# Sequential Keystroke Behavioral Biometrics for Mobile User Identification via Multi-view Deep Learning

Lichao Sun[1(✉)], Yuqi Wang[3], Bokai Cao[1], Philip S. Yu[1], Witawas Srisa-an[2], and Alex D. Leow[1]

[1] University of Illinois at Chicago, Chicago, IL 60607, USA
{lsun29,caobokai,psyu}@uic.edu, aleow@psych.uic.edu
[2] University of Nebraska–Lincoln, Lincoln, NE 68588, USA
witty@cse.unl.edu
[3] Hong Kong Polytechnic University, Kowloon, Hong Kong
csyqwang@comp.polyu.edu.hk

**Abstract.** With the rapid growth in smartphone usage, more organizations begin to focus on providing better services for mobile users. User identification can help these organizations to identify their customers and then cater services that have been customized for them. Currently, the use of cookies is the most common form to identify users. However, cookies are not easily transportable (e.g., when a user uses a different login account, cookies do not follow the user). This limitation motivates the need to use behavior biometric for user identification. In this paper, we propose DeepService, a new technique that can identify mobile users based on user's keystroke information captured by a special keyboard or web browser. Our evaluation results indicate that DeepService is highly accurate in identifying mobile users (over 93% accuracy). The technique is also efficient and only takes less than 1 ms to perform identification.

## 1 Introduction

Smart mobile devices are now an integral part of daily life; they are our main interface to cyber-world. We use them for on-line shopping, education, entertainment, and financial transactions. As such, it is not surprising that companies are working hard to improve their mobile services to gain competitive advantages. Accurately and non-intrusively identifying users across applications and devices is one of the building blocks for better mobile experiences, since not only companies can attract users based on their characteristics from various perspectives, but also users can enjoy the personalized services without much effort [15].

User identification is a fundamental, but yet an open problem in mobile computing. Traditional approaches resort to user account information or browsing history. However, such information can pose security and privacy risks, and it is not robust as can be easily changed, e.g., the user changes to a new device or using a different application. Monitoring biometric information including a


Y. Altun et al. (Eds.): ECML PKDD 2017, Part III, LNAI 10536, pp. 228–240, 2017.
https://doi.org/10.1007/978-3-319-71273-4_19

user's typing behaviors tends to produce consistent results over time while being less disruptive to user's experience. Furthermore, there are different kinds of sensors on mobile devices, meaning rich biometric information of users can be simultaneously collected. Thus, monitoring biometric information appears to be quite promising for mobile user identification.

To date, only a few studies have utilized biometric information for mobile user identification on web browser [1,19]. Important questions such as what kind of biometric information can be used, how does one capture user characteristics from the biometric, and what accuracy of the mobile user identification can be achieved, are largely unexplored. Although there are some researches on mobile user authentication through biometrics [8,20], authentication is a simplified version of identification, and directly employing authentication would either be infeasible or lead to low accuracy. This work focuses on mobile user identification, and could also be applied to authentication.

In this paper, we collect information from basic keystroke and the accelerometer on the phone, and then propose DeepService, a multi-view deep learning method, to utilize this information. To the best of our knowledge, this is the first time multi-view deep learning is applied to mobile user identification. Through several empirical experiments, we showed that the proposed method is able to capture the user characteristics and identify users with high accuracy.

Our contributions are summarized as follows.

1. We propose DeepService, a multi-view deep learning method, to utilize easy to collect user biometrics for accurate user identification.
2. We conduct several experiments to demonstrate the effectiveness and superiority of the proposed method against various baseline methods.
3. We give several analyses and insights through the experiments.

The rest of this paper is organized as follows. Section 2 provides background information on deep learning, and reviews prior research efforts related to this work. Section 3 introduces DeepService and describes the design and implementation details. Section 4 reports the results of our empirical evaluation on the performance of DeepService with respect to other learning techniques. The last section concludes this work and discusses future work.

## 2 Background and Related Work

In this section, we provide additional background information on deep learning structure, and review prior research efforts related to our proposed work.

### 2.1 Background on Deep Learning Structure

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data, which is also called deep structured learning, deep neural network learning or deep machine learning. Deep learning is a concept and a framework instead of a particular method.

There are two main branches in deep learning: Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN). CNN is frequently used in computer vision areas and RNN is applied to solve sequential problems such as nature language process. The simplest form of an RNN is shown as follows:

$$h_k = \phi(Wx_k + Uh_{k-1})$$

where $h_k$ is the hidden state, $W$ and $U$ are parameters need be learned, and $\phi(.)$ is the is a nonlinear transformation function such as tanh or sigmoid.

Long Short Term Memory network (LSTM) is a special case of RNN, capable of learning long-term dependencies [13]. Specifically, RNN only captures the relationship between recent keystroke information and uses it for prediction. LSTM, on the other hand, can capture long-term dependencies. Consider trying to predict the tapping information in the following text "I plan to visit China ... I need find a place to get some Chinese currency". The word "Chinese" is relevant with respect to the word "China", but the distance between these two words is long. To capture information of the long-term dependencies, we need to use LSTM instead of the standard RNN model.

While LSTM can be effective, it is a complex deep learning structure that can result in high overhead. Gated Recurrent Unit (GRU) is a special case of LSTM but with simpler structures (e.g., using less parameters) [6]. In many problem domains including ours, GRU can produce similar results to LSTM. In some cases, it can even produce better results than LSTM. In this work, we implemented Gated Recurrent Unit (GRU).

Also note that with GRU, it is quite straightforward for each unit to remember the existence of a specific pattern in the input stream over a long series of time steps comparing to LSTM. Any information and patterns will be overwritten by update gate due to its importance.

We can build single-view single-task deep learning model by using GRU as shown in Fig. 2(b). We choose any one view of the dataset such as the view of alphabet as used in this study. We use the normalized dataset as the input of GRU. GRU will produce a final output vector which can help us to do user Identification. A typical GRU is formulated as:

$$\begin{aligned} z_t &= \sigma_g(W_z x_t + U_z h_{t-1}) \\ r_t &= \sigma_g(W_r x_t + U_r h_{t-1}) \\ \tilde{h}_t &= tanh(W x_t + U(r_t \odot h_{t-1})) \\ h_t &= z_t \tilde{h}_t + (1 - z_t) h_{t-1} \end{aligned}$$

where $\odot$ is an element-wise multiplication. $\sigma_g$ is the sigmoid and equals $1/(1 + e^{-x})$. $z_t$ is the update gate which decides how much the unit updates its activation or content. $r_t$ is reset gate of GRU, allowing it to forget the previously computed state.

In Sect. 3, we extend the single-view technique to develop DeepService, a multi-view multi-class framework.

### 2.2 Related Work

Most previous works focus on user authorization, rather than identification, based on biometrics. For example, there are multiple approaches to get physiological information that include facial features and iris features [7,9]. This physiological information can also be used for identification, but it requires extra permission from the users. Our method uses behavioral biometrics to identify the users without any cookies or other personal information.

Recently, more research work on continuous authorization problem has emerged for mobile users. Some prior efforts also focus on studying touchscreen gestures [20] or behavioral biometric behaviors such as reading, walking, driving and tapping [3,4]. There are research efforts focusing on offering better security services for mobile users. However, their security models have to be installed on the mobile devices. They then perform binary classifications to detect unauthorized behaviors. Our work focuses on building a general user identification model, which can also be deployed on the web, local devices or even network routers. Our work also focuses on improving users' experience through customized services including providing recommendations and relevant advertisements.

Recently, some research groups focus on mobile user identification based on web browsing information [1,19]. Abramson and Gore try to identify users' web browsing behaviors by analyzing cookies information. However, our model has been designed to target harder problems without using trail of information such as cookies or browsing history. We, instead, use behavioral biometrics to identify users. Information needed can be easily collected from web browser using Javascript.

## 3 DeepService: A Multi-view Multi-class Framework for User Identification on Mobile Devices

DeepService is a multi-view and multi-class identification framework via a deep structure. It contains three main steps to identify each user from several users. This process is shown in Fig. 1 and summarized below:

1. In the first step, we collect sequential tapping information and accelerometer information from 40 volunteers who have used our provided smartphones for 8 weeks. We retrieve such sequential data in a real-time manner.
2. In the second step, we prepare the collected information as multi-view data for the problem of user identification.
3. In the third step, we model the multi-view data via a deep structure to perform multi-class learning.
4. In the last step, we compare the performance of the proposed approach with the traditional machine learning techniques for multi-class identification such as support vector machine and random forest. This step is discussed in Sect. 4.

Next, we describe each of the first three steps in turn.

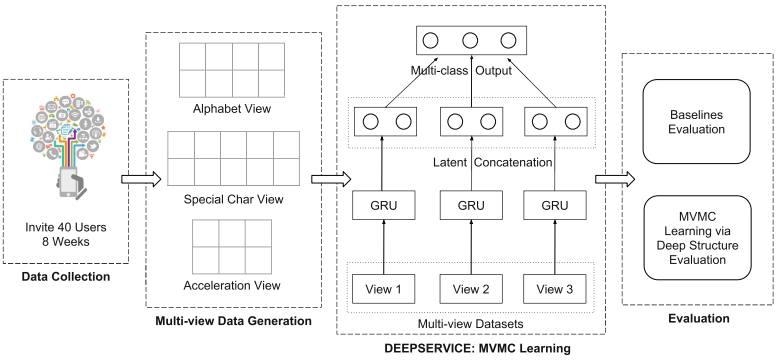


**Fig. 1.** Framework of DeepService

### 3.1 Data Collection

First, we describe the data collection process. Our study involves 40 volunteers. The main selection criterion is based on their prior experience with using smartphones. All selected candidates have used smartphones for at least 11 years (some have used smartphones for 18 years). In terms of age, the youngest participant is 30 years old and the oldest one is 63 years old.

Each volunteer was given a same smartphone with the custom software keyboard. Out of the 40 volunteers, we find that 26 of them (17 females and 9 males) have used the provided phones at least 20 times in 8 weeks. The data generated by these 26 volunteers is the one we ended up using in this study. The most active participant has used the phone 4702 times while the least active participant has used the phone only 29 times.

### 3.2 Data Processing

When users type on the smartphone keyboard either locally or on a web browser, our custom keyboard would collect the meta-information associated with the users' typing behaviors, including duration of a keystroke, time since last keystroke, and distance from last keystroke, as well as the accelerometer values along three directions. Due to privacy concerns, the actual characters typed by users are not collected. However, we do collect the categorical information of each keystroke, *e.g.*, alphanumeric characters, special characters, space, backspace. Note that such information can easily be collected from web browser using Javascript as well.

In the data collection process, there are inevitable missing data. For example, when the first time a user uses the phone, the feature *time_since_last_key* is undefined. We replace these missing values with 0. After the complement of missing values, we normalize all the features to the range of $[0, 1]$.

A typical usage of keyboard would likely result in a session consisting more than one keystroke. For example, a simple message such as *"How are you?"*

involves sequential keystrokes as well as multiple types of inputs (alphabets and special characters). In this study, one instance represents one usage session of the phone by the user. A session instance $s_{ij}$ represents the $j$-th session of the $i$-th user in the data set, which consists of three different types of sequential data. Let's denote $s_{ij} = \{c_{ij}^{(1)}, c_{ij}^{(2)}, c_{ij}^{(3)}\}$ where $c_{ij}^{(1)}$ is the time series of alphabet keystrokes, $c_{ij}^{(2)}$ is the time series of special character keystrokes, and $c_{ij}^{(3)}$ is the time series of accelerometer values. It is difficult to align the sequential features in different views because of different timestamps and sampling rates. For example, accelerometer values are much denser than special character keystrokes. Therefore, it is intuitive to treat $c_{ij}^{(1)}$, $c_{ij}^{(2)}$, and $c_{ij}^{(3)}$ as multi-view time series that together compose the complementary information for user identification.

### 3.3 Multi-view Multi-class Deep Learning (MVMC)

Now, we discuss the approach to apply deep learning for constructing the user identification model. The approach is based on Multi-view Multi-class (MVMC) learning with a deep structure.

As mentioned previously, we employ three different views. Each view $V_i$ contains different number of features and different number of samples. In Fig. 2(a), we can use fusion method to combine datasets of different views. However, due to the different number of features, and the number of records in each view of each session, it is hard to build a single-view dataset from many other views. Hence, instead of concatenating different views into one view, we choose to use them separately. This is done to avoid losing information as in the case when multiple views are combined to create a view. One major information that we want to preserve is the sequence of keystrokes. By using multi-view, we are able to maintain each view separately but then use multiple views to make predictions [5,17]. Recently, various methods have been proposed for this purpose [10–12].

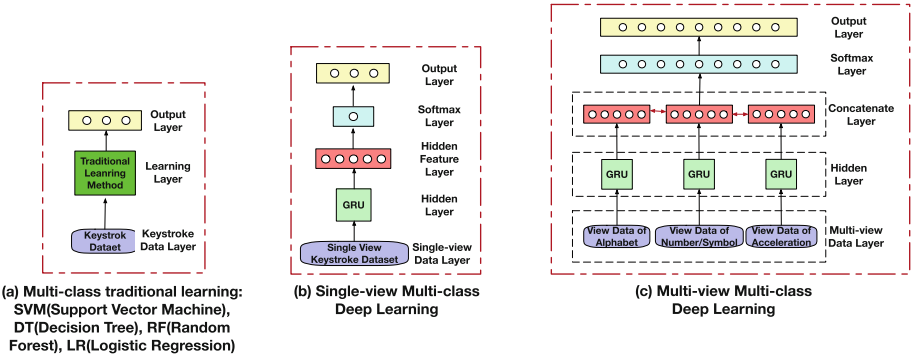


**Fig. 2.** A comparison of different frameworks of learning models: left: (a) traditional learning methods; middle (b) single-view multi-class; (c) multi-view multi-class

Before we generate multi-view multi-class learning, we first create single-view multi-class learning as shown in Fig. 2(b) (previously discussed in Sect. 2). Through that model, we can prove the multi-view can help us to improve the performance for identification and we can determine which view most contributes to the identification process. First, we separate the data set into multiple views. In this case, we have a view of alphabets, a view of numbers and symbols, and a view of tapping acceleration. Then, we use GRU and Bidirectional Recurrent Neural Network (we refer to the combination of these two approaches as GRU-BRNN) to build the hidden layers for each view. In the output layer, we use softmax function to perform multiple classifications. Finally, we can evaluate the performance on each view.

We describe the framework that uses multi-view and multi-class learning with deep structure in Fig. 2(c). Comparing to single-view multi-class, multi-view multi-class is a more general model. After we use GRU-BRNN to build the hidden layers for each view. We then concatenate the last layer information from each GRU-BRNN model of each view. We use the last concatenated layer, which contains all information from different views, for identification.

As GRU extracts a latent feature representation out of each time series, the notions of sequence length and sampling time points are removed from the latent space. This avoids the problem of dealing directly with the heterogeneity of the time series from each view. The difference between multi-view multi-class and single-view multi-class learning is that we use the multiple views of the dataset and we use the latent information of each view for prediction, which can improve performance over using only a single-view dataset. We can consider single-view multi-class as a special case of multi-view multi-class learning.

Note that in deep learning, different optimization functions can greatly influence the training speed and the final performance. There are several optimizers such as RMSprop and Adam [16]. In this work, we use an improved version of Adam called Nesterov Adam (Nadam) which is a RMSprop with Nesterov momentum.

## 4 Experiment

To examine the performance of the proposed DeepService on identifying mobile users. Our experiments were done on a large-scale real-world data set. We also compared the results with those from several state-of-the-art shallow machine learning methods. In this section, we describe how we conducted our experiments. We then present the experimental results and analysis.

### 4.1 Baselines: Keystroke-Based Behavior Biometric Methods for Continuous Identification

In previous work on keystroke-based continuous identification with machine learning techniques [2,4,14], Support Vector Machine, Decision Tree, and Random Forest are widely used for continuous identification.

*Logistic Regression (LR)*: LR is a linear model with sigmoid function for classification. It is an efficient algorithm which can handle both dense and sparse input.

*Linear Support Vector Machine (LSVM)*: LSVM is widely used in many previous authorization and identification works [2,4,14]. LSVM is a linear model that finds the best hyperplane by maximizing the margin between multiple classes.

*Random Forest/Decision Tree*: Other learning methods such as Random Forest and Decision Tree have not yet been adopted in many behavior biometric work for continuous identification by keystrokes information only. However, Decision Tree is a interpretable classification model for binary classification. It is a tree structure, and features form patterns are nodes in the tree. Random Forest is an ensemble learning method for classification that builds many decision trees during training and combines their outputs for the final prediction. Previous work [18] shows that tree structure methods can work more efficiently than SVM, and can do better binary classification comparing to SVM. We use different traditional learning methods on our data set as the baselines.

### 4.2 Evaluation of DeepService Framework

Since the number of session usage for every user is different, we use four performance measures to evaluate unbalanced results: Recall, Precision F1 Score (F-Measure), and Accuracy. They are defined as:

$$Recall = \frac{TP}{TP + FN} \qquad F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Precision = \frac{TN}{TN + FP} \qquad Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Next, we report the performance of DeepService using our data set. We measured precision, recall and accuracy, f1 score (f-measure) for different models. Based on the results, we can make the following conclusions.

- DeepService can identify between two known people at almost 100% accuracy in our experiment. This proves that our data set contains valuable biometric information to distinguish and identify users.
- DeepService can do identification with only acceleration records, even when user is not using the keyboard.
- DeepService is effective at identifying a large number of users simultaneously either locally or on a web browser.

### 4.3 User Pattern Analysis

In this section, we evaluate the feature patterns for different users. In Fig. 3, we shows the feature patterns analysis of top 5 active users in multi-view.

In the view of Alphabet graphs, each user tends to have unique patterns with respect to the duration, the time since last key, and the number of keystrokes in

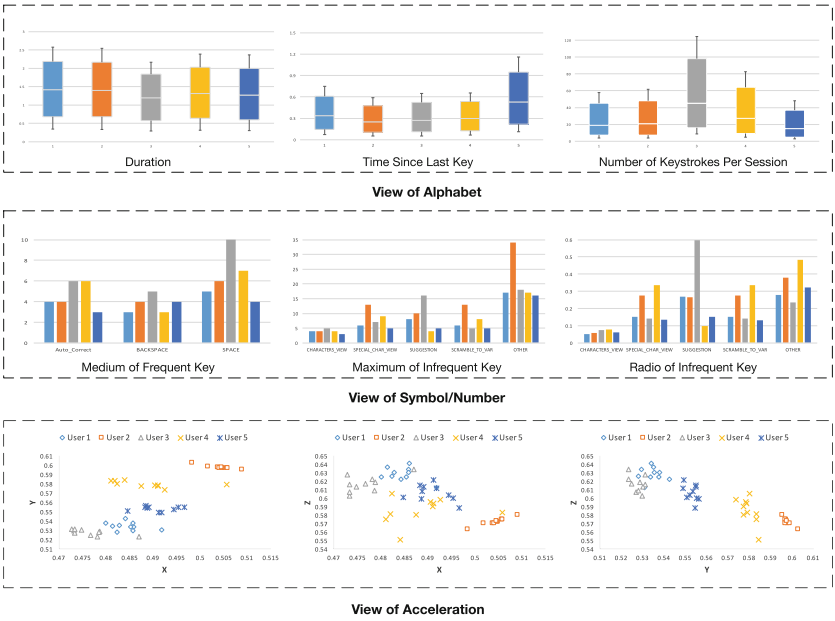


**Fig. 3.** Multi-view pattern analysis of top 5 active user: left is *user1* and right is *user5*

each session. For example, *user3* prefers to use more keystroke in every session with quicker tapping speed than other users.

In the view of Symbol/Number graphs, we have 8 different features. We separate these features into two groups: frequent keys and infrequent keys. A frequent key is defined as a key that is used more than twice per session, otherwise the key is an infrequent keys. (A user tends to use infrequent feature once per session.). We show the medium number of keystroke per session of frequent keys such as auto correct, backspace and space. We also show the range and radio of infrequent keys per session of the top five active users. For example, *user4* frequently uses auto_correct, but she infrequently uses backspace.

In view of Acceleration graphs, we show the correlation of different directions of acceleration. From the last graph, we find that the top 5 active users can be well separated, which proves acceleration can help to identify the users well. It is also proved in our experiments. In both user pattern analysis and experiment, we find view of acceleration can do better identification than other two views.

### 4.4 Identifying Users

DeepService can also perform continuous identification. Before we expand to multi-class identification, we first implement a binary-class identification based on multi-view deep learning which is a special case of MVMC learning identification.

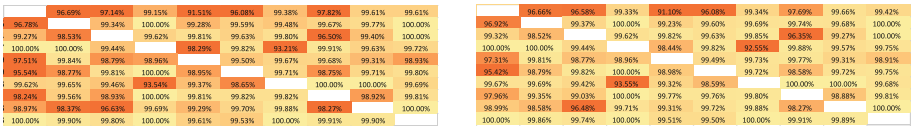

**Fig. 4.** Heatmap of multi-view binary-class identification: left is F1 score; right is accuracy

In Fig. 4, we can see that DeepService can do well identification between any two users with 98.97% f1 score and 99.1% accuracy in average. For example, private smartphone usually would not be shared by different people. However, sometimes the private phone could be shared between two people, such as husband and wife. DeepService can well separate any two people in this case.

For more general scenarios, we expand binary-identification to N active class (user) identification and use MVMC learning to figure out who is using the phone either locally or on a web browser. Table 1 reports our results.

If we increase the total number of the users in our model, it means we want to identify more people at the same time. For example, if our model is used on a home-router, it may need to identify only members of a family (3 to 10 people) at once. If we, instead, want to identify people working in a small office, we may need to identify more than 10 users. However, it is possible that a larger number of users would degrade the average performance of user identification. This is due to more variation of shared biometric patterns that introduce ambiguity into the system. That's the main reason we want to use multi-view data for user identification, since different users are unlikely to share similar pattens across all views.

**Table 1.** Results of DeepService and Baselines

| Method | 5 | | 10 | | 26 | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| LR | 66.88% | 66.85% | 44.25% | 45.31% | 27.44% | 30.26% |
| SVM | 68.18% | 68.13% | 44.39% | 45.12% | 30.33% | 31.90% |
| Decision Tree | 68.21% | 67.50% | 53.50% | 52.85% | 43.37% | 42.42% |
| RandomForest | 87.59% | 87.42% | 77.05% | 76.59% | 67.87% | 66.31% |
| Deep Single View | 82.64% | 82.48% | 78.27% | 78.33% | 61.26% | 63.11% |
| DeepService | **93.50**% | **93.51**% | **87.35**% | **87.69**% | **82.73**% | **83.25**% |

Table 1 and Fig. 5 report accuracy and F1 values of all learning techniques investigated in this paper. As shown, DeepService can identify a user without any cookies and account information. Instead, it simply uses the user's sequential keystroke and accelerometer information. Our approach (DS as shown in Fig. 5) consistently outperforms other approaches listed in Table 1. Moreover, as we increase the number of users, the performance (accuracy and F1) degrades less than those of other approaches.

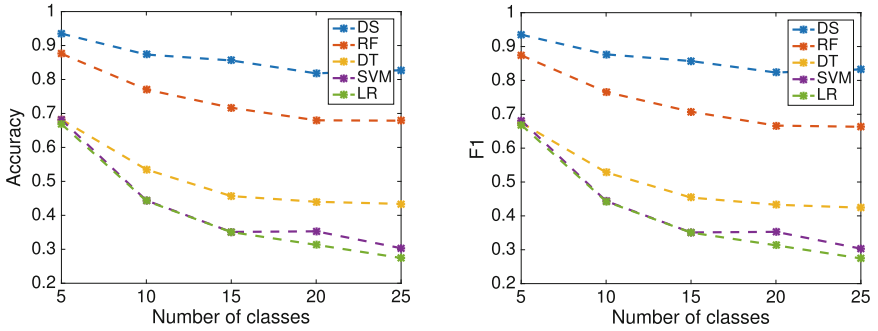


**Fig. 5.** Results with incremental number of classes (users)

In our experiments, we also experimented with using a single-view with deep learning model, and found that the accelerometer view do better identification than other two views. However, when we used information from all three views with MVMC learning, we achieved best performance when compared against the results of other baseline approaches.

### 4.5 Efficiency

To evaluate efficiency of our system, we employ a 15″ Macbook Pro 15 with 2.5 GHz Intel Core i7 and 16 GB of 1600 MHz DDR3 memory, and NVIDIA GeForce GT 750M with 2 GB of video memory. DeepService is not the fastest model (decision tree is faster), but it only takes about 0.657 ms per session which shows its feasibility of real-world usage.

## 5 Conclusion and Future Work

We have shown that DeepService can be used effectively to identify multiple users. Even though we only use the accelerometer in this work, our results show that more views of dataset can improve the identification performance.

In the future, we want to implement DeepService as a tool to help company or government to identify their customers more accurately in the real life. The tool can be implemented on the web or the router. Meanwhile, we will incorporate more sensors, which can be activated from the web browser, to further increase the capability and performance of the DeepService.

**Acknowledgements.** This work is supported in part by NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313.

## References


1. Abramson, M., Gore, S.: Associative patterns of web browsing behavior. In: 2013 AAAI Fall Symposium Series (2013)
2. Alghamdi, S.J., Elrefaei, L.A.: Dynamic user verification using touch keystroke based on medians vector proximity. In: 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp. 121–126. IEEE (2015)
3. Bo, C., Jian, X., Li, X.-Y., Mao, X., Wang, Y., Li, F.: You're driving and texting: detecting drivers using personal smart phones by leveraging inertial sensors. In: Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, pp. 199–202. ACM (2013)
4. Bo, C., Zhang, L., Jung, T., Han, J., Li, X.-Y., Wang, Y.: Continuous user identification via touch and movement behavioral biometrics. In: 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), pp. 1–8. IEEE (2014)
5. Cao, B., He, L., Wei, X., Xing, M., Yu, P.S., Klumpp, H., Leow, A.D.: t-BNE: tensor-based brain network embedding. SIAM (2017)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
7. de Martin-Roche, D., Sanchez-Avila, C., Sanchez-Reillo, R.: Iris recognition for biometric identification using dyadic wavelet transform zero-crossing. In: 2001 IEEE 35th International Carnahan Conference on Security Technology, pp. 272–277. IEEE (2001)
8. Feng, T., Liu, Z., Kwon, K.-A., Shi, W., Carbunar, B., Jiang, Y., Nguyen, N.: Continuous mobile authentication using touchscreen gestures. In: 2012 IEEE Conference on Technologies for Homeland Security (HST), pp. 451–456. IEEE (2012)
9. Goh, A., Ngo, D.C.L.: Computation of cryptographic keys from face biometrics. In: Lioy, A., Mazzocchi, D. (eds.) CMS 2003. LNCS, vol. 2828, pp. 1–13. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45184-6_1
10. He, L., Kong, X., Yu, P.S., Yang, X., Ragin, A.B., Hao, Z.: Dusk: a dual structure-preserving kernel for supervised tensor learning with applications to neuroimages. In: Proceedings of the 2014 SIAM International Conference on Data Mining, pp. 127–135. SIAM (2014)
11. He, L., Lu, C.-T., Ding, H., Wang, S., Shen, L., Yu, P.S., Ragin, A.B.: Multi-way multi-level kernel modeling for neuroimaging classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
12. He, L., Lu, C.-T., Ma, G., Wang, S., Shen, L., Yu, P.S., Ragin, A.B.: Kernelized support tensor machines. In: Proceedings of the 34th International Conference on Machine Learning (2017)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Miluzzo, E., Varshavsky, A., Balakrishnan, S., Choudhury, R.R.: TapPrints: your finger taps have fingerprints. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 323–336. ACM (2012)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)

16. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
17. Shao, W., He, L., Yu, P.S.: Clustering on multi-source incomplete data via tensor modeling and factorization. In: Cao, T., Lim, E.-P., Zhou, Z.-H., Ho, T.-B., Cheung, D., Motoda, H. (eds.) PAKDD 2015. LNCS (LNAI), vol. 9078, pp. 485–497. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18032-8_38
18. Sun, L., Li, Z., Yan, Q., Srisa-an, W.: SigPID: significant permission identification for android malware detection (2016)
19. Zhang, H., Yan, Z., Yang, J., Tapia, E.M., Crandall, D.J.: mFingerprint: privacy-preserving user modeling with multimodal mobile device footprints. In: Kennedy, W.G., Agarwal, N., Yang, S.J. (eds.) SBP 2014. LNCS, vol. 8393, pp. 195–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05579-4_24
20. Zhao, X., Feng, T., Shi, W.: Continuous mobile authentication using a novel graphic touch gesture feature. In: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 1–6. IEEE (2013)