ITI105 Milestone Report

**User Authentication Using Classical Machine Learning:**

**Leveraging Key Typing Dynamics Behavior**

*Allen Lee and Jacob Abraham*

*8-Aug-2023*

# Problem statement and solution

Computing devices, including mobile phones, use various biometric authentication methods like fingerprints or facial features to identify users. However, these methods rely on specific hardware, which can increase the overall cost. An economical alternative is to authenticate users based on their behaviour, such as typing dynamics.

Keystroke dynamics, also known as keystroke biometrics, pertain to the comprehensive timing data that precisely records the moment each key is pressed and released as an individual types on a computer keyboard. These dynamics offer valuable insights that aid in user authentication. Through capturing the intervals between key presses, key hold durations, and the periods between key releases and next key presses, significant user insights can be derived. When the user logs in again, companies can compare their present typing pattern with their past patterns, allowing for authentication to distinguish legitimate users from potential fraudulent ones.

Conventional machine learning will be applied to generate models that can infer keystroke dynamics and then authenticate users.

# Description of data

Critical information that is captured during typing a password are:

* 1. Hold duration (H): Time from when a key was pressed to when it was released.
  2. Up-to-Down (UD) duration: Time from when key1 was released to when key2 was pressed.
  3. Down-to-Down (DD) duration: Time from when *key1* was pressed to when *key2* was pressed.

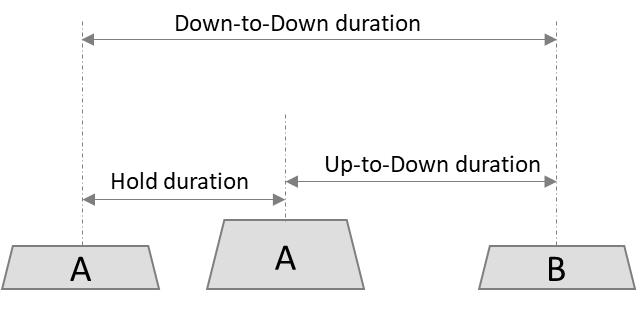


Figure 1. Critical keystroke timings

## Benchmark dataset

* Benchmark dataset was downloaded from Carnegie Melon University http://www.cs.cmu.edu/~keystroke/.
* Data consist of keystroke-timing information from 51 subjects (typists), each typing same 10-character password “.tie5Roanl” 400 times (in 8 sessions, with 50 repetitions per session).
* Total number of samples is 20400.
* Features and their data types are:

A screenshot of a computer

Description automatically generated

*Note:*

*H.period: during the period “.” Key was held.*

*DD.period.t: Time from when period “.” was pressed to when “t” key was pressed.*

*UD.period.t: Time from when period “.” was released to when “t” key was pressed.*



## Own hybrid dataset

* Create own key-sniffer program to collect 400 samples each from two new subjects (Allen and Jacob).
* A hybrid dataset will be created by using first 20 subjects’ data from the benchmark data set augmented with the additional data from the two new subjects.

# Conventional Machine Learning with benchmark dataset

## Exploratory Data Analysis (EDA) and Visualization

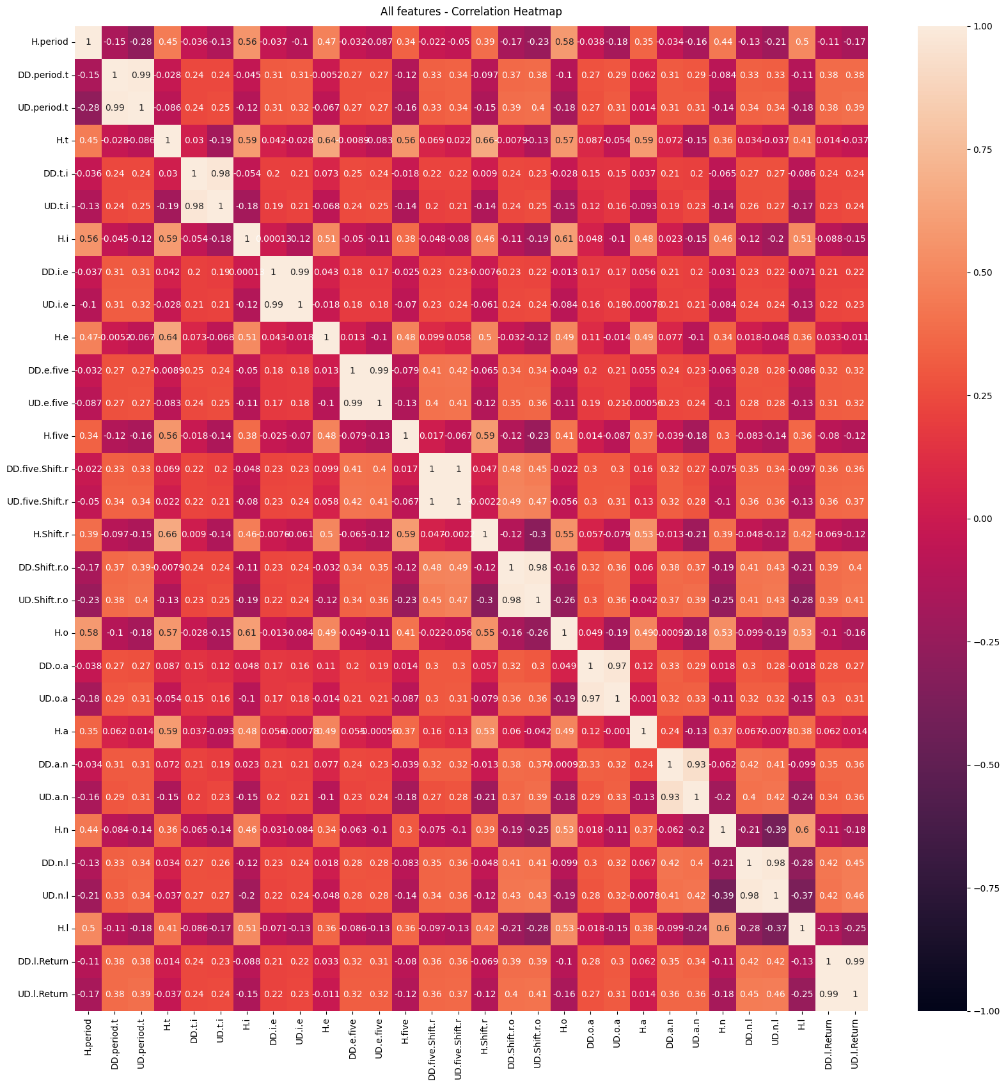
* Total number of samples is 20400.
* Total number of columns is 33.

A computer screen shot of a computer code

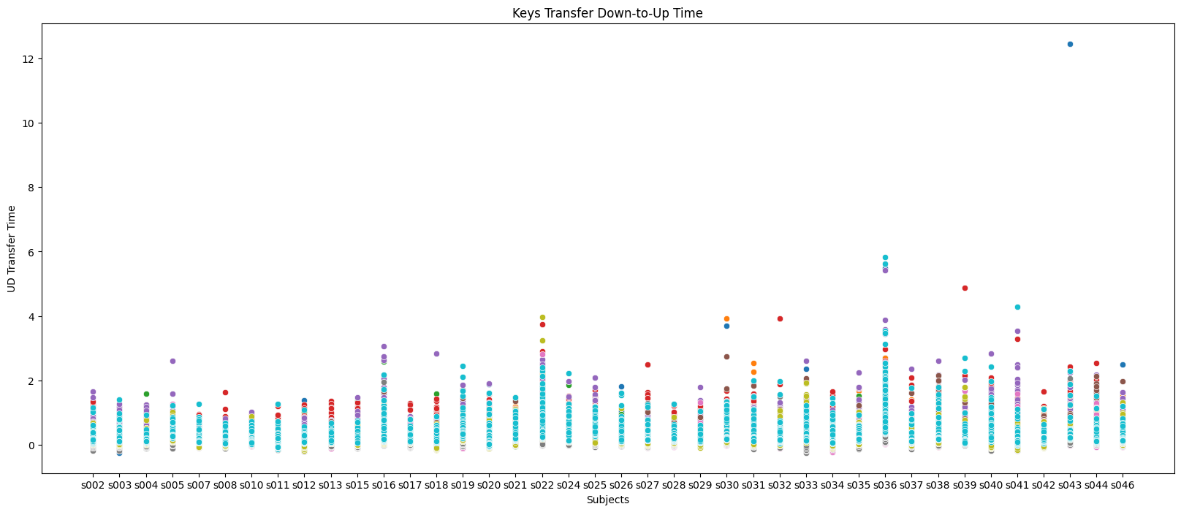
Description automatically generated

* Target is column “subject”, and its data type is “object”. Data type in other columns are “float”.
* Columns “sessionIndex” and “rep” are not used as it will not affect model training and testing.
* There are no missing values and “NaN” (refer to section 2.1).
* Columns of “Hold” duration shows low standard deviation compared to their corresponding columns of “Up-to-Down”.
* Values in all columns of “Down-to-Down” are the sum of their corresponding values in columns of “Hold” and “Up-to-Down”. Columns of “Up-to-Down” shows strong correlation with column “Down-to-Down”. Calculated correlation coefficients for these pairs of ‘Down-to-Down' and ‘Up-to-Down' for each character keypress are above 0.9 as shown below.
* Hence all columns prefixed with “DD”, are to be dropped to reduce the effects on model training due to correlated features. This reduces the number of features from 31 to 21.

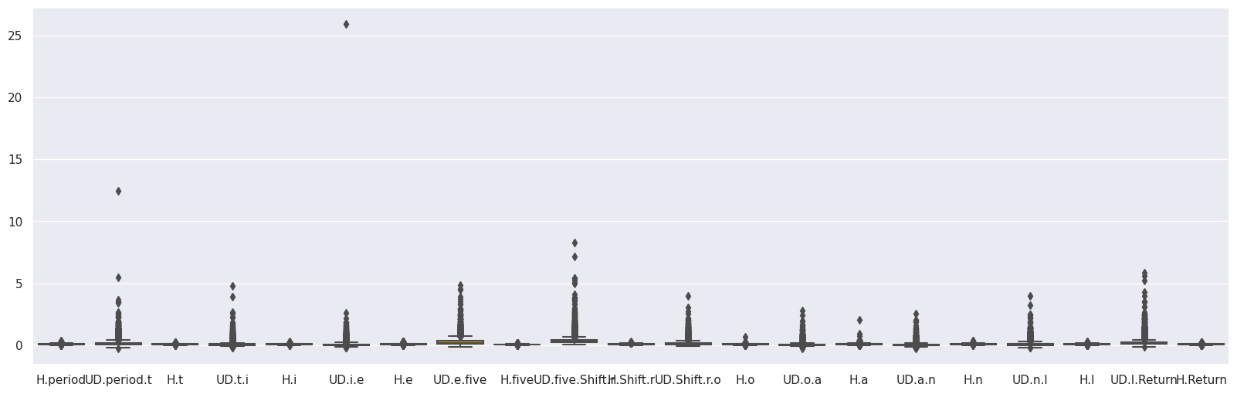
|  |  |  |
| --- | --- | --- |
| Feature #1 | Feature #2 | Correlation Coefficient (r) |
| DD.period.t | UD.period.t | 0.980 |
| DD.t.i | UD.t.i | 0.968 |
| DD.i.e | UD.i.e | 0.978 |
| DD.e.five | UD.e.five | 0.993 |
| DD.five.Shift.r | UD.five.Shift.r | 0.995 |
| DD.Shift.r.o | UD.Shift.r.o | 0.963 |
| DD.o.a | UD.o.a | 0.971 |
| DD.a.n | UD.a.n | 0.938 |
| DD.n.l | UD.n.l | 0.977 |
| DD.l.Return | UD.l.Return | 0.990 |



* It is also noted that “Up-to-Down” values can be negative. The range of “UD.period.t” is from -0.24 to 12.45.



* All the “Up-to-Down” features have outliers. Will run experiments to know their effects on the model accuracy.



* All numerical features of the dataset were scaled to ensure that the features have comparable magnitudes. Jacob has used Min-Max scaler while Allen has used Standard scaler. Experiments will be run to study the effect of type of scaler on the model accuracy.

## Feature Engineering

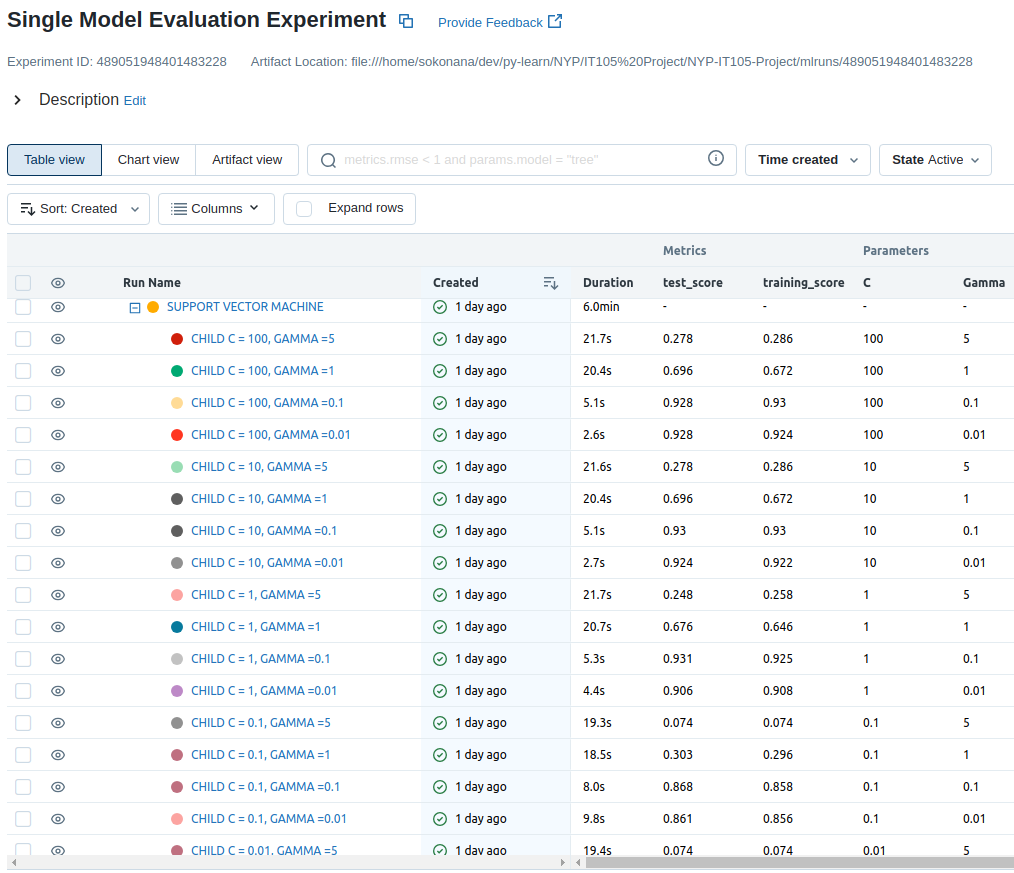
* Generating extra attributes is achievable through the calculation of the ratio between the "Hold" time and the total time for all characters present in the password.
* Experiments with engineered features will be done then to see their effects on model accuracies.

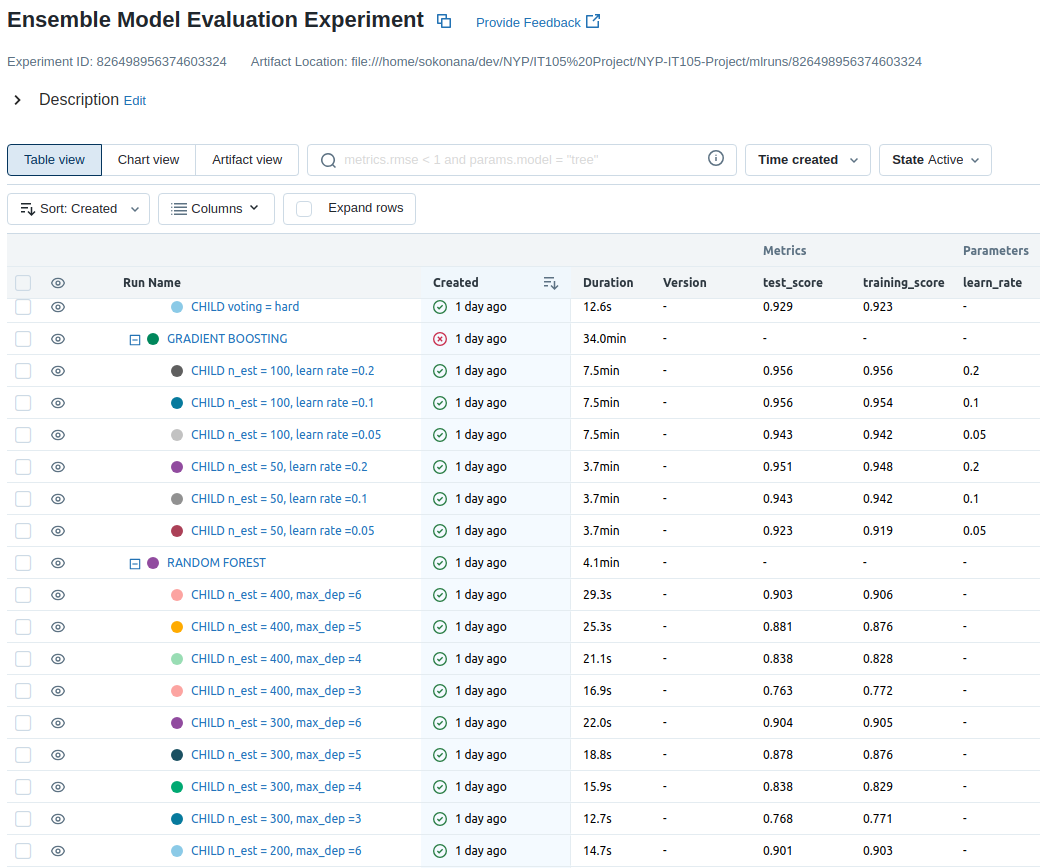
## Modelling and experimental results

* Experiments were conducted with various models with hyper tuning and the results are summarized below.



* The best model accuracy obtained so far is with ensemble Gradient Boosting (with the default base classifier Decision Tree).
* Conducting experiments is a time-consuming process due to the significant amount of machine time needed for hyperparameter tuning.
* Experiment runs are logged using “MLFlow” as shown below.





# Conventional Machine Learning with hybrid dataset

* Currently performing data collection
* EDA, Feature Engineering, Visualization, and modelling will be done based on the model that has provided best accuracies (train and test) with the benchmark dataset.

# Further actions

* Compare the accuracies obtained by Allen and Jacob.
* Experiment with outliers, scalers, and feature engineering.
* Complete the data collection from 2 new users.
* EDA, feature engineering and visualization with hybrid dataset.
* Model selection and hyperparameter tuning with hybrid dataset.
* Model deployment and inference.

# Contributions

Jacob:

* Sourced initial c code for “Key-Sniffer.
* Data visualization and Model Evaluation with Decision Tree, K-Nearest Neighbors, Support Vector Machine, Gaussian Naive Bayes, Logistic Regression, Random Forest, and Gradient Boosting.
* Platform: Google CoLab

Allen:

* Upgrading and fine tuning of “Key-Sniffer” program to collect keystroke data.
* Data visualization and Model Evaluation with Decision Tree, K-Nearest Neighbors, Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest, Gradient Boosting, and Voting Classifier
* Platform: Jupyter Notebook

# References

[1] Keystroke Dynamics - Benchmark Data Set: <http://www.cs.cmu.edu/~keystroke/>

[2] KDA on benchmark: <https://www.kaggle.com/code/ashusrivastava/kda-on-benchmark>

[3] Keystroke Dynamics Analysis and Prediction w/ XGB: <https://www.kaggle.com/code/kartik2112/keystroke-dynamics-analysis-and-prediction-w-xgb>

[4] Keystroke Dynamics Analysis and Prediction — Part 1/2 (EDA): <https://towardsdatascience.com/keystroke-dynamics-analysis-and-prediction-part-1-eda-3fe2d25bac04>