Biometric Authentication of Smartphone Users with Support Vector Machines

Nico Espinosa Dice

April 6, 2020

Contents

1 Introduction.

1.1 The Security-Convenience Trade-off.

Security is essential for the safety of users and companies. In particular, security failures pose existential threats to reputation-oriented companies, especially companies whose services center around trust with consumers. Consumers offer financial companies their personal information in return for a service. In return, consumers expect that companies will keep their information private. If

their personal information is exposed, consumers will be left vulnerable and will likely never use the service again. Without consumers, a company's business is threatened.

Convenience is crucial for maintaining a high yield of returning users. When visiting a platform, lack of convenience drives users towards competitors who offer more convenient services. Thus, convenience is equally crucial for companies.

In general, more secure platforms tend to be less convenient, forcing companies to compromise between two necessities.

1.2 Background on Biometric Authentication.

Biometric authentication offers a solution for companies to have secure platforms without sacrificing the convenience for users.

Biometric authentication involves applying machine learning techniques towards authenticating users using biometric data. Biometric data is acquired from human characteristics, and it can include body measurements and human behavior. This data is often collected from sensors located in smartphones and smartwatches that are carried by individuals on their person.

Users have become increasingly reliant on electronic devices over the past two decades, and the data produced by electronic devices has risen as well. Consequently, the biometric data available to train biometric authentication models is readily available to cybersecurity companies. Because users already use electronic devices regularly, they are largely unaffected by the biometric data collection process. Similarly, since authentication models can run in the background of phones and computers, users are not required to wait for their authentication. Unburdened from the requirement of creating, updating, and memorizing password systems, users generally find biometric authentication significantly easier than password-centered authentication. And given the surplus of biometric data produced from users' devices, biometric authentication has the opportunity to be more secure than the alternative form of authentication.

2 Dataset: Biometric Data from Mobile Devices.

The dataset used for this paper's research is provided by IDSeal, a cybersecurity company; it is available publicly on Kaggle. The dataset consists of accelerometer data collected during "normal device usage" of smartphones over a period of several months. The sources of the acceleration data are the accelerometers located in the inertial measurement units of each smartphone.

2.1 Exploratory Data Analysis.

Data Features. Each sample in the dataset contains the following features:

- Timestamp: measured in milliseconds.

- Acceleration (X-direction) – measured in g's.
- Acceleration (Y-direction) – measured in g's.
- Acceleration (Z-direction) – measured in g's.
- DeviceId (Training Dataset) – unique identification of each device.
- SequenceId (Testing Dataset) – unique number assigned to each test sample.

Data Shape. The dataset consisted of 60,000,000 samples from 387 unique devices. The testing dataset contained 90,000 consecutive samples per device. Due to the size of the dataset, the model was trained and tested on a subset of the 60,000,000 samples.

Every device had at least 6000 samples, allowing for proper fitting to each device. Additionally, periods of zero-movement that lasted 10 or more seconds were removed from the dataset. Zero-movement periods were defined as samples where all three accelerations read within a pre-established epsilon distance of zero.

3 Mathematics of Support Vector Machine Model.

A support vector machine (SVM) was chosen as the algorithm to use in this model because of recommendations in the literature (Sitova et al. 2016). Additionally, SVMs provided pedagogical benefits, as they offer geometrical perspectives on machine learning generally and model optimization specifically, augmenting the probabilistic perspective we have developed in Math189R[1].

Support vector machines are described by the following equation[2]:

$$\hat{y}(x) = sgn\left(\hat{w_0} + \sum_{i=1}^{N} \alpha_i k(x_i, x)\right), \text{ where} \tag{1}$$

$$x_i : \text{support vector (when } \alpha_i > 0), \tag{2}$$

$$\alpha_i = \lambda_i y_i, \tag{3}$$

$$k : \text{kernel function.} \tag{4}$$

We will return to the kernel function in the subsequent subsection.

It is important to note that kernelized SVMs have $O(n_{\text{features}} \times n^2_{\text{observations}})$ complexity (Murphy 2012; Sitova et al. 2016). This will become relevant in the model implementation section.

[1] Math189R: Mathematics of Big Data. Taught at Harvey Mudd College by Weiqing Gu.
[2] Equation 14.59 (Murphy 2012).

3.1 Kernel Function

The radial basis function (RBF) was selected as the kernel for the SVM. The radial basis function is given by:

$$k(x, x') = \exp(-\frac{||x - x'||^2}{2\sigma^2}) \tag{5}$$

$$= \exp(-\gamma||x - x'||^2), \text{ where} \tag{6}$$

$$\gamma = \frac{1}{2\sigma^2}, \tag{7}$$

$$\sigma^2 : \text{bandwidth.} \tag{8}$$

RBF was chosen as the kernel function for three primary reasons. First, outliers in SVMs implemented with RBF generally have less impact than SVMs implemented with other kernel functions.

Secondly, SVMs using RBF are more effective in higher dimensions. This is particularly important when training a model on a dataset with many features. While the dataset used in this paper is low-dimensional, I hope to incorporate gyroscope and magnetometer measurements into the authentication model for the final project of Math189R. When applying the model to a dataset with more features, SVMs with RBF kernel functions will likely be more effective than other kernel functions.

Lastly, and most importantly, SVMs with RBF kernel functions excel when the intersection of classes is trivial. This holds true in this application and generally for biometric authentication: there is no overlap between classes; when users of devices are authenticated, they either are or are not the actual user they claim to be.

SVMs implemented with RBF kernels have disadvantages as well. They have a long fitting time, which poses difficulties when cross-validating. (I will return to address this challenge in the Model Implementation section). Additionally, RBFs are difficult to visualize and consequently can be difficult to interpret.

Nevertheless, it was believed that using RBF as the kernel would result in the highest accuracy in model prediction, so it was selected as the kernel function.

3.2 Feature Engineering: Hyperparameter Selection.

Hyperparameters for Support Vector Machine. γ is a hyperparameter present in the equation for the RBF kernel function (Equation 6). Its value is generally determined based on cross-validation, which is detailed below. However, as stated previously, fitting an SVM kernelized with RBF is complex and takes a long time, especially when coupled with the limited computational power of a laptop. Thus, for the midterm project, I was unable to cross-validate to empirically optimize the value of γ. Instead, I chose the value of γ based on recommendations in the literature:

$$\gamma \text{ (Gamma)} = \frac{1}{\text{Number of Features} \times \text{Variance of Z-Acceleration}}. \tag{9}$$

The variance of the Z-direction was chosen because the acceleration in the Z-direction had the largest variance of all the accelerations.

C is a regularization parameter. Like γ, its value is generally optimized through cross-validation. Instead, its value was selected based on recommendations in the literature.

The *kernel function* of an SVM can be considered a hyperparameter in the sense that its value is set before fitting the model. The rationale for selected RBF for the kernel is depicted in Section 4.1.

Tolerance for Stopping Criterion is a parameter used when fitting the model. Its value determines when the model ceases to fit based on the "improvements" it is gaining. The value was suggested in the literature.

Cross-Validation for γ and C. For the final project, I will cross-validate γ and C over a two-dimensional grid consisting of the values:

$$\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}, \tag{10}$$

$$C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}. \tag{11}$$

The cross-validation range was empirically determined in the literature (Hsu et al. 2009). The literature sources for other hyperparameter selection included: Hsu et al. 2009; Murphy 2012; Sitova et al. 2016.

4 Support Vector Machine Model Implementation.

The SVM was coded in Python using the scikit-learn library, with hyperparameters set as detailed in Section 4.2. The dataset was imported and explored using the Pandas library. The code is available here.

Because of the size of the dataset, the model was trained and tested on approximately 2% of the dataset. For the final project, I will train and test on a larger subset, if not the entire dataset.

Model Implementation Pseudocode.

```
Import data.
Partition dataset into training and testing data.
Scale the data:
    Create a scaler.
    Fit scaler to the training data.
    Transform the training and testing data.
Build the SVM with relevant characteristics.
Fit the SVM to the training data.
Predict the device corresponding to the test samples.
Compute and output accuracy.
```

5 Authentication Results.

Accuracy. *Accuracy* is defined as follows: for each test sample, the model predicted what device it corresponded to, based on probability estimates for each device. Accuracy is the percentage of correct device predictions. Another interpretation is that for each sample, if the model predicted the correct device, that test sample received a 1. If the model predicted an incorrect device, that test sample received a 0. Accuracy is the sum of all test scores divided by the total number of test samples, so accuracy is the average of the scores of all test samples.

The model produced approximately 80% accuracy.

6 Discussion.

This support vector machine model shows potential for biometric authentication applications. While 80% accuracy is certainly not above the production-level threshold, the model has multiple avenues for significant improvement, which will be discussed in Section 7.2.

6.1 Comparison to Literature.

The literature (Hsu et al. 2009; Sitova et al. 2016) describes models attuned to biometric authentication that significantly outperform this paper's SVM model. There are multiple reasons for this discrepancy.

First, some models in the literature are neural networks, which have been show to generally outperform SVMs. This is also true in regards to the Kaggle competition, where teams were able to achieve accuracy in the high 90% range. While the full description of their models is not publicly available, it is believed that leaders in the competition used neural networks and ensemble techniques to achieve such high accuracy.

Secondly, when comparing this paper's model to other SVMs in biometric authentication applications, an important factor is that this SVM's hyperparameters, specifically γ and C, have not yet been cross-validated. This will likely make a significant difference in the model's performance.

Third, some models in the literature were trained on the Opportunity dataset, where the goal is activity recognition, not authentication, so some of the comparisons are not perfectly symmetric.

Lastly, many of the models were tested on a range of consecutive samples, rather than just a single sample at a time, as was the case with this model. As is discussed in Section 7.2, I plan to apply these new techniques of incorporating the time series aspect of the data better into the model, which should substantially improve the model's performance.

6.2 Future Research.

For the final project, I will empirically optimize the γ and C hyperparameters through cross-validation on the range given in Section 4.2. I will also train and test on a dataset with gyroscope and magnetometer sensors, conducting Principal Component Analysis on features. Additionally, I may empirically compare results for SVMs using other kernels, specifically polynomial and sigmoid kernels.

Lastly, I will derive alternative forms of incorporating time series data with support vector machines. In particular, I will create exponential moving average variables that allow the model to take in a measure of the previous data points. Additionally, I may test the data on an array of several consecutive samples, as I believe this will closely resemble actual biometric authentication applications; often, models will have a few minutes of data before needing to authenticate users. I will reflect this in the model by asking the model to make a prediction based on an array of multiple consecutive data points, allowing the model to take advantage of the time series aspect of the data.

7 References.

- Bhattacharyya, Debnath Ranjan, Rahul Alisherov, Farkhod Minkyu, Choi. (2009). Biometric Authentication: A Review. International Journal of u- and e- Service, Science and Technology. 2.

- Murphy, Kevin P. Machine Learning: A Probabilistic Perspective. 4th ed., Cambridge, MIT Press, 2013.

- R., Malathi Jeberson Retna Raj, Retna. (2016). An Integrated Approach of Physical Biometric Authentication System. Procedia Computer Science. 85. 820-826. 10.1016/j.procs.2016.05.271. Sitov a, Zdenka Sedenka, Jaroslav Yang, Qing Peng, Ge Zhou, Gang Gasti, Paolo Balagani, Kiran. (2015). HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. IEEE Transactions on Information Forensics and Security. 11. 1-1. 10.1109/TIFS.2015.2506542.

- Wang, Jindong Chen, Yiqiang Hao, Shuji Peng, Xiaohui Lisha, Hu. (2017). Deep Learning for Sensor-based Activity Recognition: A Survey. Pattern Recognition Letters. 10.1016/j.patrec.2018.02.010