User Authentication Using Keystroke Dynamics

Madhuri Ghorpade

Computer Science Department, California State University, Sacramento
madhurighorpade@csus.edu

Abstract: Traditional authentication methods such as PINs, tokens, passwords fail to keep up with the challenges presented as they can be lost or stolen. Whereas biometrics based on how the person behaves present a significant security advancement to meet these new challenges. Keystroke dynamics is a biometric technique to recognize and an analysis of user’s typing patterns.

The aim of this project is to implement and optimize supervised machine learning classification models to identify if a user who is trying to login into the system is a genuine user or an impostor depending upon the password typing rhythm of the user. Fully Connected Neural Network achieved highest accuracy of 97.59 % with lowest EER rate of 0.034.

Keywords: Keystroke Dynamics, Down-Down time, Up-Up time, latency, impostor, EER.

I. INTRODUCTION

With the ever-increasing technology use, there lies a threat to misuse the personal as well as official data. Hacking and cracking causes gains in unauthorized access which results in exploitation of sensitive and personal data. Traditional authentication mechanisms like passwords, pins and token possess a serious threat of lost or leaked. This calls for more secure access control in many of today’s security applications. Biometrics is the new emerging field of study where the aim is to make authentication mechanism stronger and less vulnerable. One such biometric measures is call Keystroke Dynamics.

Keystroke Dynamics is a behavioral biometric approach to improve the quality of the computer access rights. It verifies the individual person by its keystroke typing pattern. It is based on the assumption that each person has unique typing pattern. It uses two variables as measures: first is the dwell time which is period of time a user holds down a particular key and second one is the flight time, which is period of time user release those keys.

With keystroke dynamics, impostor attempts to authenticate using a stolen password could be detected and rejected because their typing rhythms differ significantly from those of the genuine user. This report is about designing various machine learning models to identify if a user, is a genuine user or an impostor based upon his keystroke dynamics using binary classification models. Also, this project will help to map a particular user with his potential user account by comparing his typing behavior with those of registered users using multi-class classification models. Support Vector Machines, K Nearest Neighbor [2], Fully Connected Neural Network and Convolution Neural Network models are implemented in this project. Deep learning models are implemented in this project as they possess the capability of pattern recognition in the data.

Application area for this project includes authentication of user at the time of login (static authentication), online examination (dynamic authentication), password recovery, account activity monitoring and intrusion detection.

II. RELATED WORK

To authenticate user against an account or resource based on his keystroke dynamics, various distance metrics are investigated in past which are effective dealing with the challenges intrinsic to keystroke dynamics data. Some of the distance metrics used are Manhattan distance, Euclidean distance etc. Few machine learning models like K means clustering, Nearest Neighbor [1], Artificial Neural Network models are experimented to authenticate a user. These models and distance metrics used equal error rate to measure the effectiveness of the models [5].

Keystroke dynamics experiment using the computer keyboard were conducted first by Umphress and Williams in 1985 [3]. In their study, they make a reference profile of a legitimate user using mean keystroke latency and mean time interval of consecutive characters so that it distinguishes the legitimate user from others. The result the study
showed was 11.7% FAR (false acceptance rate) and 5.8% FRR (false rejection rate).

Current studies show that the error rate is higher for Neural Network anomaly detectors compared to the other anomalies detectors like Manhattan, Nearest Neighbor. In this project I am optimizing the Equal Error Rate (EER) for all the models.

III. DATASET & FEATURES

Dataset is retrieved from Carnegie Mellon University [4] where they have asked 51 users to type ‘.tie5Roanl’ password 400 times each. For each user, overall typing speed, variation in the speeds, number of times back button is pressed, latency between any two keys pressed and released etc. is recorded. 20 timestamps are recorded, for each user, 10 for key-up and 10 for key-down. This includes key durations, down-down key latencies and up-up key latencies. Size of the dataset is 5 MB with 20400 records and 34 features. Each data is labeled by the ‘subjectID’ (1-51). Therefore, it is a supervised learning algorithm.

IV. APPROACH

The problem of authentication of users using their keystrokes is implemented as a classification problem. User’s keystrokes are compared against the genuine user’s recorded pattern. Depending upon the comparison it is identified if the user is genuine or an impostor.

Fig [1] is the workflow for this project. As the dataset contains 34 numerical attributes for each record, the variation of the data and correlation between them is studied first.

Feature extraction using PCA is performed to extract the important features from the 33 available features. Initially Random Forest was used for feature importance. Further it was noticed that PCA gave better feature extraction and model performance. Models are trained on the extracted important features and output is classified.

A. Exploratory Data Analysis

Fig [2] below is the data distribution of the numerical columns in the dataset. Majority of the columns appears to be right skewed.

Fig. 2 Original data distribution of all the numerical features of the dataset.

Standard scalar is applied to the features. As the data is sequential and statistical modeling methods require the data series to be stationary in order to be effective, given data is checked for stationary and non-stationary series. The data came out to be stationary. Fig [3] is the series plot which shows that the data do not have any trends and seasonal effects. Also, Augmented Dickey-Fuller test (null hypothesis test) was performed to confirm the same.

Fig. 3 Series plot showing no trends and seasonal effects

Next the inter-correlation between the attributes is studied with the help of a correlation heatmap. Fig [4] is the correlation map for the features of the dataset. A very few features are less correlated to each other while majority of them are highly correlated.
Fig [5] depicts the features importance obtained from Random Forest Classifier. Features are sorted in decreasing order of their importance to predict the target class.

Fig. 4 Correlation heatmap for the features of the dataset.

The pyplot shows that ‘DD.n.i’ which is the Down-Down time for key ‘n’ and ‘i’ is the most important feature for our target classification and features ‘DD.e.five’ (latency between pressing ‘e’ and ‘5’), ‘H.five’(latency between key ‘H’ and ‘5’), ‘rep’ and ‘sessionIndex’ doesn’t contribute towards our target so they are dropped from the final dataset. Random Forest feature selection is based on the Gini criteria which checks the amount of impurity each feature holds and decides the importance accordingly.

PCA is more versatile technique that works well in practice. It's fast and simple to implement and extracts only that features which has highest variance in the dataset. Fig [6] above is the plot of the cumulative explained variance vs number of components in PCA. To consider 100% variance, I have considered 21 components in the PCA based upon the chart.

Fig [7] is the count plot which confirms that the output class has equal number of elements for each class. Hence there is no class imbalance issue.

Fig. 7 Target class count plot showing equal number of elements for each type of the class.

Each feature is analyzed using the box-plot to determine if they have any outliers. After plotting the box-plot it came to observation that the few points of data in each feature appears to be outlier. This implies the fact that the rhythm of those 51 subjects is not consistent and there are lots of variation in their keystrokes. I plan to keep them as the variations seem to be important while identifying the genuine user.

Fig [8] is the one such box-plot for one of the features in the dataset.

Fig. 8 Box-plot of ‘H.Period’ feature

**B. TECHNOLOGY AND RESOURCES**

Both EDA and model implementation is done using Python in Jupyter Notebook.
The packages used are NumPy, Pandas, Sklearn, Keras, Scipy, Random Forest Classifier, PCA, Seaborn, Pyplot, Matplot etc.

V. IMPLEMENTATION

This project is implemented in two ways:
A. As a Binary Class classification
B. As a Multi-class classification

A. Binary Classification:

In this classification user is classified as either a genuine user or impostor with the help of binary values 0 and 1. A new dataset is created from the given dataset. Table 1 is the algorithm for it.

| TABLE I |
| BINARY CLASSIFICATION ALGORITHM |

| Input: Data set D with ‘m’ features |
| n \(\rightarrow\) the number of positive and negative records to choose per user |
| Output: New data set with ‘m+1’ feature |

1. Initialization: Create a new empty dataset D’ with m+1 feature. Append a new column ‘is_genuine’ to the D’
2. For each user U_i:
   - Select ‘n’ records from the D where S_i=U_i
   - Append these records to D’ and mark as ‘0’ (yes) in ‘is_genuine’ column.
   - Select ‘n’ records from D where S_i<> U_i
   - Append these records to D’ and mark as ‘1’ (no) in ‘is_genuine’ column.
3. Shuffle records from D’
4. return D’

After getting the new dataset D’ it is split into train and test with 70:30 ratio, for model training and testing. Test set is used to determine if the user who is trying to login into the system is a genuine user or an impostor.

B. Multi-class Classification:

As the dataset consists of 51 different users known as subjects; 51 target classes are made. Classification is done on the test set where each test record is classified to one of 51 target class. Dataset is divided in 70%-30% ratio for training and testing respectively.

If the desired class is equal to one which user is trying to log into then it can be said that the user is a genuine user, else he can be a potential impostor. Following machine learning algorithms are implemented for both types of classification:

- SVM,
- KNN,
- Fully Connected Neural Network Model,
- Convolution Neural Network Model

- Support Vector Machine (SVM)

Being a robust and accurate classification algorithm, it is used in this project as one of the base line model. SVM is also known for classifying highly non-linear data, it fits best for this dataset.

GridSearchCV is used to find the best fitting kernel ‘rbf’ both for binary classification and multi-class classification, and to avoid model overfitting with the help of cross validation. SVM helped in classifying the genuine users and impostors with the accuracy of 97.44%. For multi-class classification it gave an accuracy of 89.15%.

- K Nearest Neighbor (KNN)

KNN [2] uses complete training set as a model. Given a test data, the algorithm will find k points in the training set that are closest to the data point. Majority voting is used to predict the final class label of the dataset.

GridSearchCV is used to find the best value of K for binary classification and multi-class classification. Also, it makes sure that model is not overfitting the dataset with the help of cross validation. Best values for K were found as 3 and 7 for binary and multi-class respectively. KNN helped in classifying the genuine users and impostors with the accuracy of 97.46%. For multi-class classification it gave an accuracy of 85.27%.

- Fully Connected Neural Network (FCNN)

FCNN has the capacity of dealing with nonlinear problems and have a capability of pattern recognition. Since user authentication has nonlinear dependency towards keystrokes of user. Machine need to understand each genuine user’s typing pattern; therefore, this model is used.

Hyper parameter tuning was performed to find the best fitting model for given dataset. Best model parameters found were three dense layers with 240, 120 and 60 neurons, Sigmoid activation function and binary cross entropy loss function. Looping and Early stopping is used to avoid local minima problem. FCNN helped in classifying the genuine users and impostors with the accuracy of 97.59%. For multi-class classification it gave an accuracy of 91.12%.

- Convolution Neural Network (CNN)

CNN [4] has been studied and implemented for authenticating users based upon the keystroke
dynamics. Also, it possesses capabilities to extract underlying important feature without much human interference.

Hyper parameter tuning was performed to find the best fitting model for given dataset. Best model parameters found were two convolution layers with 32 and 128 neurons respectively and two max pooling layers, with one dense layer of 128 neurons and a dropout layer with relu activation function was found the best fitting model to the given dataset. SGD optimizer is used as ‘adam’ optimizer couldn’t perform well on given dataset for CNN. It helped in binary classification with accuracy of 96.93% and for multi-class it was 86.03%.

Training and validation error loss chart is plotted to confirm if the model is a good fit for training data. Fig [9] shows that FCNN model is a good fit for training data and there is no overfitting or underfitting. Similarly, all other neural network models were tested for overfitting and underfitting.

![Training and validation loss for FCNN model.](image)

### VI. EXPERIMENTAL RESULTS

This project was first implemented with Random Forest feature selection method, further it was observed that PCA helped in achieving more better performance. Since PCA gave the best results for all the algorithm implementations, it is considered as a final feature extraction process for this project.

Below is the summary of the results of both types of classification models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.9744</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.037</td>
</tr>
<tr>
<td>KNN</td>
<td>0.9746</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.039</td>
</tr>
<tr>
<td>Fully Connected</td>
<td><strong>0.9759</strong></td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td><strong>0.034</strong></td>
</tr>
<tr>
<td>Conv. Neural Network</td>
<td>0.9693</td>
<td>0.9693</td>
<td>0.9694</td>
<td>0.9693</td>
<td>0.037</td>
</tr>
</tbody>
</table>

![Table III Binary classification result](image)

The above summary of results states that FCNN has outperformed all other binary classification models with highest accuracy. While in multi-class classification, SVM performed best by giving highest accuracy.

Fig [10] shows the F1 comparison chart between all the models from multiclass classification as well as binary class classification. Fully connected dense neural network model gave the best F1 score and amongst all.

![Comparison of all the binary and multiclass classification models](image)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.8915</td>
<td>0.89</td>
<td>0.90</td>
<td>0.89</td>
<td>0.037</td>
</tr>
<tr>
<td>KNN</td>
<td>0.8527</td>
<td>0.85</td>
<td>0.85</td>
<td>0.86</td>
<td>0.039</td>
</tr>
<tr>
<td>Fully Connected</td>
<td><strong>0.9112</strong></td>
<td>0.81</td>
<td>0.82</td>
<td>0.81</td>
<td>0.034</td>
</tr>
<tr>
<td>Conv. Neural Network</td>
<td>0.8603</td>
<td>0.86</td>
<td>0.8635</td>
<td>0.8633</td>
<td>0.037</td>
</tr>
</tbody>
</table>

![Table II Multi-Class classification result](image)

The above summary of results states that FCNN has outperformed all other binary classification models with highest accuracy. While in multi-class classification, SVM performed best by giving highest accuracy.

Fig [10] shows the F1 comparison chart between all the models from multiclass classification as well as binary class classification. Fully connected dense neural network model gave the best F1 score and amongst all.

Fig[11] is the accuracy comparison between all the binary classification models. FCNN has the slightly better accuracy than the other used models.

![Comparison of Accuracy for binary classification models](image)

![Fig 11. Comparison of Accuracy for binary classification models](image)
It was challenging to decide the better performing model due to very less difference between accuracy and F1 score of the binary classification models. EER comparison came to the rescue. Fig [12] is EER comparison chart of the binary models, fully connected neural network (FCNN) has lowest EER. 60-70% less EER is achieved for all the models implemented in this project as compared to reference paper [1]. Fig [13] and Fig [14] are the confusion matrix and ROC curve for the same Fully connected neural network as it gave the best results.

Fig [15] is the multiclass classification confusion matrix of the SVM model as it has given best performance amongst multi-class classification models. The classes seem to be perfectly classified except a very few of them.

VII. CONCLUSIONS

It is evident from the results that Fully Connected Neural network with 3 dense layers of 240,120 and 60 neurons respectively, and with Sigmoid activation function gave the highest accuracy and lowest EER score amongst all the models. Therefore, FCNN is the best model to authenticate the user based upon the keystroke dynamics.

Grid Search with cross validation helped to find better fitting models and increased the model performance. Also, it made sure that models are not overfitting the training data. PCA helped to extract the important features more accurately which helped in having good model performance.

60-70% lower EER is achieved with this project as compared to [1] for all the models implemented in this project.

VIII. FUTURE SCOPE

The future scope for this project will be to implement and optimize the LSTM models as the data is time series dataset. LSTM models work better on such datasets.

REFERENCES