

# Configuration Manual

MSc Academic  
MSc in Cyber Security

Lakshmi Bhargav Jetti  
Student ID: X20113986

School of Computing  
National College of Ireland

Supervisor: Prof. Ross Spelman

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** Lakshmi Bhargav Jetti  
**Student ID:** X20113986  
**Programme:** MSc in Cyber Security **Year:** 2020-2021  
**Module:** Academic Internship  
**Supervisor:** Prof. Ross Spelman  
**Submission Due Date:** 16/08/2021  
**Project Title:** User Authentication Based on Keystroke Dynamics using Multi-Layer Perceptron  
**Word Count:** 570 **Page Count:** 6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Lakshmi Bhargav Jetti

**Date:** 16/08/2021

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual: Keystroke Dynamics using MLP

## 1. Introduction

This manual gives a detailed information regarding all necessary hardware and software setup required to build the entire system from scratch. The configuration manual will help in replicating the research done by using a more practical way. We are going to evaluate the complete working of the system with its user interface. (That is, how a user will interact with our system using the system user interface).

The Configuration Manual will be divided into six sections excluding this introduction part.

- Environmental Setup
- Libraries Required
- Dataset
- User Interface
- Implementation
- Code Repository

## 2. Environmental Setup

### 2.1 Hardware Requirements

- 8GB RAM.
- 250 GB HDD.
- 1.6 GHz Intel. Core i5

### 2.2 Software Requirements

- Windows 10
- Python 3.6.3

### 2.3 Programming Prerequisites

- Python (Version 3.6.3)
- Visual Studio Code 1.40.0

### 2.4 Python Environment setup

The project was implemented using python language, so Visual Studio Code is used.

It comes with a pre-installed set of environments. So Visual Studio Code was used for the overall development of this research [1].

### 3. Libraries Required

All the libraries required for building this research project are mentioned in table 1 along with commands to import them. It is required to download some libraries before using it.

pandas	import pandas as pd
numpy	import numpy as np
sklearn	Import sklearn as sk
matplotlib	import matplotlib.pyplot as plt
metrics	from sklearn import metrics
preprocessing	from sklearn import preprocessing
Keras	Keras for MLP Layers
Sklearn	KNeighborsClassifier

#### **Pandas:**

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc [2].

#### **Numpy:**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed [3].

#### **Sklearn:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib [4].

## Matplotlib:

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPython or Tkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also [5].

## 4. Data Set

The dataset so obtained for keystrokes sets a CMU benchmark [35], consists of keystroke parametric functions (H, DD and UD) collected from 51 users.

Data cleaning, filtering, and removing unwanted columns was performed using Visual Studio Code with Python Language. Csv data of keystrokes was taken as input for further processing.

### 4.1 Load Data

The Data preprocessing was performed using Python in Visual Studio Code environment. The initial task involved importing the cleaned data in pandas data frame and check whether the data is imported correctly. The code snippet for the same can be found under the heading load Data. And Reducing & Aggregating the columns into H, DD, UD

```
def load_data():
    df = pd.read_csv('data/DSL-StrongPasswordData.csv')

    H_columns = [col for col in df.columns if col.startswith('H')]
    DD_columns = [col for col in df.columns if col.startswith('DD')]
    UD_columns = [col for col in df.columns if col.startswith('UD')]

    data = {}
    data['total'] = df.drop(columns=['subject', 'sessionIndex', 'rep'])
    data['H'] = df[H_columns]
    data['DD'] = df[DD_columns]
    data['UD'] = df[UD_columns]
    data['pca3'] = pd.DataFrame(PCA(n_components=3).fit_transform(data['total']))
    data['pca10'] = pd.DataFrame(PCA(n_components=10).fit_transform(data['total']))

    return data, df['subject'].values
```

## 5. Results:

```
=====KNN Model=====
Accuracy for total : 0.69 (+/- 0.25)
Accuracy for H     : 0.65 (+/- 0.12)
Accuracy for DD    : 0.55 (+/- 0.20)
Accuracy for UD    : 0.60 (+/- 0.23)
Accuracy for pca3  : 0.19 (+/- 0.09)
Accuracy for pca10 : 0.58 (+/- 0.22)
```

### =====Grid Search CV with KNN=====

Below are the ranked models based on best feature selection

```
Model with rank: 1
Mean validation score: 0.696 (std: 0.008)
Parameters: {'n_neighbors': 3}
```

```
Model with rank: 2
Mean validation score: 0.665 (std: 0.008)
Test accuracy : 0.7220588235294118
```

```
Model with rank: 3
Mean validation score: 0.644 (std: 0.007)
Parameters: {'n_neighbors': 7}
```

```
=====MLP Model=====
Data : total, Nodes : 300
  Accuracy for train set : 0.9871
  Accuracy for test set  : 0.8973
Data :   H, Nodes : 300
  Accuracy for train set : 0.7573
  Accuracy for test set  : 0.6260
Data :  DD, Nodes : 300
  Accuracy for train set : 0.7688
  Accuracy for test set  : 0.6392
Data :   UD, Nodes : 300
  Accuracy for train set : 0.8962
  Accuracy for test set  : 0.7225
Data :  pca3, Nodes : 300
  Accuracy for train set : 0.2106
  Accuracy for test set  : 0.2130
Data :  pca10, Nodes : 300
  Accuracy for train set : 0.9381
  Accuracy for test set  : 0.7350
```

## 6. References

1. Python Release Python 3.6.3 [online] (2021) Python.org, available: <https://www.python.org/downloads/release/python-363/> [accessed 16 Aug 2021].
2. Pandas - Python Data Analysis Library [online] (2021) available: <https://pandas.pydata.org/> [accessed 16 Aug 2021].
3. Overview — NumPy v1.21 Manual [online] (2021) available: <https://numpy.org/doc/stable/> [accessed 16 Aug 2021].
4. Scikit-Learn: Machine Learning in Python — Scikit-Learn 0.24.2 Documentation [online] (2021) available: <https://scikit-learn.org/stable/> [accessed 16 Aug 2021].
5. Matplotlib: Python Plotting — Matplotlib 3.4.3 Documentation [online] (2021) available: <https://matplotlib.org/> [accessed 16 Aug 2021].