE TMC*, vol. 14, no. 9, pp. 1961–1974, 2015.

[18] W.-H. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *Proc. IEEE ICISSP*, 2015, pp. 1–11.

[19] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE TIFS*, vol. 11, no. 5, pp. 877–892, 2016.

[20] W.-H. Lee, X. Liu, Y. Shen, H. Jin, and R. B. Lee, "Secure pick up: Implicit authentication when you start using the smartphone," in *Proc. ACM SACMAT*, 2017, pp. 67–78.

[21] A. Buriro, B. Crispo, and Y. Zhauniarovich, "Please hold on: Unobtrusive user authentication using smartphone's built-in sensors," in *Proc. IEEE ISBA*, 2017, pp. 1–8.

[22] C. Shen, Y. Li, Y. Chen, X. Guan, and R. A. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE TIFS*, vol. 13, no. 1, pp. 48–62, 2018.

[23] W. Xu, G. Lan, Q. Lin, S. Khalifa, N. Bergmann, M. Hassan, and W. Hu, "Keh-gait: Towards a mobile healthcare user authentication system by kinetic energy harvesting." in *NDSS*, 2017.

[24] Z. Weng, Q. Chen, and Y. Sun, "Evolutionary stable participation game of smartphones in crowdsourced sensing," *International Journal of Distributed Sensor Networks*, vol. 12, no. 9, p. 1550147716665516, 2016.

[25] J. M. Smith, *Evolution and the Theory of Games*. Cambridge university press, 1982.

[26] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Courier Corporation, 2012.

[27] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[28] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[29] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, p. 11, 2017.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[32] J. E. Trost, "Statistically nonrepresentative stratified sampling: A sampling technique for qualitative studies," *Qualitative sociology*, vol. 9, no. 1, pp. 54–57, 1986.

[33] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensors," in *IEEE consumer communications and networking conference*. IEEE, 2013, pp. 914–919.

[34] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[35] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.

[36] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 681–688.

[37] G. Chevalier, "Lstms for human activity recognition," 2016.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] D. Gafurov, E. Snekkenes, and T. E. Buvarp, "Robustness of biometric gait authentication against impersonation attack," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2006, pp. 479–488.

[40] D. Gafurov, E. Snekkenes, and P. Bours, "Spoof attacks on gait authentication system," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 491–502, 2007.

[41] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3209–3221, 2017.

[42] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable." in *Proc. ISOC NDSS*, 2014.

[43] A. Das, N. Borisov, and M. Caesar, "Tracking mobile web users through motion sensors: Attacks and defenses," in *Proc. ISOC NDSS*, 2016.

[44] X. J. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.

[45] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," *arXiv preprint arXiv:1112.6209*, 2011.

[46] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The curious robot: Learning visual representations via physical interactions," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–18.

[47] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[48] B. Han, G. Niu, J. Yao, X. Yu, M. Xu, I. Tsang, and M. Sugiyama, "Pumpout: A meta approach for robustly training deep neural networks with noisy labels," *arXiv preprint arXiv:1809.11008*, 2018.

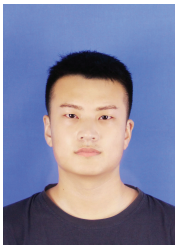
[49] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," *arXiv preprint arXiv:1712.05055*, 2017.

[50] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8527–8537.

[51] I. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2930–2939.

Tiantian Zhu received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2019. He is currently a lecturer with the college of computer science and technology, Zhejiang University of Technology, China. His research interests include mobile security, OSN security and artificial intelligence.

ZhengQiu Weng received the B.Eng. degree in computer science and technology from the Beijing Armored Force Engineering Institute, Beijing, China, in 2003, and the M.S. degree in software engineering from the Beijing Institute of Technology, Beijing,China, in 2005. She is currently pursuing the Ph.D. degree in computer science and technology with the Zhejiang University of Technology, and also an Associate Professor with the Wenzhou Polytechnic, Zhejiang, China. Her current research interests include networks security and big data technologies.



QiJie Song received the B.Eng. degree in electronic and engineering from Zhejiang Sci-tech University, Zhejiang, China, in 2016. He is currently pursuing M.Eng. degree in software engineering in Zhejiang University of Technology. His current research interests include mobile security, IoT and blockchain.

Yuan Chen received the B.E. degree in Internet of Things Engineering from Zhejiang University of Technology, Zhejiang, China, in 2017. She is currently pursuing M.S. degree for computer science in Zhejiang University of Technology. Her current research interests include mobile security, IoT and blockchain.

Qiang Liu received the Bacheleor of Science degree in electronic information engineering at Beijing Institute of Technology, Beijing, China, in 2018. He is currently a Ph.D. student in College of Computer Science of Zhejiang University, Hangzhou, China. His research interests include system security, software security and firmware analysis.

Yan Chen received the Ph.D. degree in computer science from the University of California, Berkeley, CA, USA, in 2003. He is a Professor with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. Based on Google Scholar, his papers have been cited over 7000 times and his h-index is 34. His research interests include network security, measurement, and diagnosis for large-scale networks and distributed systems. Prof. Chen won the Department of Energy (DoE) Early CAREER Award in 2005, the Department of Defense (DoD) Young Investigator Award in 2007, and the Best Paper nomination in ACM SIGCOMM 2010.

Mingqi Lv received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2012. He is currently an associated professor with the college of computer science and technology, Zhejiang University of Technology, China. His research interests include spatiotemporal data mining and ubiquitous computing.

Tieming Chen received the Ph.D. degree in computer software and theory from BeiHang University, Beijing, China, in 2011. He is now a professor with the college of computer science and technology, Zhejiang University of Technology, Hangzhou, China. He is also a member of IEEE and ACM. His research interests include cyberspace security and intelligence security.