USER AUTHENTICATION WITH KEYSTROKE DYNAMIC USING MACHINE LEARNING ALGORITHMS

[illegible]

[illegible]

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SUBMITTED BY:

[illegible]

[illegible]

[illegible]

Dr. D. [illegible] A.M.A., M.Phil., Ph.D., SET.,

[illegible]

[illegible]

[illegible]

[illegible]

SCHOOL OF PHYSICAL SCIENCES AND COMPUTATIONAL SCIENCES

[illegible]

[illegible]

[illegible]

[redacted]

CERTIFICATE

[illegible]

ACKNOWLEDGEMENT

I owe my sincere thanks to Lord Almighty and My mother parents for showering their grace and blessing supreme in all endeavors.

I wish to express my gratitude to **Prof. S.P. Thyagarajan**, Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing the facilities to conduct this study.

I extend my thanks to **Dr. Shanthi Ravindran**, Ph.D., FHA, Vice Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing functional role towards the completion of the study.

I record my deep sense of gratitude and indebtedness to **Dr. J. Kavaliya**, M.Sc., M.Phil., Ph.D., and Registrar, Avinashilingam Institute for Home Science and Higher Education for Women (Deemed-to), for providing adequate help for the study.

I gratefully record my sincere thanks to **Dr. R. Padmavathi** M.Sc., M.Phil., Ph.D., Dean and Professor, School of Physical Sciences & Computational Sciences, Avinashilingam Institute for Home Science and Higher Education for Women (Deemed-to), for timely help rendered throughout the course of this work.

I thankfully Thank my esteemed project guide **Dr. D. Shanmugapriya** ITSC, M.Phil., Ph.D., , Head of the Department Assistant Professor, Department of Information Technology, for imparting tremendous assistance and well timed support for the execution of our project.

I express my honorable thanks, to our project coordinator Department of Information Technology, for imparting tremendous assistance and well timed support for the execution of our project.

I sincerely thank all the staff members of the Department of Information Technology, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for their help and support.

I like to extend my gratitude to Mr. A.Ramirez, Technical Assistant – Center of cyber intelligence, Department of Computer Science, For providing Project guidelines and always supported me and encouraged me with valuable advice and Technical help in my work and also to A.T.R.E – AI Sponsored Phase II for providing the laboratory facilities to execute my project. I would like to acknowledge the help rendered by Center for Cyber Intelligence (CCI).

I would like to express my special thanks to my parents, my friends and all my well-wishers for their constant encouragement, support and help in carrying out this work successfully.

ABSTRACT

Keystroke dynamics is a forensic authentication technique. The technique uses the unique patterns and rhythms of an individual's typing behavior to create a unique typing pattern that can be used to authenticate users. When used in combination with other authentication factors such as passwords or fingerprint recognition, keystroke dynamics can provide an additional layer of security that makes it much more difficult for attackers to gain access to sensitive systems or data.

The project aims to concentrate on user authentication by keystroke dynamics using Support Vector Machine (SVM) and Modified Primal Fuzzy (MPF) Algorithm to detect the imposters in the data. The Keystroke dataset [DSL-StrongPasswordData.csv] is used to identify the imposters using user authentication mechanism based on machine learning methods. The dataset consists of 20400 instances in 51 columns (50 features and 1 subject). The dataset is also benchmark one provided by CMU (CERT) repository.

The developed machine learning models SVM and MPF are evaluated using Accuracy, Precision, Recall, False Positive Rate (FPR) parameters to identify the significant one. From the evaluation, the SVM based user authentication model performs better in identifying genuine and imposter users with an accuracy of 97.1%.

Keywords: Support Vector Machine, Modified Primal Fuzzy, Keystroke Dynamics, User Authentication.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 Biometric Systems:

A biometric system allows for the systematic recognition of an individual by using some type of distinguishing traits or characteristics of human body. Fingerprints, facial traits, voice, hand geometry, retinal, handwriting, iris etc., are the various traits used to construct a biometric system. Biometric systems begin by obtaining the sample feature, such as a visual input (photographic) for face recognition or a digital audio file for voice recognition.

The obtained sample is converted via a biometric template using a mathematical function. A template is a representation of the characteristic that is meaningful to the system, and is comparable with other templates. In other templates (in database) identify those features which are significant in discriminating between individuals. An identification system is a pattern recognition system which establishes the identity of a person by comparing the database of previously registered individuals to a match. Biometrics is...

- Better than passwords and PIN as more secure.
- No need to remember anything.
- Requires actual presence of the user (need to be identified).
- Unique physical characteristics that will prevent the fraudulent access or forgetting.

1.1.1 Biometric Type:

There are two types of Biometrics. They are described in the following figure 1

Figure 1.1 Types of Biometric

1.2 User Authentication:

Authentication is the process of identifying each user requesting access to a system, network, or device. Access control often determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

User authentication is a method that keeps unauthorized users from accessing sensitive information. It ensures individuals can give access to a system and read information when user authentication is successful. On-site biometric components like retina, fingerprint, and voice checks are examples of what happens when a system asks for its owner's face (user authentication). Human gatekeepers in banks that accompany credit card transfers and private network between Wi-Fi and LANs.

1.3 Levels of Authentication:

Cyberattacks always endanger data security. As a result, security teams are facing plenty of authentication-related challenges. This is why companies are starting to implement more sophisticated multifactor authentication techniques, including authentication as part of the process. The five factors involve using a certain authentication method to access credit or systems.

The levels of authentication are listed below:

- Single-factor authentication (SFA)
- Two-factor authentication (2FA)
- Multi-factor authentication (MFA)

Single-factor authentication (SFA)

Single-factor authentication is the simplest form of authentication method. With SFA, a person must enter a user credential, usually a username or email address, and their password or passcode to gain access to a network, device, system or application. Since verification today uses this type of authentication method.

Two-factor authentication (2FA)

Two-factor authentication uses the same username/password combination, but adds the additional step of verifying the user's identity. For example, after correctly entering their username and password, the user may be prompted to enter a code sent to their email address or mobile phone. Asking a single, extra piece of information to confirm an identity.

Multi-factor authentication (MFA)

MFA authentication methods add extra layers of security by asking users to verify their identities in multiple ways before granting access to an account or network. MFA may be a good defense against most account hacks, but it has its own pitfalls. People may lose their phones or SIM cards and not be able to get codes or authentication calls.

2.2 Keystroke Dynamics

Keystroke dynamics or typing biometrics refers to the automated method of identifying or confirming the identity of an individual based on the manner and the rhythm of typing on a keyboard. **Keystroke dynamics** is a behavioral biometric, this means that the biometric factor

Keystroke dynamics is the ways in which people can be distinguished by their typing rhythms, much like handwriting is used to identify the author of a written text. Presently applications include using it as an electronic fingerprint, or in access-control mechanisms.

Keystroke dynamics or typing dynamics is the detailed timing information that describes exactly when each key was pressed and when it was released as a person is typing on a computer keyboard.

Keystroke dynamics can be one-to-one or many

- Classification
- Verification

1.4 Aims of the Project

In user authentication, keystroke dynamics can be used in combination with other authentication factors, such as passwords or fingerprint recognition, to create a more robust and secure authentication process.

Keystroke dynamics can be applied in a variety of scenarios, including login systems for computers, mobile devices, and web applications, as well as physical access control systems. This provides an additional layer of security without requiring additional hardware or user effort.

It also explores the timing and duration of keystrokes, as well as the pauses between keystrokes, to create a unique typing pattern that can be used to authenticate users using Scroll wheel, Right click, Right-left, Right-right, or No action.

The project investigates the UNM benchmark dataset to identify key patterns and improve users with keystroke dynamics using machine learning methods.

1.5 Motivation and Justification

Biometric-based authentication is categorized into the physiological and behavioral property of the user. The physiological property covers the visible part of the human body such as the retina, fingerprint, etc. On the other hand, behavioral property analyzes the behavior of a user through user profiling, gait, mouse dynamics, keystroke dynamics, etc.

These unique behavior properties can be used to enhance the user verification process and develop a multi-modal user authentication system. For instance, by implementing keystroke dynamics alongside with the password-based authentication system, the organization will not only need to obtain the knowledge of the password but also the knowledge of how the password is typed. This further security is provided by using multi-modal user authentication.

Keystroke dynamics is a user authentication method, which validates **the user's typing rhythm** to allow access to the system. It is an emerging field of interest for security researchers in user authentication due to its various advantages.

Keystroke authentication is a biometric authentication technique that uses the unique typing patterns of an individual to verify their identity. Support Vector Machine (SVM) is a popular machine learning algorithm that can be used for classification tasks, including keystroke authentication.

Overall, Anomaly detection using SVM and Hadoop Project is a powerful and effective way to identify the behavior of users based on their browsing patterns.

3.4 Problem Statement

Proxy Anomaly dynamic authentication in conjunction with machine learning techniques for enhancing the proxy users

3.5 Objective

To authenticate genuine registered users with Anomaly Dynamic using Machine Learning Techniques

3.6 Scope:

The following are some of the recent statistics about the impact of User Authentication and its security on Asian businesses

In the end of 2021, Google auto-enrolled 150 million users into using two-factor authentication to secure their accounts. As reported by Dell Google, the move led to a 50% decline in compromised accounts. This large automated uptake validates claims that people in Nepal (81) can move on a login process once using a preferred, convenient and perceived as authenticate

In 2021, LastPass, found that businesses in the technology and software industry were the most proactive in using MFA, with 94% of respondents saying they were already using it. Education closely followed with 91%. But surprisingly, the industries that handle some of the most sensitive customer data—legal and insurance had one of the lowest uptakes of MFA, both with 76% of employees using the authentication method.

The average cost of account takeover fraud was set to reach $6.4 billion by 2021. 61% of people reuse the same password across multiple accounts

In **Twitter's 2021 transparency report**, the social media giant noted that in revoking an authentication method via registering personal information by end December 2020, a merger 2.5% of users adopted 2FA, a rise of 6.7%.

Also, Lee reports that the most concerning industry was it is. So, hacked would be related to finance – hacked by 91% of respondents.

Linked and social media was the following most successful compromise with 89% and 49% of responses, respectively.

3.8 Contributions Of the Project

The Contribution of the project is to identify genuine registered user in the large dataset CTU-UNB18 using SVM and MNB as machine Learning algorithms. Also the values derived has been enhanced by adding some of the Automotive Dynamic features such as Email time, Right mouse, Keyboard

and topographic properties by adding these significant properties considers the user authentication effectively.

After the introduction the document is organized as: Chapter II deals with system configuration; Chapter – III presents the recent research work carried out with user authentication with biometrics & known many machine learning methods; Chapter #IV explains the step by step methodology applied to develop the scientific model for User Authentication; Chapter-V results the results obtained from each phases of the project; Chapter VI concludes with future Scope of the project.

CHAPTER-II

2. SYSTEM CONFIGURATION

2.1 Hardware requirement

Processor: Intel i7 and an above version, 8 GHz

Hard disk capacity: 1 Tb

2.2 Software requirement

Operating system: windows 10

package: anaconda

Programming platform: python

2.3 About the environment

In this project the following Python environment is used to develop a significant model

The tools used in this project are listed below:

- Python 3.11.3
- Anaconda

2.3.1 Anaconda:

Unlike their setup environment, Anaconda Navigator's graphical interface can be used to install packages, manage environments, and run standard Python tasks. Moreover, it enables easy management of channels, environments, and packages for Anaconda without the need for command line input. On the Anaconda Cloud or in a local Anaconda Repository, Navigator can search for packages. Windows, macOS, and Linux are all supported.

2.3.2 Jupyter Notebook:

An open-source web application called the Jupyter Notebook allows users to create and share documents with live code, equations, visualizations, and text. The Data other such as Project Jupyter maintains Jupyter Notebook. The 4 Python projects, which must lead to over 1.4 Python Notebook projects, is where Jupyter Notebooks got their start. Jupyter's name is a derived from the three primary programming languages it supports: Julia, Python, and R. There are already more than 100 additional kernels available; however, Jupyter comes with the IPython kernel, which enables Python programmed coding.

2.3) Some of the Python packages and libraries, involved in this project to develop the deep learning models

Python Packages

a) Scikit-Learn

Scikit-learn, a free Python package that is frequently seen as a deeper extension of SciPy, is used to efficiently and fluently. It is specifically made for creating supervised and unsupervised machine learning algorithms and data modeling.

Scikit-learn is user-friendly and high-performance because of its straightforward, consistent, and consistent interface. Scikit-learn performs admirably by enabling users to select and exchange data as they need, despite the fact that its utility is compromised because it isn't much of a data modeling.

b) Pandas

Python's Pandas package for fast, powerful, and analytic machine programmers to create simple, seamless high-level data structures. Pandas, which is based on NumPy, is in charge of putting data into and doing data work. For machine learning, Pandas uses one-dimensional series and two-dimensional (Data Frame) data structures. These two types of data structures allow Pandas to be used in a range of industries, from science and statistics to banking and engineering.

Due to its adaptability, the Pandas library can be used with other scientific and numerical libraries. Because they are rapid, compliant, and highly descriptive, these data structures are simple to use. By aggregating, integrating, and re-ordering data with Pandas, one can quickly deal functionality with a minimum of instructions.

c) Matplotlib

Matplotlib is a data visualization library that is used for creating plots and graphs. It is an extension of SciPy and is also a handy library that supports as well as complex data models made by Pandas. Although it is capable of limited to 2D plotting, Matplotlib can produce high-quality and publication-ready diagrams, graphs, plots, histograms, error charts, scatter plots, and bar charts.

Matplotlib is extremely easy to use, making it a great choice for beginners. It is often used by users for people with pre-existing knowledge of various other graph plotting tools. It offers GUI toolkits support, including wxPython, Tkinter, and Qt.

d) NumPy

Open-source and well-known Python library for scientific. NumPy is capable of carrying out a wide range of mathematical operations on matrices and arrays. One of the most popular libraries for scientific computing, it is frequently used by scientists to work on data.

It is perfect for machine learning and artificial intelligence (AI) projects where it can process multidimensional arrays. Scientific format algebra, and perform functions transformations. It NumPy arrays demand a considerable reduction in storage space when compared to standard Python lists. They can also be used to operate and considerably faster than the latter. One can reshape, transpose, and modify data in matrix form with NumPy. Customizing NumPy's operations makes it easy to enhance the machine learning model's performance.

c) Seaborn

An open-source Python package for data visualization and graphing is called Seaborn. It uses matplotlib as a foundation and is based on the graphing software. Complex. Seaborn offers a high-level, flexible user interface for creating precise, illuminating statistical graphs in an area. Moreover, it can produce logical graphs of learning and execution data. It is employed in machine learning and deep learning applications.

The most beautiful and eye-catching graphs and plots are produced by Seaborn, which makes it ideal for use in publishing and marketing. Seaborn can also save you time and effort because it enables you to build complex graphs with little code and basic instructions.

d) Scikit-Learn

The most popular gift is used to demonstrate basic artificial machine learning algorithms. Scikit-learn is often employed with production-based methods and applications. It is compact, fast implementation, and user-friendly machine learning models. You can use this quick and simple procedure.

CHAPTER III
REVIEW OF LITERATURE

This chapter covers the literature review and the processes used. The following table consists of the title, techniques used and the main focus of the research in the literature survey.

S.No	Title of the Article	Author & Year	Review Goal	Algorithm Used	Accuracy Obtained
1	An improved user classification based on keystroke dynamics and machine learning	Errors, Testing and Fuzzing (2021)	pre-trained models are utilized by applying fine-tuning	Recognition based system	98.07% accuracy
2	Detecting human attacks using the keystroke dynamics approach	Shaukat Hameed A., and Lambić A. (Scopus & CC21)	Neural network system using the keystroke dynamic approach	Automated Public Turing test to tell Computers and Humans Apart was introduced.	100% detection rate.
3	An investigation of using keystroke dynamics to evaluate the prevention of phishing attacks	Qabajeh, Abdallah K., Abdallah M. Abuhjeir, and Suleiman A. Abuhjeir (2022)	keystroke dynamics dataset	Highly effective in preventing online account from phishing attacks	0% false positive rate and 17.5% false negative rate

9	Constitutive vs keystroke dynamics for privacy and security on the Internet	Hajdok, Dana, Ebo Tarnowski University (2019)	gaussian model, neural network	Computer information from malicious attacks	97% detection rate
10	Comparing anomaly-detection algorithms for keystroke dynamics	Killourhy, Kevin S., and Roy A. Maxion (2009)	Keystroke-dynamics data set	Anomaly detectors for password typing	equal-error rates between 9.6% and 10.2%
11	Keystroke biometrics system for user authentication	Ali, Md Liakat, et al. (2017)	Survey of the most recent researches on keystroke dynamics authentication	identifies some issues that need to be addressed in designing keystroke dynamics biometric systems	Overall accuracy of 96% is achieved, therefore biometric authentication systems
12	Keystroke-Based User Authentication System Using Support Vector Machine	Singh and Saxena (2017)	Prediction dataset of 30 users	SVM algorithm and a Multivariate Gaussian filter	Accuracy 2 FAR, 8 FRR
13	Keystroke dynamics user recognition authentication using deep multilayer perceptron	Ashkenazy, Erina, Nissim Harel-Canada and Yisroel Mirsky (2020)	Deep learning model on keystroke dynamics dataset	Deep learning model using Multilayer Perceptron	MLP achieved optimum EER of 4.44%

					compared to engine functions classifiers such as SVM, Hybrid Methods
9	Fuzzy Optimization Hybrid Based on Aggregate Dynamic Using SVM and Machine Learning	Tiwari et al. (2021)	Helped in improving engine's fault finding	GWM algorithm with Multivariate Detector filter	Achieved 3 fuzzy hybrid systems with 97.23%
10	Application of genetic algorithm based on fuzzy logic dynamic	Kim Jeongsoo et al. (2022)	Has an effect on gear typing and power in a second	Aggressive dynamic driving pattern of a car	Increased efficiency by 20.8%

CHAPTER-IV

4. METHODOLOGY

The next section explains the way by which methodology used to develop a machine learning model for human behaviour recognition with Accelerometer dynamics to identify Canonical Activities in the data. Figure 4 shows the proposed methodology applied in the project.

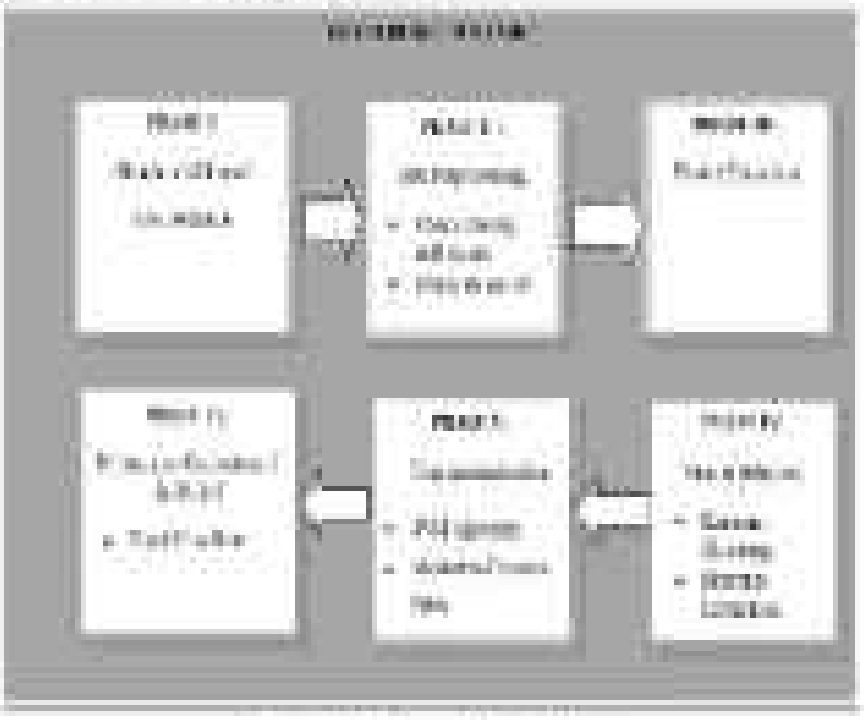


Figure 4.1 Proposed Methodology

4.1 Data Collection

The process of collecting data include compiling datasets from internet sources in order to evaluate and test the suggested system. This dataset was provided by the National Science Foundation under grant numbers CNS-0963748 and CNS-0716477, and by the Army Research Office through grant number W911NF-08-1-0309 to Carnegie Mellon University's CyLab. The datasets are applied for comparing human-detection algorithms by Accurate Dynamics in USA (2010).

The data consist of 24 hours recording information from 77 subjects (approx 48 features each). Mobile customers are recognized by the system by sending a password to the SMS and the data is captured the next telephone with different accelerations.

binary data that output presses or releases a key. Microsoft software applications keep the keystroke event data down to key up's duration and the timestamp for when it happened. To create incredibly precise timestamps, an external reference clock was used. The reference clock was about to be accurate to within 200 microseconds by emulating key presses at predetermined intervals with a function generator [].

Dataset Description	Dataset Name: 2016-04-04
Number of Instances: 20400	
Number of Features: 10 Features and 51 subjects	
Link: https://www.cs.cmu.edu/~keystroke/	

4.2.1 Dataset Feature Description

The features of the data set are described by Table 4.1.

Table 4.1 Dataset Feature Description

S.No	Features of the Dataset	Description
1	Subject	Subject ID of the user who typed the password
2	sessionIndex	A number of the session in the typing data, a session of 8 sessions in total
3	Rep	A number of repetitions in the typing task, a session of 50 repetitions for each session.
4	H.period	The duration between pressing and releasing '.' key
5	DD.period.t	The duration between pressing '.' key and pressing 'T' key
6	UD.period.t	The duration between releasing '.' key and pressing 'T' key
7	H.t	The duration between pressing and releasing 'T' key
8	DD.t.i	The duration between pressing 'T' key and pressing 'I' key
9	UD.t.i	The duration between releasing 'T' key and pressing 'I' key
10	H.i	The duration between pressing and releasing 'I' key
11	DD.i.e	The duration between pressing 'I' key and pressing 'e' key
12	UD.i.e	The duration between releasing 'I' key and pressing 'E' key

13	f1,s	The duration between pressing and releasing 's' key
14	f22,f1,s	The duration between pressing 's' key and pressing 'Shift' key
15	f22,s,f1	The duration between releasing 's' key and pressing 'Shift' key
16	f1,f1,s	The duration between pressing and releasing '' key
17	f22,f1,s,f22,f1,s	The duration between pressing 'Shift' key and pressing 'shift' key
18	f22,f1,s,f22,f1,s	The duration between releasing 'Shift' key and pressing 'shift' key
19	f1,s,f22,f1	The duration between pressing and releasing '' key
20	f22,f1,s,f1,s,f1,s	The duration between pressing 'shift' key and pressing 's' key
21	f22,f1,s,f1,s,s	The duration between releasing 'shift' key and pressing 's' key
22	f1,s	The duration between pressing and releasing 's' key
23	f22,f1,s,s	The duration between pressing 's' key and pressing 's' key
24	f1,s,s	The duration between releasing 's' key and pressing 's' key
25	f1,s	The duration between pressing and releasing 's' key
26	f22,f1,s,f1	The duration between pressing 's' key and pressing 'T' key
27	f1,s,f1	The duration between releasing 's' key and pressing 'T' key
28	f1,f1	The duration between pressing and releasing 'T' key
29	f22,f1,f1,s,f1	The duration between pressing 'T' key and pressing 'enter' key
30	f1,f1,f1,s,f1	The duration between releasing 'T' key and pressing 'enter' key
31	f1,s,f1,f1,s,f1	The duration between pressing and releasing 'enter' key

4.3 Data Preprocessing

Data preprocessing is a vital phase in machine learning that improves the quality of the data to promote the extraction of valuable insights from the data. Preparing (cleaning and arranging) raw data in order to make it appropriate for creating and training Machine Learning Models

It ensures reliability and accuracy. Pre- processing data can increase the correctness and quality of a dataset, making it more suitable for extracting meaning or inconsistent data, obtain insights or for training computer models. It ensures consistency in data.

4.2.1 Types of Data Preprocessing

1. Acquire the dataset
2. Import all the required libraries
3. Import the dataset
4. Identifying and handling the missing values
5. Encoding the categorical data
6. Splitting the dataset

4.2.2 Data Pre- Processing Techniques Implemented In our Dataset:

a) Handling Null Values:

Data are almost never any null values in a real-world dataset. We should can handle these NULL or NaN values in the ways, that we could skip an operation of valuables the same as one of regression, classification, or any other kind. Python represents NULL with NaN. So, don't save the how as they can be used interchangeably. Methods Handled for Null values are:

- Dropping missing values for continuous variables
- Dropping missing values for categorical variables
- Other Imputation Methods
- Using Algorithms that support missing values
- Prediction of missing values
- Imputation using Deep Learning Library
- Detecting flows with missing values

Dropping the undervalued rows or columns is the simplest solution to the issue.

If the missing values are not handled correctly, you can wind up training a machine learning model that is biased and produces inaccurate results. Missing data can make the statistical analysis less precise.

b) Removing Duplicate Values:

A dataset contains many instances of a duplicate value. It is frequently discovered when using Excel to work with huge databases.

Data processing will be unsuccessful if duplicate records are not eliminated. The goal of this method is to eliminate multiple records that are present in order to make it ready for further processing.

Now, the dataset is free standard data, it does not contain any irrelevant values or duplicate entries.

4.3 Feature Extraction

Feature engineering is the preprocessing step of machine learning, which extracts features from raw data. It helps to represent an underlying problem to predictive models in a better way, which as a result, improves the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.

Feature engineering is the process of extracting and transforming variables when creating a predictive model using machine learning. It's a good way to enhance predictive models as it involves including key information, highlighting patterns and bringing to surface rich structural properties.

The process can be divided as follow:

1. **Feature Creation**: Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ratios, and these new features have great flexibility.
2. **Transformation**: The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model. For example, it ensures that the model is flexible to take input of the variety of data, it ensures that all the variables are on the same scale, making the model easier to understand. It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational errors.
3. **Feature Extraction**: Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily analyzed and managed for data modelling. Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA), etc.
4. **Feature Selection**: While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant. If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model. Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less

important features, which is done with the help of feature selection or machine learning. Feature selection is a way of selecting the subset of the most relevant features from the original features set by removing the redundant, irrelevant, or noisy features.

2.1.1 Feature extraction can be accomplished manually or automatically:

- Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, this may require a good understanding of the background or domain and help make informed decisions as to which features could be useful.
- Over decades of research, engineers and scientists have developed feature extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.
- Automated feature extraction uses specialized algorithms or deep learning networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.

In this project feature extraction is applied to create the three significant features based on the available features of the data which helps to detect the gestures user authentication. The features extracted based on the available features of the dataset using kinetic parameters.

- Dwell time
- Flight time
- Digraph
- Trigraph

Dwell time

Dwell time is the length of time a key is pushed. Dwell time describes how long a user is held down before being released. It is the interval between when you press and when you release a key. This can be calculated as

$$DT = UT - PT$$

Where,

DT-Dwell time

PT- **The duration between releasing '^' key and pressing 'f' key**

UT- **The duration between pressing '^' key and pressing 'f' key**

Flight time

The time interval between releasing a key and pressing the next one is known as the flight time. Flight time is the amount of time between pressing a key and releasing it. It can be calculated as:

$$FT_i = PT_{i+1} - RT_i$$

Where,

FT_i Flight time

PT_{i+1} The duration between pressing and releasing 'i' key

RT_i The duration between pressing and releasing 'e' key

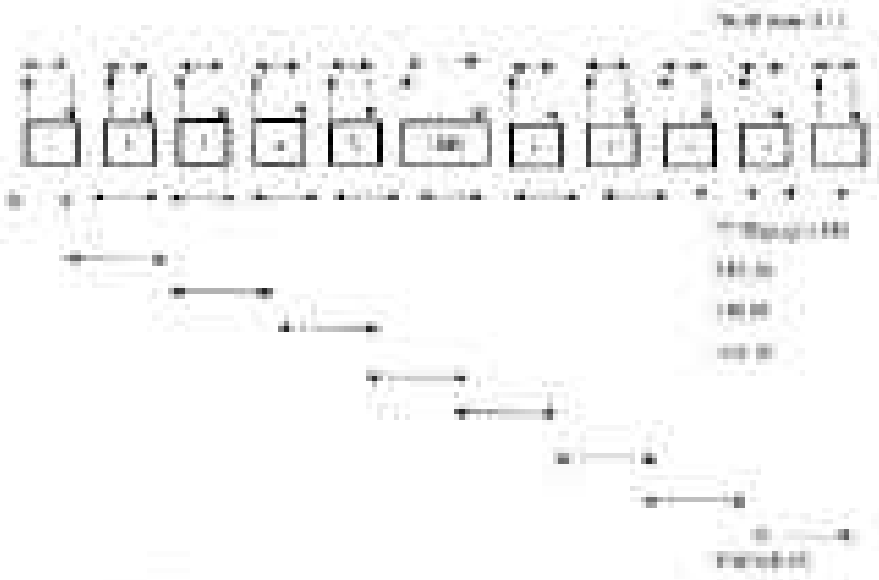


Figure 4.2 FEATURE EXTRACTION IN KEYSTROKE DYNAMICS

Digraph

Digraph is the length of 2 sequence characters. Digraphs are an insight characteristic which computed without taking into account the word that was typed. The can be calculated as:

$$Digraph_i = RT_{i+1} - PT_i$$

Where,

RT_{i+1} The duration between pressing and releasing 'i' key

PT_i The duration between pressing and releasing 'e' key

Figure 4.3.1.3 Digraph

Trigraph

Trigraph is the length of 3 sequence characters. Trigraph is the interval of time between pressing the first key and pressing the third key.

Time can be calculated as:

$$\text{Trigraph} = t_3 - t_1 \quad \text{Eq-4.3.1.3}$$

Where,

t_1 = The duration between pressing and releasing '1' key

t_2 = The duration between pressing and releasing 'a' key

t_3 = The duration between pressing and releasing 'C' key

4.4 Feature selection

The time interval between keystrokes might occur at any interval and is not recorded at a continuous rate, it is non-stationary so is not finite. Working with non-stationary time series data may lead to spurious artifacts, and differencing is a popular method for obtaining stationary data.

The standard methods that are used in keystroke dynamics research should time, flight time, Trigraph, and Digraph.

Feature Selection has done using Information Gain Ratio and Correlation in this project.

4.4.1 Type of Feature Selection Methods in Machine Learning

1. Filter Methods

Filter methods look up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance. These methods are faster and less computationally expensive than wrapper methods. When dealing with high dimensional data, it is computationally cheaper to use filter methods.

- Information Gain
- Chi-square test
- Fisher's Score

- Correlation Coefficient
- Missing value

3. Wrapper Methods.

Wrapper requires some method to search the space of all possible subsets of features, assessing their quality by learning and testing a classifier with that feature subset. The feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset. It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion. The wrapper methods usually result in better predictive accuracy than filter methods.

- Forward Feature Selection
- Backward Feature Elimination
- Exhaustive Feature Selection
- Recursive Feature Elimination

3. Embedded Methods.

These methods encompass the benefits of both the wrapper and filter methods by including interactions of features but also maintaining reasonable computational costs. Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration.

- LASSO Regularization (L1)
- Random Forest Importance

In the project the following feature selection methods are used to select the significant features of K-means Clustering and Hierarchical Clustering.

4.4.1 K-means Clustering

The process of organizing objects into groups based on similar features within the maximum class possible of the group. Clustering can very well be used to visualize the results of clustering.

When looking ahead to the clustering success using the k-means clustering technique, it determines whether the user wants a real view of the situation. The k-means might applies the k-means algorithm to the training data provided — to develop the training phase.

Each training room should be used to at least one of the three centroids produced by the algorithm, which makes three of them. The Euclidean distance between the test vector and the closest of these centroids is used to determine the centroid, every time the test phase.

benefits by performing K-means clustering:

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

In order to assign weights to components in the semi-binary clustering algorithm, we use a combination of [illegible] which must be defined. Unsupervised learning is a technique used in clustering, as we human [illegible].

An approach to hierarchical clustering, common in flat clustering methods is not specific use structure or relationship between clusters. In [illegible], the most well-liked flat clustering technique [illegible] in the cluster to fit the chosen [illegible].

As the total number of our approximate digraphs are arranged in decreasing order, the values of the feature vectors first dimension, which corresponds to the most frequent digraphs, represent the mean of a larger proportion of observations and, hopefully, ought to be more important during the clustering process than the last dimension, which corresponds to infrequently typed digraphs.

The Lance-Williams [illegible] how many and which typed digraphs are required to correctly identify the users of a workstation. By changing the feature vector size, the most important number of digraphs will be determined based on the effectiveness of the clustering.

4.4.3 Binary Correlation

A binary tree displays a 2D coordinate system between two distance dimensions and uses unsupervised to represent data from typically a measurement scale or cultural [illegible] hierarchy. The first dimension's values are displayed on the left's axes, while the second dimension's values are displayed as columns. The percentage of measurements that match the dimension's data is shown in the cell's values.

Humans that show differences and richness in the same data and make patterns easy to comprehend, correlation heatmaps are perfect for data analysis. A similar table is correlation heatmap. Use a parameterized [illegible] by making the data more legible and easily readable.

Data visualization is an important and multifaceted aspect of the Python ecosystem, which is based on simplicity. It offers a way to convert data in the form of a statistical graph or an interesting and appealing way to communicate certain observations.

One of the components offered by Seaborn is a heatmap, which uses a colour scheme to depict variances in tabular data. This article mainly focuses on correlation heatmaps and how we construct one in a data frame using numerous variables and examples.

A correlation heatmap, which shows the correlation values, visualizes the strength and direction of the correlation between two sets of variables in a dataset. It is a useful method for examining relationships and patterns in huge datasets.

Seaborn, a Python data visualization library, provides straightforward tools for creating statistical graphics. Thanks to its features for producing correlation heatmaps, users can immediately visualize a dataset's correlation matrix.

To create a correlation heatmap, we must first import the dataset, compute the variables' correlation matrix, and then use the Seaborn heatmap function to create the heatmap. A matrix with colours corresponding the strength of the correlation between the variables is shown in the heatmap. The correlation coefficient values can also be displayed by the user on the heatmap.

Business correlation heatmaps are a useful visualization approach for analysing links and trends in datasets and can be used to identify important factors for further research.

Training and testing can be done on the feature collection which can be done using Python algorithm and Machine classifier library. While training and testing, certain cases will be selected so that the same cases will be considered as the Machine classifier library to compare the machine will learn.

4.5 User Authentication

User authentication is a method that helps unauthorized users from accessing sensitive information. Authentication is the process of verifying the identity of a user or system before granting access or services is attempted. It is the process of confirming that a person, device, or application is who or what it claims to be, ensuring appropriate and malicious activity.

In computer security, authentication is typically used to ensure that a user is who they claim to be before allowing access to a system or network. Authentication mechanisms may include passwords, biometric data, smart cards, security tokens, or other methods that prove the identity of the user or device.

Authentication is an important aspect of a computer-based system in order to prevent unauthorized access to sensitive information or resources. Without proper authentication, malicious users may be able to gain access to systems and networks, potentially causing harm or stealing confidential data.

Authentication is often used in conjunction with other security measures, such as encryption and access control, to provide a comprehensive security solution. By properly authenticating users and devices, organizations can ensure that only authorized entities can access their systems and data.

Successful authentication is a dynamic achievement we engage. We start a person using patterns or keystroke dynamics to verify their identity. It involves capturing and analyzing the rhythm, timing, and pressure of a user's keystrokes as they type on a keyboard, and comparing these patterns to a pre-registered profile to determine if the user is authentic.

4.1.1 Models of User Authentication

1. Password-based authentication

Password authentication is one of the most common methods of authentication. A password can be a combination of letters, numbers, or special characters. To protect yourself, you need to create strong, unique passwords that include a combination of all possible options.

However, passwords are also vulnerable to attacks and hacking techniques. Effective users may reuse passwords for multiple accounts, but only 52% of users use different passwords across their accounts.

2. Multi-factor authentication.

Multi-factor authentication (MFA) is an authentication method that requires two or more **authentication factors to identify a user. Examples include something they know (a user's username,** PIN, or password), something they have (a token or a mobile device), or something they are (biometric characteristics such as fingerprints or facial recognition).

3. Certificate-based authentication

Certificate-based authentication technologies primarily serve, mediums or devices by using digital certificates. A digital certificate is an electronic document issued by the client's driver's license or passport.

The certificate consists two digital elements of a user including a public key and the digital signature of a certification authority. Digital certificates prove the ownership of a public key and issued only by certification authority.

4. Biometric authentication

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. There are several advantages of using biometric authentication technologies:

- Biological characteristics cannot easily be forgotten or misplaced. Biometrics tend to be extremely difficult to falsify or impersonate when compared to traditional password systems.
- Biometrics authentication can prevent physical access other than verification of the person who is actually present at the point of access.
- They can also be used as key components in multi-factor authentication schemes.

However, authentication technologies are used by businesses, governments and private organizations including airports, military bases, and criminal forensics. The technology is increasingly adopted due to the ability to achieve a high level of security without causing hassle for the user.

5. Token-based authentication

Token-based authentication technologies (which means in order that a individuals have user receive a unique encrypted string of random characters as token). You can think this can be refers to some protected system instead of requiring every individuals all your inputs. The digital token proves that you already have access permission. Any type of token-based authentication (e.g. RFID, USB tokens) are used by multiple biometrics authentication.

Biometric authentication typically involves the following steps:

Enrollment: The user types a variety of predefined phrases or sentences into a computer or device. These keystrokes are recorded and used to create a profile of the user's typing patterns.

Authentication: The user types the same predefined phrases or sentences, and the system compares these keystroke patterns to the previously recorded profile to verify their identity.

Keystroke authentication can provide a convenient, simple, and low-cost way to authenticate users without requiring any additional hardware or software. However, it has some limitations, such as the potential for false positives or false negatives due to changes in the user's typing patterns over time or variations in the typing environment (e.g. typing speed, typing angle, or keyboard layout).

Despite these limitations, biometric authentication can be a useful additional layer of security when combined with other authentication methods, such as passwords or biometric authentication, to provide a more robust authentication system.

4.1.3 Model Building

The science and practice of machine learning may allow computerized computers to learn from the data provided to them. Computers are frequently trained on the data training set provided to them throughout the machine learning process and they can demonstrate their performance on a specific assignment. Here, in this approach, the problem is resolved with the least amount of human involvement. When traditional methods are ineffective, machine learning is widely used.

Following is a list of possible uses:

- It has the ability to solve complex problems which traditional approaches are unable to do.
- It is capable of identifying extensive amounts of complex data.
- It is capable of resolving complex issues for which standard approaches are ineffective.
- Machine learning techniques can offer better answers when the nature of the environment will be constantly changed or renewed (new solutions, respectively).
- Different surroundings can support it. This enables a frequently used to apply machine learning techniques to a new circumstance.

Supervised Learning

The training data had been appropriately identified and labelled using the user. It's common for dataset to contain information (tags/labels) regarding the type of each item (such as normal or harmful). These tags are compared to the findings discovered by the algorithm in the following steps. Its overall performance, and the algorithm's success is also determined. The strategy has a good track record of success. Although expensive, supervised learning when we provide services (24) manual (tagging) for labeling, and the process is repeated until the algorithm performs at a high degree of accuracy.

Support Vector Machines (SVM) Algorithm

Support Vector Machines (SVM) can be used for biometric recognition, which is the process of identifying individuals based on their unique patterns. SVM is a popular machine learning algorithm that can be used for classification tasks, including biometric recognition.

The first step in using SVM for biometric recognition is to collect data from individuals. This data should include the unique patterns of individuals, such as the structure of key presses, the time between key presses, and the duration of time spent during typing.

Once the data is collected, it needs to be preprocessed to extract relevant features. Feature extraction is an important step in machine learning because it helps to reduce the dimensionality of the data and identify the most relevant features for the classification task. Key keystroke dynamics features can include the duration of key presses, the time between key presses, and the number of errors made during typing.

Once the features are extracted, the SVM model can be trained using a labeled dataset. The labeled dataset should include examples of keystroke patterns from different individuals and their corresponding labels. The labels indicate which individual the keystroke pattern belongs to.

After the SVM model is trained, it can be used to classify new keystroke patterns. When a new keystroke pattern is presented to the model, it will use the features extracted from the pattern to predict which individual it belongs to.

Overall, SVM is a powerful machine learning algorithm that can be used for keystroke dynamics. It is important to collect a large and diverse dataset for training the model and to carefully select the relevant features for the classification task.

Advantages of SVM:

- Effective in cases with high dimensions.
- Due to the decision function's use of support vectors, a subset of training points, it maintains memory efficiency.
- For the decision functions, various kernel functions can be employed, as well as custom kernels.

Limitations of SVM:

- Inconsistent in Large datasets.
- Large training time.
- Noisy datasets may complicate.
- Poor performance on high noise.
- Does not directly provide probability.

Manhattan Distance Filter

Manhattan distance is commonly referred to as the taxicab distance. We can find the absolute value of the difference between each dimension rather than the squared difference between them. Manhattan distance is not scale invariant and is comparable to how you may walk through a city block.

The Manhattan distance filter, also known as the city block distance filter or L1 norm filter, is a technique used in image processing and computer vision to filter out noise and preserve edges in an image.

the filter is useful by measuring the Manhattan distance between each pixel and its surrounding neighbors. The Manhattan distance between two points is the sum of the absolute differences between their x and y coordinates. For example, the Manhattan distance between the points (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

To apply the filter to an image, a sliding window is moved over each pixel, and the Manhattan distance between the pixel and its neighbors within the window is computed. If the distance is below a certain threshold, the pixel is considered part of an edge and is preserved. If the distance is above the threshold, the pixel is considered to be noise and is filtered out.

The Manhattan distance filter is particularly effective at preserving edges in images because it does not blur or smooth them out, as some other filters can do. Instead, it only removes pixels that are not part of an edge, while leaving the edges intact. This can result in sharper and more detailed images.

Overall, the Manhattan distance filter is a useful tool for improving the quality of images in computer vision and image processing applications, particularly those that require edge preservation.

Advantages of Manhattan distance filter

1. Simplicity: The Manhattan distance filter is straightforward to implement and understand. It calculates the distance between two points by summing the absolute differences between their coordinates. This simplicity makes it easy to incorporate into algorithms and systems.
2. Computational efficiency: The Manhattan distance filter is computationally efficient to calculate compared to other distance metrics, such as the Euclidean distance. It avoids the need for computing square root operations, which can be time-consuming, especially in large-scale applications.
3. Feature selection: The Manhattan distance filter gives equal importance to all features or dimensions of the data. It is useful in scenarios where each feature contributes independently to the similarity or dissimilarity between two points. This can be advantageous when dealing with high-dimensional data, as it prevents any single feature from dominating the distance calculation.
4. Robustness to outliers: The Manhattan distance filter is less sensitive to outliers compared to the Euclidean distance filter. Outliers have a larger impact on the Euclidean distance due to the squared term in the formula, whereas the Manhattan distance considers only the absolute differences. As a result, the Manhattan distance filter can provide more robust results when dealing with noisy or outlier-prone data.

4. Spatial proximity: The Manhattan distance filter is particularly useful in scenarios where spatial proximity matters. It accurately captures the "city block" or "taxicab" distance between two points in a grid-like or city-like environment. This property makes it suitable for applications such as route planning, urban planning, and image processing tasks involving grid-like structures.
5. Interpretability: The Manhattan distance filter produces interpretable results because the distance value corresponds directly to the sum of absolute differences between coordinates. This interpretability makes it easier to understand and explain the reasoning behind the filtering or decision-making process.

Limitations Of Manhattan Distance Filter

1. Directional insensitivity
2. Sensitivity to scale
3. Limited representation of curvilinear relationships
4. Inability to capture non-linear relationships
5. Sensitivity to feature dependence

4.6 Performance Metrics:

Performance metrics, also known as evaluation metrics or assessment measures, are quantitative measures used to assess the performance or effectiveness of a specific model, algorithm, or process. These metrics provide objective criteria for evaluating the quality, accuracy, efficiency, or other desirable attributes of a particular solution or approach. Performance metrics are commonly used in various fields, including machine learning, data analysis, optimization, and system evaluation.

Performance metrics can vary depending on the specific problem and context. Here are some commonly used performance metrics in different domains:

1. Classification Metrics:

a. Accuracy: The proportion of correctly classified instances.

b. Precision: The proportion of true positive predictions among all positive predictions.

c. Recall (Sensitivity or True Positive Rate): The proportion of true positive predictions among actual positive instances.

d. F1 Score: The harmonic mean of precision and recall, providing a balanced measure.

e. Specificity (True Negative Rate): The proportion of true negative predictions among actual negative instances.

f. ROC Curve (Receiver Operating Characteristic Curve): A graphical representation of the trade-off between true positive rate and false positive rate for different classification thresholds.

g. Area Under the ROC Curve (AUC-ROC): A metric that summarizes the ROC curve, providing a measure of the overall performance of a classifier.

2. Regression Metrics:

a. Mean Absolute Error (MAE): The average absolute difference between predicted and actual values.

b. Mean Squared Error (MSE): The average squared difference between predicted and actual values.

c. Root Mean Squared Error (RMSE): The square root of MSE, providing a more interpretable measure in the same unit as the target variable.

d. R-squared (coefficient of determination): The proportion of the variance in the target variable explained by the model.

e. Mean Absolute Percentage Error (MAPE): The average percentage difference between predicted and actual values, commonly used for relative error assessment.

3. Clustering Metrics:

a. Silhouette Coefficient: Measures the compactness of clusters and separation between clusters.

b. Calinski-Harabasz Index: Measures the ratio of between-cluster dispersion to within-cluster dispersion.

c. Davies-Bouldin Index: Measures the average similarity between each cluster and its most similar cluster.

4. Information Retrieval Metrics:

a. Precision at K: The proportion of relevant items among the top K retrieved items.

b. Recall at K: The proportion of relevant items retrieved among all relevant items.

4.4.1 Equal Error Rate

Equal Error Rate (EER) is a performance metric commonly used in biometric systems and binary classification tasks. It provides a balance between false acceptance rate (FAR) and false rejection rate (FRR) by finding the point at which these rates are equal.

In biometric systems, such as facial recognition or fingerprint identification, the goal is to accurately identify individuals or other patterns (positive) or reject (negative). The FAR represents the point at which the system makes an equal number of false acceptances and false rejections. In other

exactly, the EER is the threshold or operating point where the rates of false acceptance and false rejection coincide.

To calculate the EER, a receiver operating characteristic (ROC) curve is constructed by varying the decision threshold of the classifier, which determines the classification boundary between positive and negative classes. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings.

The EER is obtained by finding the threshold at which the FPR and FRR are closest to each other. This ensures that the EER represents the point where the probability of incorrectly accepting a negative sample is approximately equal to the probability of incorrectly rejecting a positive sample.

The EER is often used as a single summary metric to compare different biometric systems or classification algorithms. It provides a concise measure of performance, but it is important to note that it may not capture the entire performance profile of the system. Additional metrics, such as the area under the ROC curve (AUC-ROC), or graphical visualizations can provide more detailed insights into the performance characteristics of the system.

In summary, the Equal Error Rate (EER) is a threshold-based performance metric that represents the point at which false acceptance rate (FAR) and false rejection rate (FRR) are equal in biometric systems and binary classification tasks.

The Equal Error Rate (EER) is calculated by finding the threshold at which the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are equal.

$$
\begin{aligned}
&\text{FAR} = \text{FP} / (\text{FP} + \text{TN}) \\
&\text{FRR} = \text{FN} / (\text{FN} + \text{TP}) \\
&\text{EER} = \text{FAR} = \text{FRR}
\end{aligned}
$$

where:

- FP: False Positives (incorrectly classified negative samples as positive)
- TN: True Negatives (correctly classified negative samples as negative)
- FN: False Negatives (incorrectly classified positive samples as negative)
- TP: True Positives (correctly classified positive samples as positive)

To calculate the EER, you need to vary the decision threshold of the classifier and calculate the FAR and FRR at each threshold. The EER is then obtained by finding the threshold where the FAR and FRR are approximately equal.

CHAPTER-V

5.RESULTS AND DISCUSSION

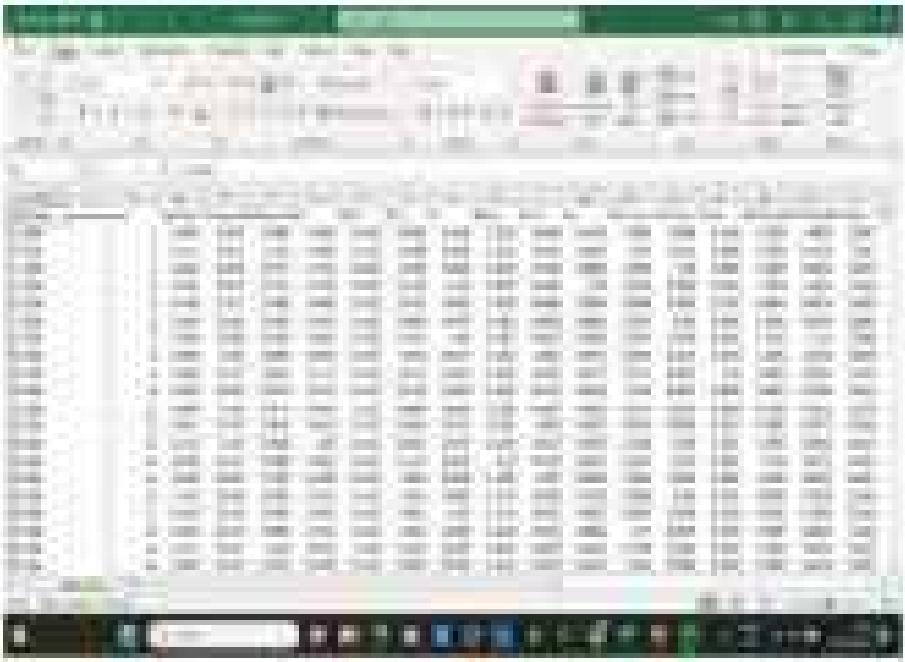
[redacted]

YEAR : 1

5.1 Data Collection

[redacted]

[redacted]



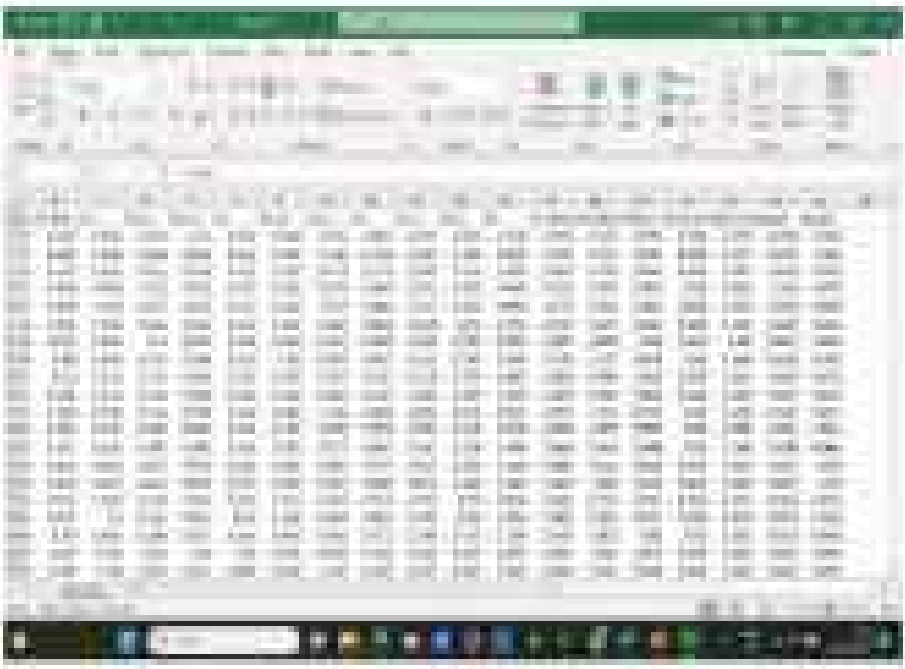


Figure 4.1 Data Collection

4.1.1 Importing Libraries

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

[illegible]

5.1 Data Preprocessing

In this project for data pre-processing, I used to check the information of the dataset such as if values and duplicate attribute. Figure 5.1 and 5.2 shows the results of data pre-processing.

Data preprocessing

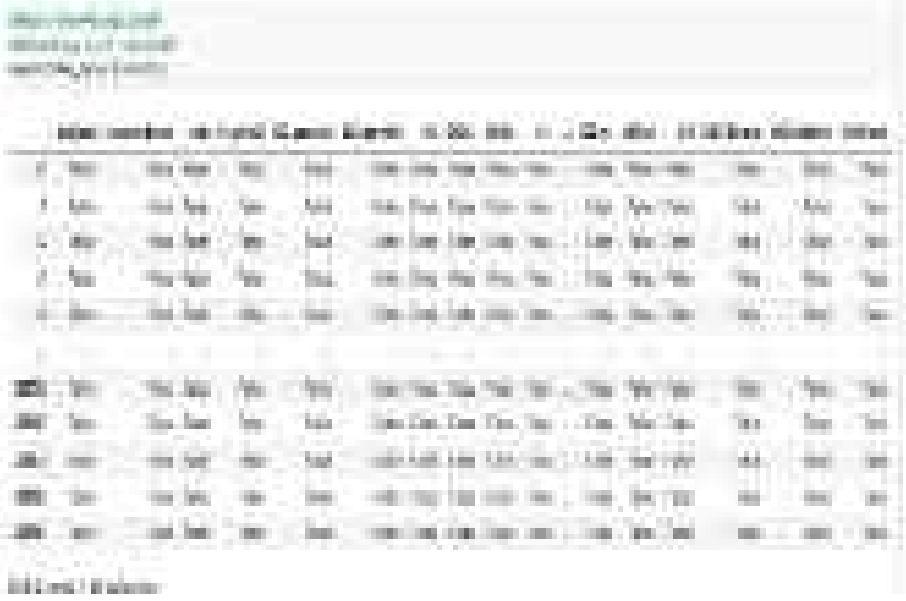


Figure 3.2 After checking null values

Figure 3.3 After checking duplicate records

From figure 3.2 and figure 3.3 it is observed that this dataset does not contain any null values and duplicate records.

PHASE III

Feature Extraction

[illegible]

Feature extraction

[illegible]

Figure 5.6 After Feature extracting

[illegible]

Figure 5.7 Added history parameters

Figure 4.2 Measures Of binary parameters

From Figure 4.1 and Figure 4.2 it is observed that the feature should time, Night time, daynight and trigraph has added as the 2 MAHENDRA dataset

CHAPTER IV

Feature Selection

Feature Selection has done with the intension of lessening and eliminating irrelevant, less significant and redundant using KNN and SVM algorithms, in addition,also the interactive improvement

Shell for Modelling

```
[illegible]
```

```
[illegible]
```

Figure 5.8 Coordinates for STEM

Figure 2.7 Training and Testing in SVM

ManhattanFilter

```
from scipy.spatial.distance import euclidean, cityblock
import numpy as np
cg.set_printoptions(suppress = True)
import random
from PIL import ImageEnhance
import seaborn as sns
```

```
class RandomWalk():
def test_for_training(): function changes, error if() program over

    def __init__(self, id=0):
        self.user_rules = []
        self.imputer_state = []
        self.mean_center = []
        self.subject_lower = subject_brown

    def training(self):
        self.mean_center = self.knob.mean().values
        self.df_center = self.knob.std() values
        dropping_list = []
        for i in range(self.knob.shape[1]):
            dec_score = ImageEnhance.Color(self.train.iloc[:]).values,
                                           self.mean_center)
            if (dec_score > float(self.std_center)).all() == True:
                dropping_list.append(i)
        self.rain = self.knob.drop(self.knob.index[dropping_list])
        self.mean_center = self.knob.mean().values

    def testing(self):
        for i in range(self.knob.shape[1]):
            dec_score = ttest_ind(self.test_preview.iloc[:], values,
                                  self.mean_center)
            self.user_scores.append(dec_score)
```

Figure 13 Correlation for Business Account Time

The train and test method is applied to divide the dataset for training phase and testing phase.

Figure 5.1 Training and Testing in Machine Learning Tutorial Point

From Figure 5.1 to Figure 5.9 it is observed that it shows clustering and recommend training of the training and testing, and has done on the [illegible] dataset from data8 base to graph.

Selecting Random User for SVM

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

```
[illegible]
```

Figure 7.10 Selecting Random User for SVM

Selecting Random User for Mushroom Poisonous Filter

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

```
[illegible]
```

Figure 7.11 Selecting Random User for Mushroom Poisonous Filter

In Figure 7.10 and Figure 7.11 selecting random user for SVM and Mushroom Poisonous filter

RESULT

Core Architecture

Signal-level performance analysis validates the proposed architecture at the RTL and GDSII stages.

```
[illegible]
```

```
[illegible]
```

Figure 5.12 Performance analysis of SERV for STA1

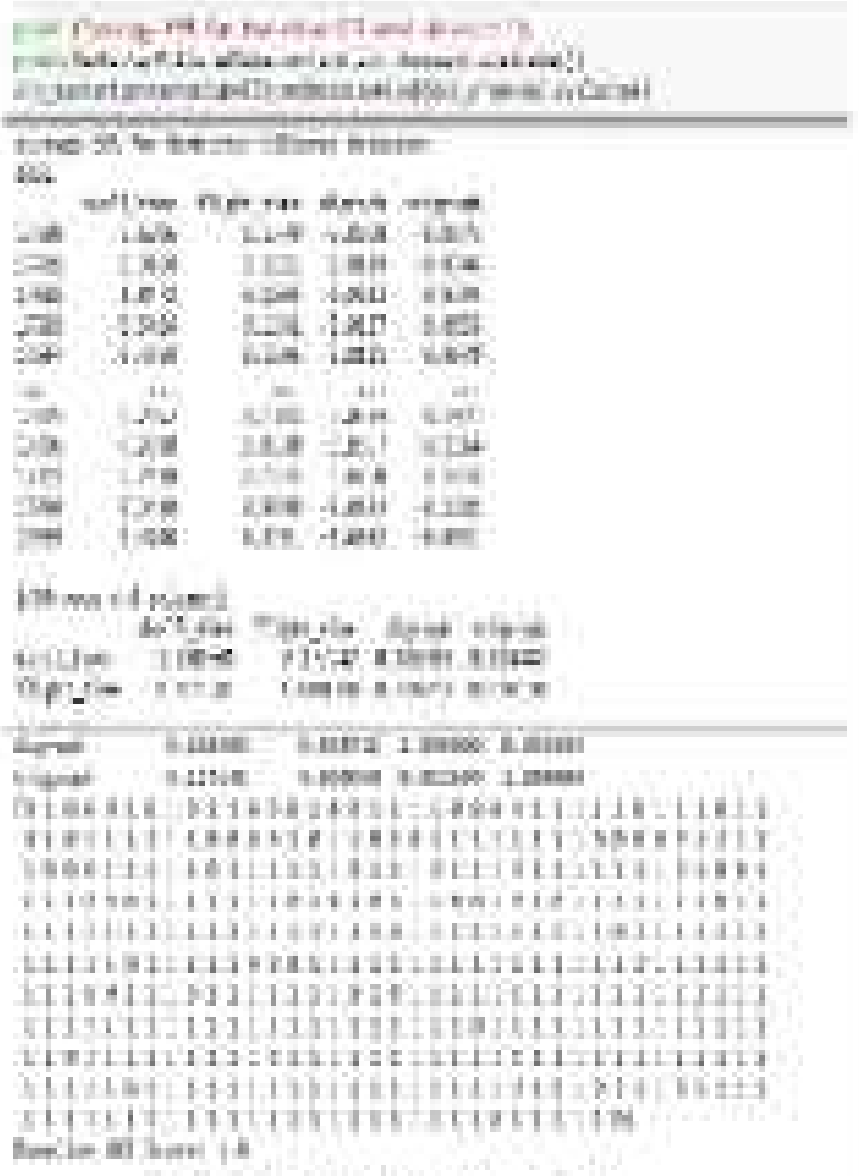
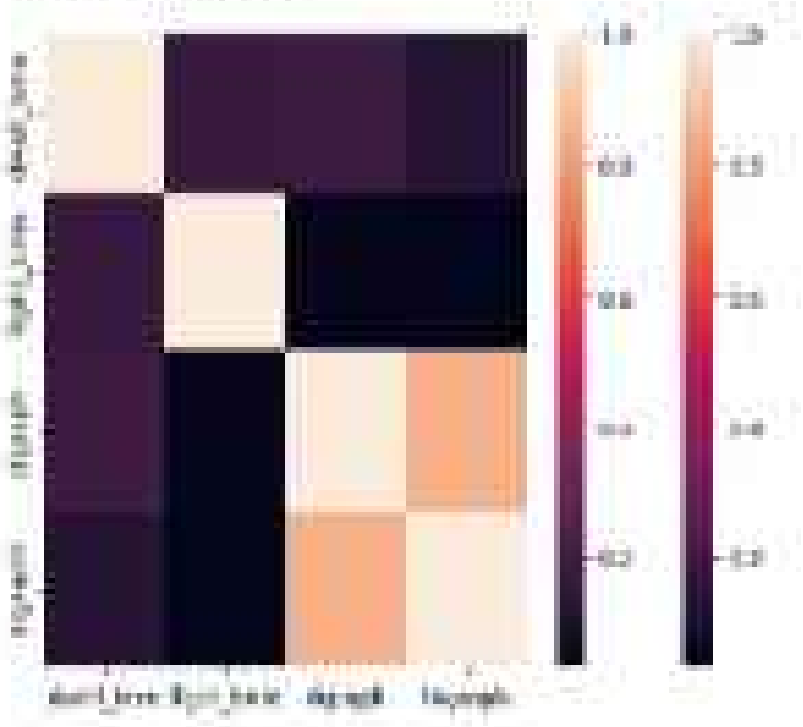


Figure 5.11 Performance analysis of ERM by SDF

[illegible] Performance Evaluation of the Model



[illegible]

[illegible]

Figure 1 [illegible] Correlation graph for ANN model with [illegible] Score

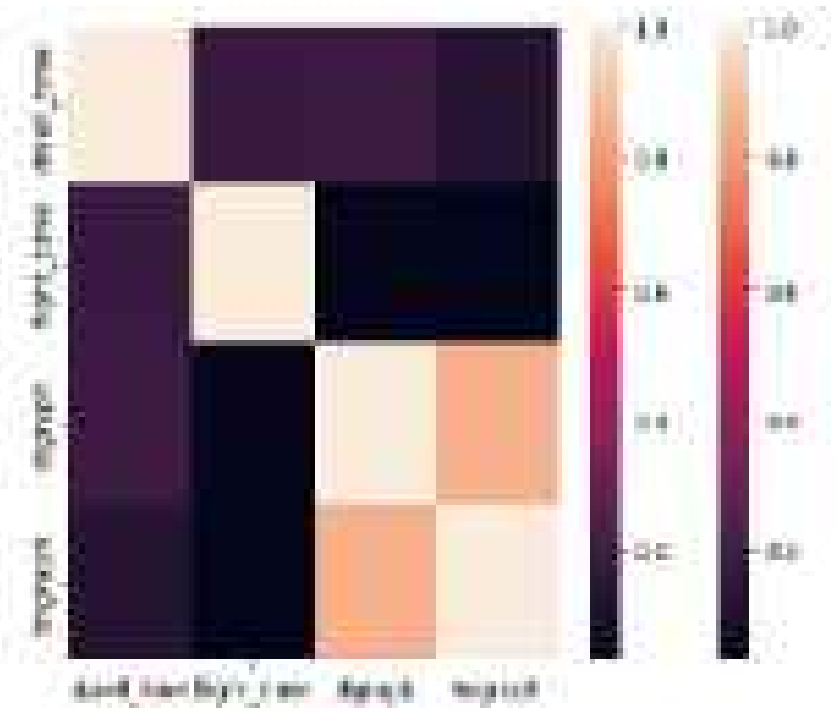


Figure 1.14 Correlation graph for SDF model with IEB. Inset

Algorithm Comparison

Algorithm comparison

```
[illegible]
[illegible]
    [illegible]
    [illegible]

[illegible]
    [illegible]
    [illegible]

[illegible]
    [illegible]

[illegible]
    [illegible]
```

Figure 1.11: Comparing HMM and MEMM models

[illegible]

[illegible]	[illegible]
[illegible]	[illegible]

Figure 1.12: [illegible]

Figure 1.11 shows that the result of this experiment was by comparing the Support Vector Machine and Maximum Entropy Markov Model and it shows that MEMM model gives us the lowest WER score.

Accuracy

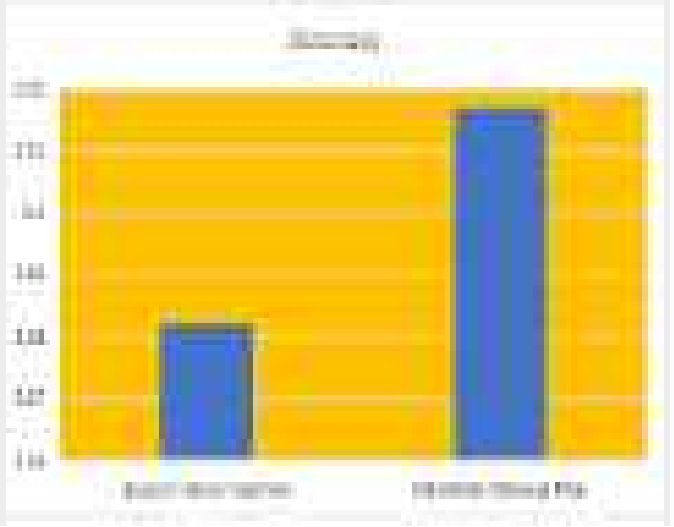


Figure 5.27 Accuracy Value

6. CONCLUSION AND FUTURE SCOPE

misclassify. It can not possible to compare the performance of different anomaly detection systems available in the keystroke dynamics literature. Our objective in this work is develop an evaluation procedure and measure the performance of many algorithms on a typical basis.

In the process, we established which detectors have the lowest error rates on our data. Any FRR, and we provide a data set and evaluation methodology that can be used by the community to assess new detectors and reports comparative results.

In conclusion, the feature scope of keystroke authentication using MLM and Continuous detection filter is vast and there are several exciting directions for research and development. With the advancement of technology and the increasing demand for secure authentication systems, the field of dynamic authentication is expected to continue to grow and evolve in the coming years. Also the meta-layer of security design Capabilities can be included and authenticate the user in the frame of Multi-factor Authentication.

These create the base of a potentially effective way of enhancing overall security using by playing a significant role as part of a larger multifactor authentication mechanism. Low implementation cost, high user acceptance, and ease of integration to existing security systems.

6.1 Overview

In this project ML and MLM Machine Learning models are developed to identify the genuine registered user with user authentication based Keystroke Dynamic. By comparing the ML and MLM models on the basis of ACER. From figure 6.1.1 and figure 6.1.2 it is evident that the ACER value, is lesser for ML model with 0.34 compared with MLM model with equal MLM 0.2 0. Based on the analysis it is suggested that ML model identifies the genuine user more precisely the user authentication problem.

REFERENCES

[1] Ernest, Nasang, and Rahkar Karmo. "Iris signature and identification based on Lyapunov dynamics and monte carlo sampling." IEEE Vol. 10 2022.

[2] Mookhoey, Nirmal A., and Lakhdar K. Zbakocte. "Detecting zombie attacks on IoT based CAPTCHA using the Lyapunov dynamics approach." IET Information Security 14.2 (2019): 179-204.

[3] Alsaedi, Zainebhal A., Abdulfatah M. Abugabah, and Sultanah A. Alzahrani. "An comparison of using variable lyapunov dynamics to enhance the prevalence of phishing attacks." Indian Journal 14.3 (2022): 62.

[4] Mughal, Owais. Contributions to Lyapunov dynamics for privacy and security in the Internet-of-Things. Northumbria University, 2019.

[5] Ali, Md Liakat, et al. "Keystroke biometric system for user authentication." Journal of Signal Processing Systems 86.2 (2017): 175-190.

[6] Yow, Kin Choong, et al. "Incognito: Exploring Lyapunov exponents in real-world scenarios." IEEE Access 11 (2023): 1811-1840.

[7] Kashmar, Rima, Manoj Jayabalan, and Fawzia Thong Jachmann. "Lyapunov dynamics based anti-collision sleep scheduling in perceptron." International Journal of Machine Learning and Computing 10.1 (2020): 134-139.

[8] Kulkarni, Kunal S., and Ajoy S. Ravanan. "Computing anomaly detection algorithms for lyapunov dynamics." 2020 IEEE/ACIF International Conference on Dependable Systems & Networks. 2014. 2040.

[9] Kale, Sushmapadma, et al. "Lyapunov Detection of anomalies and user based on lyapunov dynamics." arXiv preprint arXiv:2009.00018 (2021).

[10] Singh, and Sanjeev. "Automatic Hand Lyapunov Authentication System Using Support Vector Machine", M.Sc. Vol. 19, 2021.

Document Links

[redacted]

Website References

[redacted]

8. APPENDIX

8.1 Sample Coding

Importing Libraries

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
model_df = pd.read_csv("C:\Users\Admin\Desktop\Algorithms\CSE Data")
```

Loading the Dataset

```
df = pd.read_csv("C:\Users\Admin\Desktop\Algorithms\CSE Data")
```

Data Preprocessing

```
print("DATA TYPE PREPROCESSING:")
df.fillna(0, inplace=True)
print(df.info(), df.describe())
X = df.drop(columns=['Algorithm'])
y = df['Algorithm']
print(X.shape, y.shape)
```

Feature Extraction

```
df['Algorithm'] = df['Algorithm'].astype('category')
df
print(df)
X = df.drop(columns=['Algorithm'], axis=1)
y = df['Algorithm']
print("Features (X):\n", X.head())
print("Target (y):\n", y.head())
```

TTS for Modeling

```
from sklearn.externals import joblib
import numpy as np
np.set_printoptions(suppress=True)
import pandas
import random
import sys, os, re, csv
from tflib import model_tf
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics.cluster import normalized_mutual_info_score, adjusted_rand_score

class TTSModel:
    '''A TTS model based on deep neural networks.

    def __init__(self, obj):
        self._words = []
        self._ners = []
        self._tokens_count = []
        self.obj = obj

    def _prepare(self):
        self.df = DataLoader().load_pos_data(obj)
        self.df.fillna('', inplace=True)

    def _extract_words_and_tags(self):
        self._words = self.df['Word'].values.tolist()
        self._ners = self.df['Tag'].values.tolist()
```

```
def evaluate(self):
    self.__()
    print(self.score)
    guesses_per_date = {cycle.day_data[self.score.date_data.origin] : self.score.
                        'result_count':'tag.graph'}
    print(guesses_per_date)
    # guess data that will be plotted
    # collection viewer/epoch per selected by cycle() return
    # they are stored for the plot
    original_submit as str
    print(guesses_pred_data.count())

    # printing a validation heading
    data[self.id].save(tag.guesses_per_date.count())

    # displaying list
    # We are going to use the A_Classic algorithm
    model_instance = AggloClassical.fit(n=height_clusters=2)
    plot = model_instance.fit_predict(guesses_per_date)
    print(plot.sum)

    # We are going to use the 1991 forms to measure the quality performance of this clustering
    heading_score = normalized_mutual_info_score(model_instance.data_, plot)
    print("Baseline MIS Score:", heading_score)

    print("Reduced features:", model_instance.feature_names_in_)

    cluster_data = form.data_frame[form.data_frame['cluster'] == cluster_]

    # Feature Reduction
    self.train = preview_sum_data[:200]
    self.test_preview = [preview_sum_data[200:]
    self.final_response = tmp.read_data.groupby('subject')
```

```
```

Unsharp Mask Filter for Blurring

```
```

```
    def object_decoder(obj):
        if '__type__' in obj and obj['__type__'] == 'complex':
            return complex(obj['real'], obj['imag'])
        return obj

    data = '{"__type__": "complex", "real": 1, "imag": 2}'
    x = json.loads(data, object_hook=object_decoder)
    print(type(x), x)
    # <class 'complex'> (1+2j)
```

```
// release_device_lock was called by user() once
// the() we need to  the pin
import subprocess as sp
processor_use_disconnect()

# photo_overflow_timestamp
datapoint = b.load_graph(power_occ_disconnect)

# explaining fix
# the old prog is too fly & throws algorithm
model_browser = f'gpt-{random.choice(range(1, 2))}'
price = model_browser.fit_position(private_river_state)
print(price)

# this sad prog is too fix 503 metric in research this quality-performance of fix cleaning
headline_cache = normalize_bread_mb_socket(model_browser.state_price)
print('Baseline fell home', headline_cache)

print("Editorial Eastern : ", model_process.nation_name_is_)
response_data = data.iloc[data.shape[1] - when_diverse, ]
soft_train = generate_occ_data[:50]
soft_bot_private = generate_bot_data[2:]
soft_occ_response = response_data.groupby('adjust')
                    .load(1.loc[], 'break_river': mapped})
soft.training()
soft.backup()
sun.upgrade_visual_kill('pull test', sum,
                        soft.income_current)
return q_customers;
```

```
path = "C:\Project Work\project dataset\Book_1001.csv"
data = pandas.read_csv(path)
numbers = data["category"].unique()
category_classes
print("Loading 1000 Sci Mathematical Physics Books...")
print(df.shape,end="\n\n")#prints the shape of the dataset
cm_matrix= Mathematical Work(df['title'],df['category'],shape=0.4,state=0)
cm_matrix
```

Algorithm Comparison

```
accuracy = {}
cf mat_cm = cm_matrix
best_cm= mx_cm
cfg = "Sigmoid Kernel Machine"

df1 mx_cm = cm_matrix
best_cm= cm_matrix
cfg = "Mathematics Work"

if best_cm == Machine:
    acc= "Learning Level"

else:
    acc= "Improving"
```

Database

```
Model= pickle.dump(cm_matrix,fig,best_acc)
# Verifying the score and its respective model
save_mx = "{} has {} with {}".format("Algorithm",cfg," Limit Rate",
best_acc)

# Inserting the method in a database name as the Mathematics
result_df.to[mx]=result_df[] + save_mx
result_df
```