Supervised Learning Methods for Biometric Authentication on Mobile Devices

Valerie Ding[1], Stephanie Dong[2], Jonathan Li[3]
Department of Computer Science, Stanford University
[1]dingv@stanford.edu, [2]sxdong11@stanford.edu, [3]johnnyli@stanford.edu

Abstract

- We develop **logistic regression** models and **deep neural networks** to classify mobile device users based on biometric typing pattern data.
- We present a secure, space-efficient, extensible framework for **real-time biometric fraud detection** on mobile devices.

Introduction

- Keystroke pattern and dynamics classification is an important application of machine learning to computer security and authentication. The massive increase in popularity and **computing power of mobile devices** in the last ten years has spurred significant interest in biometric authentication models for mobile devices.
- Existing literature emphasizes need for more nuanced **security protocols** in personal devices. As mobile devices store increasingly valuable and confidential information, learning classifiers to detect fraud is becoming ever more applicable and important.
- At the same time, a general, space-efficient, and real-time framework is required to be viable in practice. To this end, we develop fraud detection algorithms that use real-time keystroke dynamics data, and propose a **space-efficient real-time authentication framework** that can be integrated into native software across all mobile devices.

Data and Features

- The MEU-Mobile KSD (Keystroke Dynamics) Data Set from the UCI Machine Learning Repository contains 51 records for each of 56 subjects - 2856 records total - of haptic, momentum, and timing features measured of a common sequence (.tie5Roanl) typed on a Nexus 7 mobile device. There are 71 features monitored, characterized by the attributes Hold, Up-Down, Down-Down, Pressure, Finger-Area, Average Hold, Average Pressure, Average Area.
- We trained a variety of binary classifiers to detect if the concatenation the feature vectors of two data-points, forming a vector 142 features, was typed by the same user or not. This allowed us to train one model and have it generalize user verification to any new users not from the training dataset, as long as we one at least one keystroke record for any new users.
- As part of data processing we implemented a flexible **resampling framework** that can utilize a variety of undersampling and oversampling methods to undersample the majority class and oversample the minority class as necessary. This ensures parity between labels of different user and same user in the training data.

71 features

56 subjects
51 records/
subject

Subject	Hold .	...	AvA
1	89	…	0.2880184
…	…	…	…
56	80	…	0.260369

Table 1. Feature vector format.

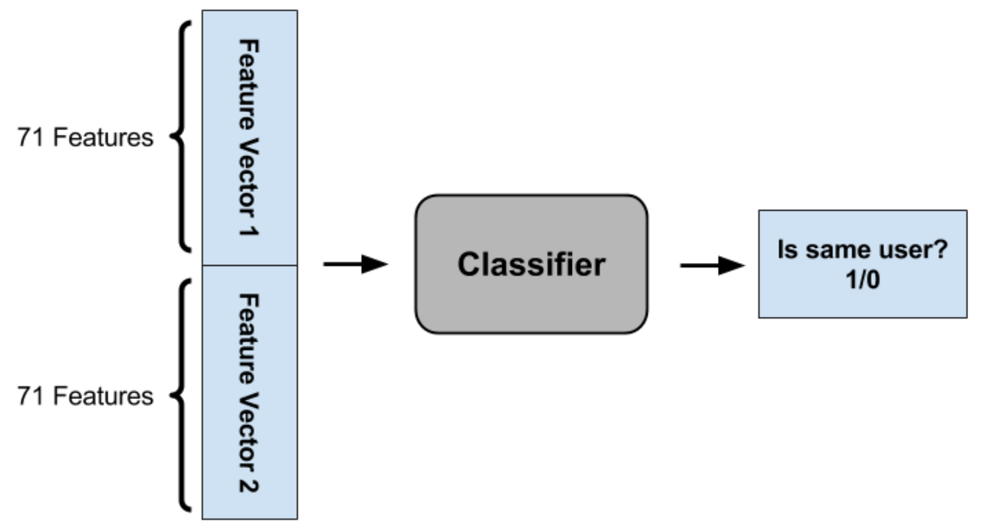

Figure 1. User Verification Classifier.

Models and Discussion

- **Logistic regression with cross entropy loss and no resampling**
For the first 10 examples each of the first 10 users, with 70%-30% train-validation split, we achieved **89.6% accuracy** with 0% precision, 0% recall, and 100% specificity. Upon inspection, the model consistently predicted the 0 label for every single validation example. We hypothesized that the disproportionate prediction of label 0 was due to heavily unbalanced data, with a significant majority class 0.

- **Logistic regression with cross entropy loss and 50-50 undersampling**
In our next attempt, we used resampling techniques to balance the majority and minority class to parity. With 50-50 undersampling using random undersampling of the majority class, and using our entire post-processed dataset of 8 million comparative examples, we achieved 50.0% accuracy, 50.0% precision, 100% recall, and 00.02% specificity. This means 50% likelihood of predicting same user when the user was in fact different. This is not more effective than a random guess, so the challenge will be lowering the false positive count, as it is more important, from a security point of view, to minimize false positives (predict same user, but actually different) than false negatives (predict different user, but actually same).

$$\mathrm{IsSameUser}(x) = [\sigma(Wxb) > 0.5]$$

- **Fully Connected Deep NN with cross entropy loss and 50-50 undersampling**
From the results in the section above, we hypothesized a single logistic unit could not represent enough complexity to capture the relationship between the 142 features of our input. Hence, we trained a variety of fully connect deep neural networks and compared their validation accuracy. Deep neural nets ranging from 1 hidden layer to 5 hidden layers, with 10 neurons per hidden layer, with ReLU activation, and sigmoid activation on the output layer. The loss function remained cross entropy. We trained each DNN model for 25 epochs from randomly initialized weights and measured their validation accuracy. This was repeated 10 times for each DNN model, and the average of validation accuracy of 10 trials was recorded as a benchmark of how each additional layer improve the performance of the model.

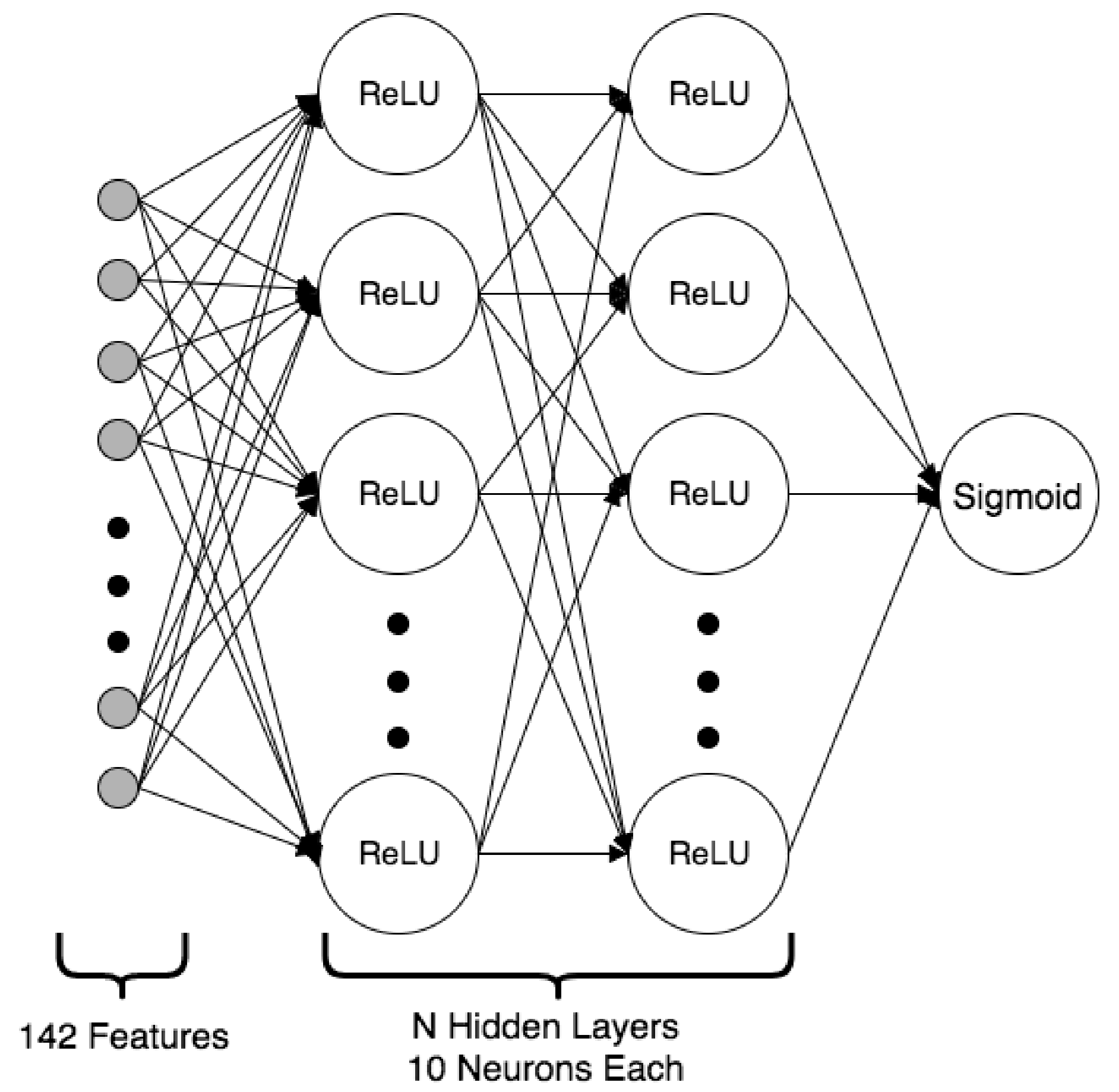

Figure 2. Deep neural network training architecture. We use ReLU activation for hidden layers (10 units each) and sigmoid for output. We experimented on varying number of hidden layers.

Results

- From this investigation, we conclude with 10 ReLU activated neurons per hidden layer, the 3 hidden layer model achieves the best accuracy vs. training speed trade-off. Hence, we retrained that model for 100 epochs, and produced the following validation results: **79.8% accuracy, 81.7% precision, 76.8% recall, and 95.1% specificity.**

Model	Loss	Accuracy	Recall	Precision	False Negative Rate
Logistic	6.661	58.12%	23.88%	78.18%	76.12%
DeepNN 1	0.690	69.10%	49.05%	**83.31%**	50.95%
DeepNN 2	7.905	50.41%	**99.78%**	49.76%	**0.22%**
DeepNN 3	0.536	73.73%	70.37%	76.04%	29.63%
DeepNN 4	0.695	49.73%	0.21%	59.89%	99.79%
DeepNN 5	0.531	73.45%	72.52%	74.30%	27.48%
DeepNN 6	0.409	81.70%	91.18%	77.40%	8.82%
DeepNN 7	0.484	76.88%	97.16%	69.15%	2.84%
DeepNN 8	0.527	72.93%	80.23%	70.79%	19.77%
DeepNN 9	0.425	80.29%	81.69%	80.19%	18.31%
DeepNN 10	0.450	79.78%	77.46%	81.37%	22.54%
Triangle	**0.380**	**84.28%**	89.76%	81.14%	10.24%

Table 2. Deep Neural Net models and their average validation accuracy of 10 training trials

Conclusions and Future Work

- We developed logistic regression and deep neural network classifiers for user verification through biometric typing pattern data on mobile devices, achieving **79.8% accuracy**. This means that 79.8% of the time, our classifier was able to successfully differentiate between two user inputs.
- Applications of these discriminatory classifiers are in enhancing device security by adding another **layer of verification**. By writing this classifier onto mobile devices and training it on a user's featurized password input, we can ensure that even if a password's content is typed in properly, it must be typed in with the learned cadence of the original user in order to be verified. This will effectively proof every mobile device from brute force password attacks by adding an unknown number of additional features the attacker must account for. Additionally, this has the ability to continually verify the authenticity of the user based on their typing patterns as they use the mobile device, hardening against device takeover by ensuring that only the primary user has access to the phone.
- **Future work:** In this research, we resampled by downsampling the majority class. Our flexible resampling framework allows for different resampling techniques, which can be explored in the future. Additionally, we can perform data augmentation using adversarial examples or other upsampling techniques. This would help inform next-generation development of discriminatory classifiers.

Acknowledgments

We would like to thank Prof. Dan Boneh for motivation behind the research and valuable insight on approaches and application, and Prof. Andrew Ng for guidance in machine learning techniques. We would also like to thank Steve Mussmann, Christopher Sauer, and Alisha Rege for advice and feedback on our research.

References

[1] N. Al-Obaidi. MEU-Mobile KSD Data Set. UCI Machine Learning Repository, 2016.
[2] I. de Mendizabal-Vazquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Sanchez-Avila. Supervised classification methods applied to
Keystroke Dynamics through Mobile Devices. *ICCST*, 2014.
[3] T. Cho. Pattern Classification Methods for Keystroke Analysis. *SICE-ICASE*, 2006.
[4] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research*, 2014.
[5] A. Fawzi, S. Moosavi-Dezfooli, P. Frossard. Robustness of classifiers: from adversarial to random noise. *NIPS*, 2016.
[6] C. Dwork, A. Roth. Differential privacy. *Foundations and Trends in Computer Science*, 2014.
[7] Y. Gal, Z. Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. *arXiv:1506.02158*, 2016.
[8] P.S. Teh, N. Zhang, A.B.J. Teoh, K. Chen. A survey on touch dynamics authentication in mobile devices. *Computers & Security*, 2016.
[9] H. Bae, S. Monti, M. Montano, M.H. Steinberg, T.T. Perls, P. Sebastiani. Learning Bayesian Networks from Correlated Data. *Nature Scientific Reports,* 2016.