Biometric Authentication of Smartphone Users with Support Vector Machines

Nico Espinosa Dice

May 11, 2020

Contents

1 Introduction. 2
1.1 The Security-Convenience Trade-off. . . . 2
1.2 Background on Biometric Authentication. . . . 2
1.3 Experimental Question. . . . 3
1.3.1 Hypothesis. . . . 3

2 Dataset: Biometric Data from Mobile Devices. 3
2.1 Exploratory Data Analysis. . . . 3
2.1.1 Data Features. . . . 3
2.1.2 Data Shape. . . . 4

3 Mathematics of Support Vector Machine Model. 4
3.1 Kernel Functions. . . . 4
3.1.1 Linear Function. . . . 4
3.1.2 Polynomial Function. . . . 5
3.1.3 Sigmoid Function. . . . 5
3.1.4 Radial Basis Function. . . . 5
3.1.5 Kernel Function Comparison: A Visual Perspective. . . . 6
3.2 Hyperparameters. . . . 7
3.2.1 γ (Gamma). . . . 7
3.2.2 C (Regularization Parameter). . . . 7
3.2.3 Kernel Function. . . . 7
3.2.4 Tolerance for Stopping Criterion. . . . 7
3.3 Cross-Validation and Hyperparameter Tuning. . . . 7
3.3.1 Kernel Function. . . . 7
3.3.2 Cross-Validation for γ and C. . . . 8

4 Support Vector Machine Model Implementation. 8
4.1 Code. . . . 8
4.2 Model Implementation Pseudocode. . . . 9

5 Authentication Results. **9**
5.1 Accuracy Definition. 9
5.2 Model Performance. 9
5.3 Tradeoff: Accuracy vs. Number of Devices. 10

6 Discussion. **10**
6.1 Comparison to Literature. 10
6.2 Future Research. 11

7 Acknowledgements. **11**

8 References. **12**

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.

1.3 Experimental Question.

In this paper, I will examine the use and performance of support vector machine (SVM) models in biometric authentication applications. Furthermore, this paper will empirically examine the performance of varying kernel functions, including linear, polynomial, sigmoid, and radial basis functions.

1.3.1 Hypothesis.

For reasons articulated in Section 3, I predict that the support vector machine implemented with a radial basis kernel function will outperform the SVM implemented with linear, polynomial, and sigmoid kernel functions.

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.

2.1.1 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.

2.1.2 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.

3.1 Kernel Functions.

3.1.1 Linear Function.

The linear kernel function is given by the following equation:

$$k(x, x') = x^\top x'. \tag{5}$$

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

3.1.2 Polynomial Function.

The polynomial kernel function is given by the following equation:

$$k(x, x') = (\frac{x^\top x'}{2\sigma^2} + c_0)^d \tag{6}$$

$$= (\gamma x^\top x' + c_0)^d, \text{ where} \tag{7}$$

$$\gamma = \frac{1}{2\sigma^2}, \tag{8}$$

$$\sigma^2 : \text{“bandwidth,”} \tag{9}$$

$$d : \text{kernel degree.} \tag{10}$$

3.1.3 Sigmoid Function.

The sigmoid kernel function is given by the following equation:

$$k(x, x') = \tanh(\gamma x^\top x' + c_0), \text{ where} \tag{11}$$

$$\gamma : \text{“slope,”} \tag{12}$$

$$c_0 : \text{“intercept.”} \tag{13}$$

3.1.4 Radial Basis Function.

The radial basis function is given by:

$$k(x, x') = \exp(-\frac{||x - x'||^2}{2\sigma^2}) \tag{14}$$

$$= \exp(-\gamma||x - x'||^2), \text{ where} \tag{15}$$

$$\gamma = \frac{1}{2\sigma^2}, \tag{16}$$

$$\sigma^2 : \text{“bandwidth.”} \tag{17}$$

The radial basis function (RBF) was initially selected as the kernel for the SVM.

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. 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, which is discussed in Section 4. 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 primary kernel function.

3.1.5 Kernel Function Comparison: A Visual Perspective.

After having discussed the mathematics behind the kernel functions, the following images can offer an intuitive, visual explanation for why radial basis functions are predicted to perform better in biometric authentication applications.

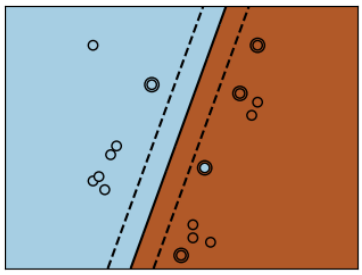

Figure 1: Linear Kernel

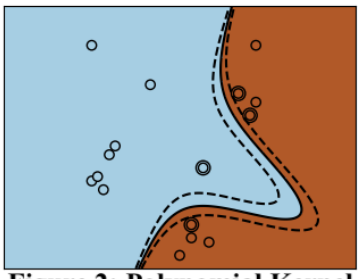

Figure 2: Polynomial Kernel

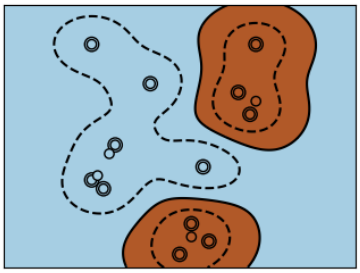

Figure 3: RBF Kernel

Source: scikit-learn.org

We see that the radial basis kernel function has the ability to partition the data points non-linearly. In biometric authentication, we assume that users will have sufficiently unique movements such that they can be distinguished. Because the model seeks to distinguish users on the individual level, the model must be able to find all types of patterns. In particular, we have limited knowledge in how the users will distinguish themselves, and we have no evidence to suggest that they will do so linearly.

3.2 Hyperparameters.

3.2.1 γ (Gamma).

γ is a hyperparameter present in the polynomial, sigmoid, and radial basis kernel functions – Equations 7, 11, and 15, respectively. Initially, a value of γ was chosen based on recommendations in the literature:

$$\gamma \text{ (Gamma)} = \frac{1}{\text{Number of Features} \times \text{Variance of Z-Acceleration}}. \tag{18}$$

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

3.2.2 C (Regularization Parameter).

C is a regularization parameter. Like γ, its value is generally optimized through cross-validation. The literature recommends a value of 1; specifically, the default value in the scikit-learn SVM library is:

$$C = 1. \tag{19}$$

3.2.3 Kernel Function.

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 3.1.

3.2.4 Tolerance for Stopping Criterion.

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 selected value was suggested in the literature:

$$\text{Tolerance for Stopping Criterion} = 1 \times 10^{-3}. \tag{20}$$

The literature sources included: Hsu et al. 2009; Murphy 2012; Sitova et al. 2016.

3.3 Cross-Validation and Hyperparameter Tuning.

3.3.1 Kernel Function.

γ and C were cross-validated over a two-dimensional grid consisting of the values:

$$\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}, \tag{21}$$

$$C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}. \tag{22}$$

The cross-validation range was empirically determined in the literature (Hsu et al. 2009).

The result of the cross-validation was:

$$\text{Kernel} : \text{Radial Basis Function}, \tag{23}$$

$$C = 1. \tag{24}$$

Furthermore, the following table offers a perspective on the performance of each kernel:

Kernel Function	C	γ	**Degree**	**Accuracy**
Sigmoid	1	1/(Number of Features)	N/A	53%
Linear	1	N/A	N/A	70%
Polynomial	1	1/(Number of Features)	3	77%
RBF	1	1/(Number of Features)	N/A	79%

Table 1: Accuracies of SVMs with Varying Kernel Functions.

3.3.2 Cross-Validation for γ and C.

Once the radial basis function was determined as the ideal kernel, γ and C were cross-validated on an SVM with an RBF kernel:

$$C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}, \tag{25}$$

$$\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}. \tag{26}$$

The cross-validation range was empirically determined in the literature (Hsu et al. 2009; Murphy 2012).

The result of the cross-validation was:

$$\gamma = 2, \tag{27}$$

$$C = 2^9 = 512. \tag{28}$$

Because of the complexity of training a support vector machine with a radial basis function, the cross-validation was implemented using a randomized search through the 2-dimensional grid of values. Thus, not every combination of γ and C values was tested; instead, a random subset of them were tested.

4 Support Vector Machine Model Implementation.

4.1 Code.

The SVM was coded in Python using the scikit-learn library, with hyperparameters set as detailed in Section 3.2. The dataset was imported and explored using the Pandas library. The code is available on my Github.

Because of the size of the dataset and limited computational power, the model was trained and tested on approximately 2% of the dataset.

4.2 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.

5.1 Accuracy Definition.

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.

5.2 Model Performance.

The models produced the following results:

Kernel Function	C	γ	**Degree**	**Accuracy**
Sigmoid	1	1/(Number of Features)	N/A	53%
Linear	1	N/A	N/A	70%
Polynomial	1	1/(Number of Features)	3	77%
RBF	1	1/(Number of Features)	N/A	79%
RBF (CV)[3]	2^9	2	N/A	83%

Table 2: Accuracies of SVMs with Varying Hyperparameters.

[3] Cross-validated.

5.3 Tradeoff: Accuracy vs. Number of Devices.

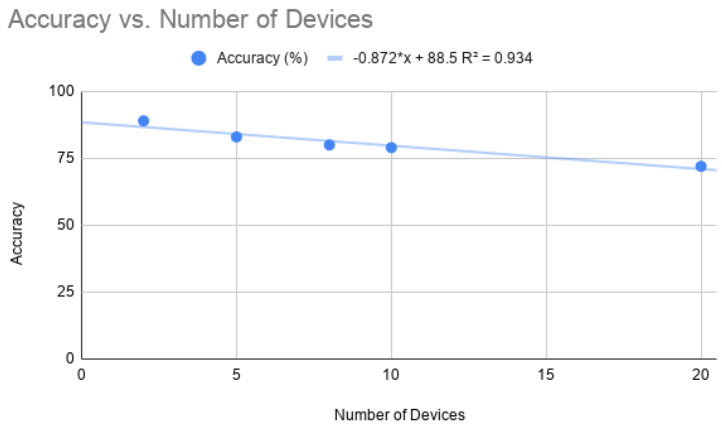

Figure 4: Accuracies of SVMs vs Number of Devices Trained and Tested On.

Figure 4 demonstrates the tradeoff between accuracy and how many devices the model is trained and tested on. It is clear that there is a negative slope, suggesting that as the number of users the model must distinguish increases, the accuracy decreases.

Moving forward, significant work will be required to improve this tradeoff, especially as biometric authentication companies seek security on global scales.

6 Discussion.

This paper demonstrates that support vector machine models show potential for biometric authentication applications. While 83% accuracy is certainly not above the production-level threshold, the model has avenues for significant improvement, which will be discussed in Section 6.2.

Furthermore, the results support the hypothesis that the radial basis function outperforms other kernel functions, including linear, polynomial, and sigmoid functions.

Additionally, the results demonstrate that the empirically determined γ and C values differ significantly from the suggested and default values in the literature, highlighting the importance of cross-validation in biometric authentication applications.

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, were cross-validated on using a randomized search process. Consequently, it is possible that the optimal values of γ and C were not evaluated, suggesting the possibility of further room for improvement.

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 6.2, 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.

In future research, I plan to train and test on a dataset with gyroscope and magnetometer sensors, conducting Principal Component Analysis on features. At Professor Gu's suggestion, this may require techniques beyond the scope of Math189R: Mathematics of Big Data. Professor Gu's course, Math178: Nonlinear Data Analytics, may offer the necessary skills to incorporate gyroscope and magnetometer sensor data into this model.

Additionally, I may look to deriving alternative forms of incorporating time series data. In particular, recurrent neural networks have been shown to excel with time series data, and they are a model that I plan to research further. This would allow for testing the model 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. In model development, I will reflect this situation 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 Acknowledgements.

Thanks to Professor Gu for teaching Math189R: Mathematics of Big Data and offering guidance throughout this project.

8 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.

- Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

- 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