Improving User Identification System on Walking Patterns

What is the problem?

The traditional user authentication methods (passwords, fingerprints, face locks, verification code s, etc.) have many drawbacks such as being time-consuming, only at entry-points, and more importantly, susceptible to attacks including side-channel, shoulder surfing, social engineering [3], etc. As a result, there's an increasing need for building a continuous user authentication system as additional support to improve information security. This project is based on research "An empirical evaluation of activities and classifiers for user identification on smartphones" by C. Tang and V. V. Phoha [2], who had tried 10 different classifiers to classify users based on accelerometer and gyroscope data captured by mobile devices. They claimed that k-NN produced a consistent and outstanding performance (almost 100% accuracy) under dynamic behaviors (walking, upstairs, downstairs), static acts (sitting, standing, lying) and transitions between them, as well as under unlabeled activities.

In our project, we point out the deficiencies of their work. Moreover, we focus on the lack of parameter tuning of their research and try to improve their system by tuning the parameters on the same dataset [5] (Walking Pattern Data Set from UCI Machine Learning Repository). Our design is similar to theirs.

Significance of our work

Tang et al.'s work [2] is a good approach for user identification on smartphones. However, there are many deficiencies. Our work pointed out the weaknesses of their work and tried to improve the system performance.

Literature search

Serwadda et al.'s work [1] takes the first steps towards bridging the gap among various experimental methodologies for touch-based authentication and presents a benchmark dataset for further researches. They also performed a user-level performance evaluation in which they set up EER thresholds for the "failure to enroll" policy. One question emerges after reading this paper: how to set the optimal EER threshold to balance the system performance and user enrollment rate, or can we define a metric to measure the cost-benefit trade-off between EER and the user enrollment rate.

Tang et al.'s work [2] is what we base on in our project. They tried 10 different classification models (Random Forest, SVM, Naïve Bayes, J48, Neural Network, k-NN, RPart, JRip, Bagging, and AdaBoost). However, there are some deficiencies in their work, and we try to fix some of these issues.

1. The dataset (the number of users and the number of samples, especially transition samples) isn't big enough to make sure that this system can be deployed in real-world scenarios where there could be thousands of users, some of which might have very similar patterns.
2. Although K-NN gives outstanding performance on these datasets, the time complexity of computing could be insanely high. We might need to try approximation algorithms for k-NN.

3. There could be many other scenarios that this research didn't consider: put on a table, carried on a vehicle, hold by hands, etc., under which (especially on a table) the k-NN classifier would probably work. We might need to try training a more generalized model and/or combining this system with touch-based systems, etc. This issue was addressed in Kumar et al.'s work [3].
4. This research uses default or straightforward parameters for models, which could influence a lot on their performances. Tuning is an art.

Kumar et al.'s work [3] proposed a continuous user authentication system which entirely relies on unlabeled phone movement patterns in a completely unconstrained environment. They use k-means to get clusters of preprocessed data (feature vectors) and use these clusters (feature vectors and cluster IDs) to train the Context Identification Model for the system and Authentication Models for each cluster. The preprocessed testing data first go through the Context Identification Model to get cluster IDs and then the Authentication Model to check if a user is genuine.

One common deficiency among these researches is that they didn't tune the parameters thoroughly: they either set parameters arbitrarily or use the default settings.

Our approach to solving the problem

Data Preprocessing

The Walking Pattern Data Set from the UCI Machine Learning Repository contains accelerometer time series data from 22 users. It's easy to tell that there's an apparent discrepancy between different users' walking patterns from our first glance at the x-axis accelerometer data.

We partition the raw data into fixed-width (100 samples) sliding windows with a 50% (50 samples) overlap and then extract 9 features (mean values, standard deviations, and median absolute deviations of x, y, and z axes) from every window. We remove users 3, 5, 16, and 19 as they have too few samples (whose csv file size is less than 5kb) from our data set and randomly partition the dataset into a training set (80% of data) and a testing set (20% of data).

Models

In our project, we try Decision Trees, Discriminant Analysis, Naïve Bayes, SVM, k-NN, and Ensemble with parameter tuning.

After preprocessing the dataset, the data can be loaded into MATLAB Classification Learner, a comprehensive GUI that integrates many state-of-the-art and mainstream machine learning models (Logistic Regression, Decision Trees, Discriminant Analysis, Naïve Bayes, SVM, k-NN, and Ensemble) and supports parameter tuning.

Results

Comparing the training set accuracies under default parameters

Model	Accuracy (train)	Model	Accuracy (train)
Tree (Fine)	73.5%	SVM (Coarse Gaussian)	46.7%
Tree (Medium)	52.0%	**KNN (Fine)**	**82.3%**
Tree (Coarse)	34.6%	KNN (Medium)	78.2%

Discriminant (Linear)	52.0%	KNN (Coarse)	58.7%
Discriminant (Quadratic)	68.3%	KNN (Cosine)	76.9%
Naïve Bayes (Gaussian)	49.1%	KNN (Cubic)	77.9%
Naïve Bayes (Kernel)	68.7%	**KNN (Weighted)**	**81.8%**
SVM (Linear)	66.9%	Ensemble (Boosted Trees)	67.6%
SVM (Quadratic)	**82.5%**	**Ensemble (Bagged Trees)**	**81.7%**
SVM (Cubic)	**83.5%**	Ensemble (Subspace Discriminant)	40.0%
SVM (Fine Gaussian)	76.3%	**Ensemble (Subspace KNN)**	**84.0%**
SVM (Medium Gaussian)	78.1%	Ensemble (RUSBoosted Trees)	59.5%

Parameter Tuning

After the first glance, we found that Quadratic SVM, Cubic SVM, Fine KNN, Weighted KNN, Bagged Trees, and Subspace KNN have a good accuracy over the training set. We select these models for further parameter tuning. After tuning, we get the accuracies below. We can see with simple tuning the model can be better.

Model	Quad SVM	Cubic SVM	Fine KNN	Weighted KNN	Bagged Trees	Subspace KNN
Acc (Train)	82.9%	83.6%	82.6%	82.7%	82.2%	84.9%

Test Models

We export these tuned models and test their performances on the test set.

Model	Quad SVM	Cubic SVM	Fine KNN	Weighted KNN	Bagged Trees	Subspace KNN
Acc (Test)	83.74%	84.62%	84.44%	84.62%	83.39%	87.94%

Under Subspace KNN, we calculate the sensitivity and specificity for each user.

User ID	Accuracy	Sensitivity	Specificity
1	0.994755	0.8666667	0.9982047
2	0.986014	0.5555556	0.9928952
4	0.986014	0.8235294	0.9962825
6	0.98951	0.862069	0.9963168
7	0.996503	0.9545455	0.9981818
8	0.986014	0.6363636	0.9928699
9	0.984266	0.8148148	0.9926606
10	0.991259	1	0.9910873
11	0.987762	0.92	0.9908592
12	0.994755	1	0.994575
13	0.987762	0.8461538	0.9945055
14	0.991259	0.9433962	0.9961464
15	0.986014	0.5	0.9964286
17	0.966783	0.8913043	0.98125
18	0.973776	0.8846154	0.9878543
20	0.987762	0.9666667	0.9902344
21	0.994755	1	0.9946903
22	0.973776	0.8333333	0.9849057
Avg	0.986597	0.8499452	0.9927749

Comparison

We compare our result with Tang et al.'s work. We can conclude that tuning the parameters (or even trying different default parameters set by different software) has significant potential in improving the system performance.

Classifier	Accuracy (our work)	Accuracy (Tang et al.'s work)
Decision Trees	*73.5% (Fine, without tuning)*	78.0% (J48)
Discriminant Analysis	*68.3% (Quadratic, without tuning)*	-
Naïve Bayes	*68.7% (Kernel, without tuning)*	48.3%
SVM	**84.62% (Cubic, with tuning)**	77.2%
KNN	84.62% (Weighted, with tuning)	**99.8%**
Bagging	**83.39% (with tuning)**	69.9%
Boosting	*67.6% (without tuning)*	71.9% (AdaBoost)
Subspace KNN	**87.94% (with tuning)**	-
Random Forests	-	85.7%
Neural Network	-	76.3%

Challenges we have faced and how we addressed those challenges

One challenge is tuning the parameter, which requires good understanding in this field (gait recognition), experience in machine learning models, and lots of effort. After reading Wikipedia, we learned that many approaches could be used for hyperparameter optimization, including grid search, random search, Bayesian optimization, gradient-based optimization, evolutionary optimization, population-based training, etc. However, we don't have enough time working on all these methods.

Other researches by Phoha et al. have considered phone usage contexts by training context identification model and combing continuous phone movement pattern model with touch-based models, etc. With the increasing demand to try much larger data sets to see if the model still works consistently, however, there's no large enough dataset available publicly. We might need to do a more extensive experiment by advertising the app collecting movement patterns publicly to collect sufficient data for the study.

Although we started parameter tuning and k-NN approximation, there's still a long way to go on this path, and we haven't got a good result for k-NN approximation yet.

What are the venues where our work can be shared?

Classroom.

References

[1] A. Serwadda, V. V. Phoha and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, 2013, pp. 1-8.

[2] C. Tang and V. V. Phoha, "An empirical evaluation of activities and classifiers for user identification on smartphones," 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), Niagara Falls, NY, 2016, pp. 1-8.

[3] Kumar, Rajesh, Partha Pratim Kundu, Diksha Shukla, and Vir V. Phoha. "Continuous user authentication via unlabeled phone movement patterns." In 2017 IEEE International Joint Conference on Biometrics (IJCB), pp. 177-184. IEEE, 2017.

[4] Google and Wikipedia

[5] Walking Activity Data Set from UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/User+Identification+From+Walking+Activity