A PRELIMINARY REPORT ON

TWO FACTOR AUTHENTICATION USING BEHAVIORAL ANALYTICS

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
(COMPUTER ENGINEERING)

SUBMITTED BY

HIMANSHU YADAV	Seat No: B150224222
PREM SAKORE	Seat No: B150224237
SAURAV SEN	Seat No: B150224247
SHAURYA KHURANA	Seat No: B150224249

DEPARTMENT OF COMPUTER ENGINEERING

ARMY INSTITUTE OF TECHNOLOGY

ALANDI ROAD DIGHI, PUNE 411015

SAVITRIBAI PHULE PUNE UNIVERSITY
2018 -2019

CERTIFICATE

This is to certify that the project report entitles

“Two Factor Authentication using Behavioral Analytics”

Submitted by

HIMANSHU YADAV	Seat No: B150224222
PREM SAKORE	Seat No: B150224237
SAURAV SEN	Seat No: B150224247
SHAURYA KHURANA	Seat No: B150224249

is a bonafide work carried out by students under the supervision of **Prof. Sushama A. Shirke** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Sushama A. Shirke)
Guide,
Department of Computer Engineering

(Prof.(Dr.) S.R.Dhore)
Head of Department,
Department of Computer Engineering

(Dr. B.P. Patil)
Principal,
Army Institute of Technology, Dighi, Pune – 411015

Place: Pune

Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on **“TWO FACTOR AUTHENTICATION USING BEHAVIORAL ANALYTICS”.** We would like to take this opportunity to thank our internal guide **Prof. Sushama A. Shirke** for giving us all the help and guidance we needed. We are grateful to her for her kind support. Her valuable suggestions were very helpful. We are also grateful to **Prof. Dr. Sunil R. Dhore**, Head of Computer Engineering Department, AIT, Pune for his indispensable support, suggestions. We would like to thank the project coordinator Prof. Anup Kadam for his valuable suggestions and support throughout the last three months of the semester.

Himanshu Yadav
Prem Sakore
Saurav Sen
Shaurya Khurana

ABSTRACT

Every organization loves to have new security features installed in their workstations. Usually in every organizations for each and every employee they have their personal workstations assigned to them, this brings two problems one is if a workstation is not in use and a new person wants to use it he or she can’t because he is not authorized to use that.

Another problem lies in security. It is often seen that employees left their workstation open, this may lead to security breach as any other person can use that workstation in absence of him and do some mal practices which is harmful to that employee. For this we aim towards developing a software product that will use the behavior of the user and will authenticate the user. We propose a framework that detects and recognizes the genuine user of that workstation using machine learning . Our approach intuitively identifies relevant features associated with behavior of user such as the speed with which is types.

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
UBA	User Behavior Analytics
SIEM	Security Information and Event Management

LIST OF FIGURES

LIST OF TABLES

INDEX

Table of Contents

1 INTRODUCTION

User Behavior Analytics (UBA) uses big data and machine learning algorithms to assess the risk, in near-real time, of system user activity within your organization. Why is this analysis necessary? Think about it: everyday, your employees are using user credentials to access the organization's systems from the company office during regular business hours. One day you are notified that an individual's credentials were used to connect to a database server and run queries that this user has never performed before. Is a database administrator running maintenance checks or has the system been compromised? User behavior analytics can help an organization determine what normal behavior should look like within their systems and when to be cautious of unusual activity.

According to the recent SANS Analytics and Intelligence Survey, only about one-third of organizations today collect user behavior monitoring data, but approximately three-fourths of respondents say they intend to start collecting this data in the future. Understandably so—user behavior analytics offer visibility into potential insider threats, show early red flags for when accounts have been compromised by external attackers and are most useful to measure changes in user behavior. Ultimately, the foundation of a behavior analytics program is to understand what normal behavior looks like to catch irregularity in the system. Below are 3 key areas to focus on when establishing behavior analytics and measuring user behaviors.

Determining human and machine behaviour

Normal behaviour for accounts used by humans will look different than that of service accounts that are used to carry out automated application activity. These machine accounts usually have a large amount of permissions; however, their activity is much more predictable than human user accounts. In addition, the volume activity of automated accounts is usually much higher than human accounts.

When tracking user behaviour, it is important to which type of account is being looked at when determining what unusual behaviour is.

Track mobile device location data

Mobile devices provide a great opportunity for tapping into the power of user behavior analytics. Forward-looking security programs are able to use the location tracker on smartphones as a data point in user behavior analytics. Through tracking mobile devices, security teams are able to flag any situation where an authentication is coming from a different physical location than the location of the smartphone.

Keep tabs on machine admin accounts

Companies must keep track of local machine administrator accounts in addition to active directory accounts. Cyber criminals tend to leverage these local accounts to move work their way into a system until they can break into a more critical user account. These hackers are usually successful within companies that use a standard image for rapid desktop deployment and keep local domain administrator passwords identical to simplify helpdesk requests.

User behaviour analytics are helping to transform security and fraud management by enabling organizations to detect when legitimate user accounts have been compromised by external attackers or are being abused by insiders for malicious purposes.

Traditional Model vs New Approach

Security Information and Event Management or SIEM, is the traditional model that uses complex set of tools and technologies that gives a comprehensive view of the security of your IT system. It makes use of data and event information, allowing you to see patterns and trends that are normal, and alert you when there are anomalous trends and events. UEBA works the same way, only that it uses user (and entity) behavior information to come up with what's normal and what's not.

SIEM, however, is rules-based, and advanced hackers can easily work around or evade these rules. What's more, SIEM rules are designed to immediately detect threats happening in real time, while advanced attacks are usually carried out over a span of months or years. UEBA, on the other hand, does not rely on rules. Instead, it uses risk scoring techniques and advanced algorithms, allowing it to detect anomalies over time.

One of the best practices for IT security is to use both SIEM and UEBA to have better security and detection capabilities.

1.1 MOTIVATION

- Develop an algorithm to avoid spoofing
- Secure access
- Access away from home and physical tokens or on a plane with no network coverage for SMS but an internet connection
- Build the model with low development and maintenance cost

1.2 PROBLEM DEFINITION

To build a fault tolerant, attack resistant software system using behavior analytics that will use behavior of user while using his or her workstation such as typing pattern , speed and use that to authenticate the user.

Goals and Objectives

- To perform a detailed search in the field User Behavior Analytics
- To propose a new model which overcomes the secure access.
- Comparative study of the proposed and existing systems.
- To implement the proposed model
- Making the platform user-friendly and more flexible

2 LITERATURE SURVEY

1) Shepherd, S. J. "Continuous authentication by analysis of keyboard typing characteristics.": 111-114

This paper describes a simple, software based keyboard monitoring system for the IBM PC for the continuous analysis of the typing characteristics of the user for the purpose of continuous authentication. By exploiting the electrical characteristics of the PG keyboard interface together with modifications to the internal system timer, very accurate measurements can be made of keystroke interval and duration, including measurements OF rollover. Rollover patterns, particularly when typing common diphthongs, can be highly characteristic of individual users and provide quite an accurate indication of the users identity. There are a number of different aspects of keystroke characteristics that can be used as identification criteria :-

- Intervals between keystrokes:-These can be analyzed on the basis of a simple mean time interval across all keystrokes or between particular pairs of keystrokes of significance such as common pairs of characters (digraphs).
- Duration of keystrokes.
- Frequency of errors, This could be monitored by detecting the specific use of the delete and backspace keys.
- Force of keystrokes. While this might give valuable additional information, no computer keyboard offers the ability to measure this quantity.
- Rate of typing. The average number of words or characters per minute.
- Statistics of text. The individual language use or style of a user might be analyzed but this would require significant natural language processing and would only be applicable in those situations where a reasonably large amount of text processing was being carried out. .

The keyboard hardware interrupt occurs once for each key depression and for each key release. The hardware scan codes associated with each key are transmitted as well so that each physical key can be identified. It is important to note that much of this information is lost after processing by the BIOS. For example, capital letters can be obtained by pressing either ship key - there is no apparent difference between them. Likewise, most keyboards have two CRTL keys and two ALT keys to suit the convenience of both right and left handed user. The BIOS keyboard processor generates exactly the same output for both shift keys or CRTL .The keyboard interrupt handler is called each time a key is pressed or released. The key associated with each operation is noted and the number of timer ticks between events is recorded.From this data, the: duration of each keystroke, the interval between, keystrokes and the overlap between keystrokes is computed. A running update of the mean and variance: of these quantities is kept and is available for display in the demonstration system.

2) **Panasiuk, Piotr, and Khalid Saeed. "A modified algorithm for user identification by his typing on the keyboard."** ***Image Processing and Communications Challenges***
In this paper the authors modify their previous kNN algorithm and present a modification to improve the algorithm by considering key inner and interclass distinguishability. The suggested approach is tested on a large group of individuals with data gathered over Internet using browser-based WWW application. The obtained results are promising and encouraging for further development in this area. Data have been gathered in non-supervised way with the web-based platform. Samples consist of five phrases that everyone has their unique features. In both language versions are the same dependencies in sample selection. Each phrase in a sample is stored in the database as a series of key events written as a text. At the beginning we read the SQL file and load it into the testing subprogram. Each keyboard event in the database is recalculated from the time a key is pressed or released to the flight time and dwell time. Flight time is the time between releasing a key and pressing the second. Dwell time is the time when a specific key is in pressed state. After loading the database file into the testing platform backspace and delete keys are removed from samples with affected keys information. Later are removed samples with different events count than the most common. After this, the users with less number of samples than the training set size plus at least one test sample per user are removed from the database. The next step is splitting the remained database into test and training sets. Previous steps provided constant count of training samples per user. In the following step every test sample in the remained database is being classified.

3) **Akinsola, J E T. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. International Journal of Computer Trends and Technology (IJCTT).**

Supervised Machine Learning (SML) is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances. Supervised classification is one of the tasks most frequently carried out by the intelligent systems. This paper describes various Supervised Machine Learning (ML) classification techniques, compares various supervised learning algorithms as well as determines the most efficient classification algorithm based on the data set, the number of instances and variables (features).Seven different machine learning algorithms were considered:Decision Table, Random Forest (RF) , Naïve Bayes (NB) , Support Vector Machine (SVM), Neural Networks (Perceptron), JRip and Decision Tree (J48) using Waikato Environment for Knowledge Analysis (WEKA)machine learning tool.To implement the algorithms, Diabetes data set was used for the classification with 786 instances with eight attributes as independent variable and one as dependent variable for the analysis. The results show that SVMwas found to be the algorithm with most precision and accuracy. Naïve Bayes and Random Forest classification algorithms were found to be the next accurate after SVM accordingly. The research shows that time taken to build a model and precision (accuracy) is a factor on one hand; while kappa statistic and Mean Absolute Error (MAE) is another factor on the other hand. Therefore, ML algorithms requires precision, accuracy and minimum error to have supervised predictive machine learning.

4) **Juola, Patrick, et al. "Keyboard-behavior-based authentication."** ***IT Professional***

One of the most important ways of interacting with a computer is through the keyboard (and mouse), and keyboard interaction includes not just behavioral data (such as typing speed) but also cognitive and linguistic data. Researchers have successfully applied the analysis of language usage to infer the authorship of written documents, but these technologies aren't commonly used for authentication. The theory behind stylometric technology is that everyone has their own unique style , a unique set of idiolectal choices that describe their speaking and writing style. The application of stylometric technology to authentication is fairly straightforward. Instead of using a training set of documents, we use a pseudo document containing the user’s long-term behavior, and we verify that the recent behavior at the keyboard is consistent with this long-term behavior. A significant inconsistency, of course, would trigger a security response.

5) **Banerjee, Salil P., and Damon L. Woodard. "Biometric authentication and identification using keystroke dynamics: A survey."** ***Journal of Pattern Recognition Research***

Computers have become an ubiquitous part of the modern society. In early 2011, online attacks on companies resulted in the shutdown of their networks and compromised the passwords and personal information of millions of users. Since we depend so much on computers to store and process sensitive information, it has become all the more necessary to secure them from intruders. For user authentication and identification in computer based applications, there is a need for simple, low-cost and unobtrusive device. A user can be defined as a person who attempts to access information stored on the computer or online using standard input device such as the keyboard. Use of biometrics such as face, fingerprints and signature requires additional tools to acquire the biometric which leads to an increase in costs. Use of a behavioral biometric which makes use of the typing pattern of an individual can be obtained using existing systems such as the standard keyboard, making it an inexpensive and extremely attractive technique. One of the major advantages of this biometric is that it is non-intrusive and can be applied covertly to augment existing cyber-security systems. The features that can be extracted from the raw typing data are the key pressed and the time at which it was pressed. In addition to this, it can also record the time at which the same key was released and for how long it was pressed. Keystroke dynamics has a strong psychological basis which should be explored to gain deeper understanding of the motor behaviour during typing. Using these concepts, models could be built to better understand the processes involved in typing. An understanding of how different people or groups of people type may provide insight into patterns in soft biometric features such as age and gender. This might help in the development of better classifiers which could improve the accuracies of existing systems.

3 SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

3.1.1 Purpose

The project “Two Factor Authentication using Behavioural Analytics” aims towards developing a software product which enhances the security against intruders in personal information by adding a factor of authentication.

3.1.2 Document Conventions

- All the points mentioned in this document in order of priority from Higher to Lower.
- Each diagram showcases the actual implementation of each topic under which it is mentioned.

3.1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- Developers
- Project Managers
- Software Testers
- Industrial project guide

3.1.4 Project Scope

- Accops System Pvt. Ltd. is going to use this project in their product which is a multi-factor authentication solution.
- User will not have to bother passwords and other stuffs .
- The project follows all the rules and regulations provided by Accops System Pvt Ltd.

Software context

This is a machine learning based system that will compete against the traditional models that uses rule based systems. There are many benefits of using the proposed system over existing system. The proposed system will make the system fault tolerant, attack resistant and immutable. The user will able to authenticate himself where the system will intelligently recognize him using his behavior. The system is cost efficient in terms of gas used for each recognition. Issues of the rule based authentication system will get solved .

Major constraints

- Limited time of implementation.
- Trust factor.
- Nature of user behavior.
- Accuracy of the model used to detect.
- Time to switch from traditional system to the proposed system.

3.1.5 User Classes and Characteristics

Following are the user classes expected in the software:

- Customer
- Developers
- User administrator

The following is an advanced class diagram which showcases all the classes and their relationships with each other.

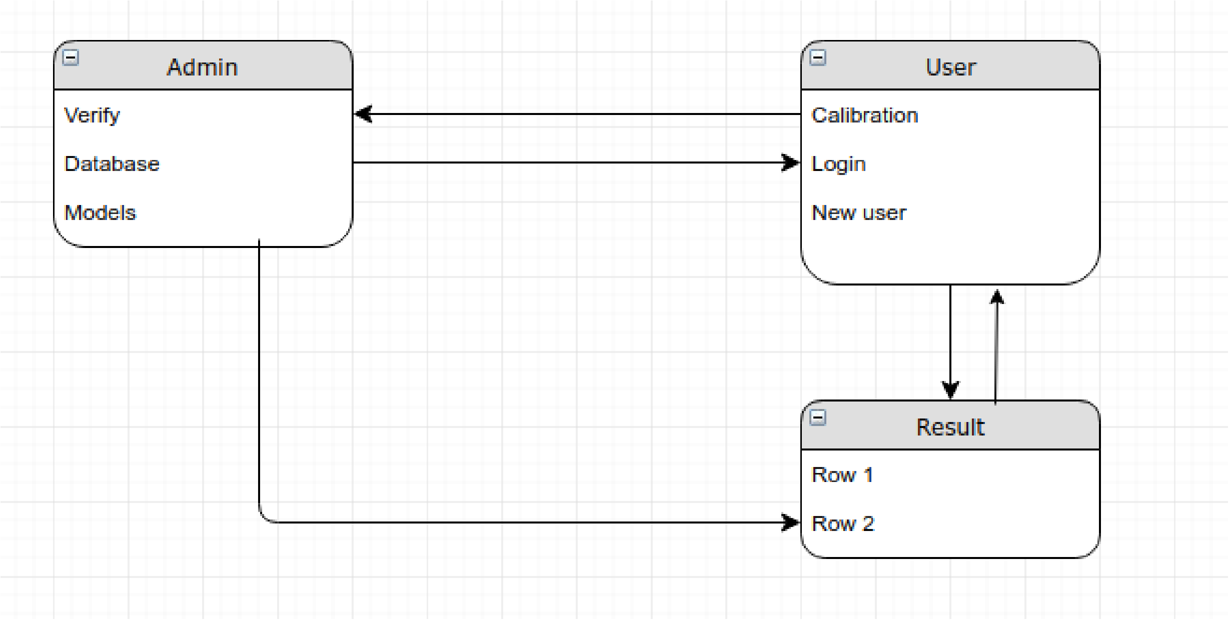


Figure 1. Class Diagram

3.2 Functional Requirenments

3.2.1 System Features

The following diagram describes all the features of our system and the actors involved in it.

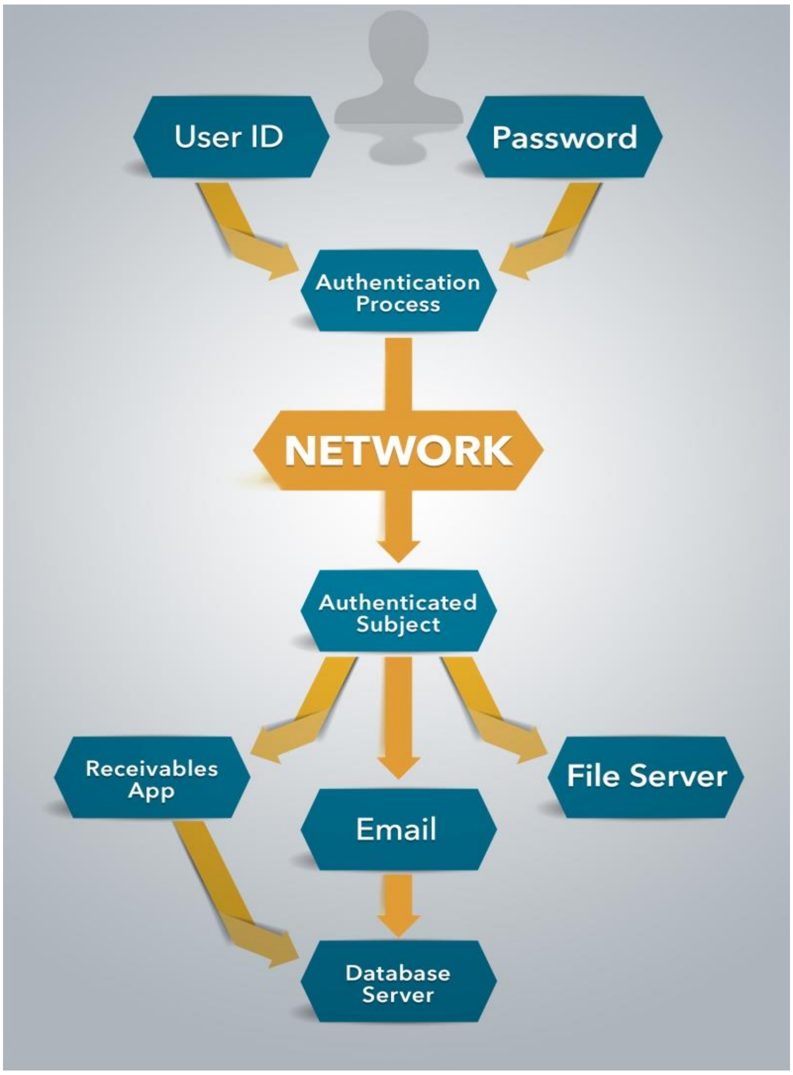


Figure 2. System/Work Flow Diagram

3.3 External Interface Requirements

3.3.1 Software Interfaces

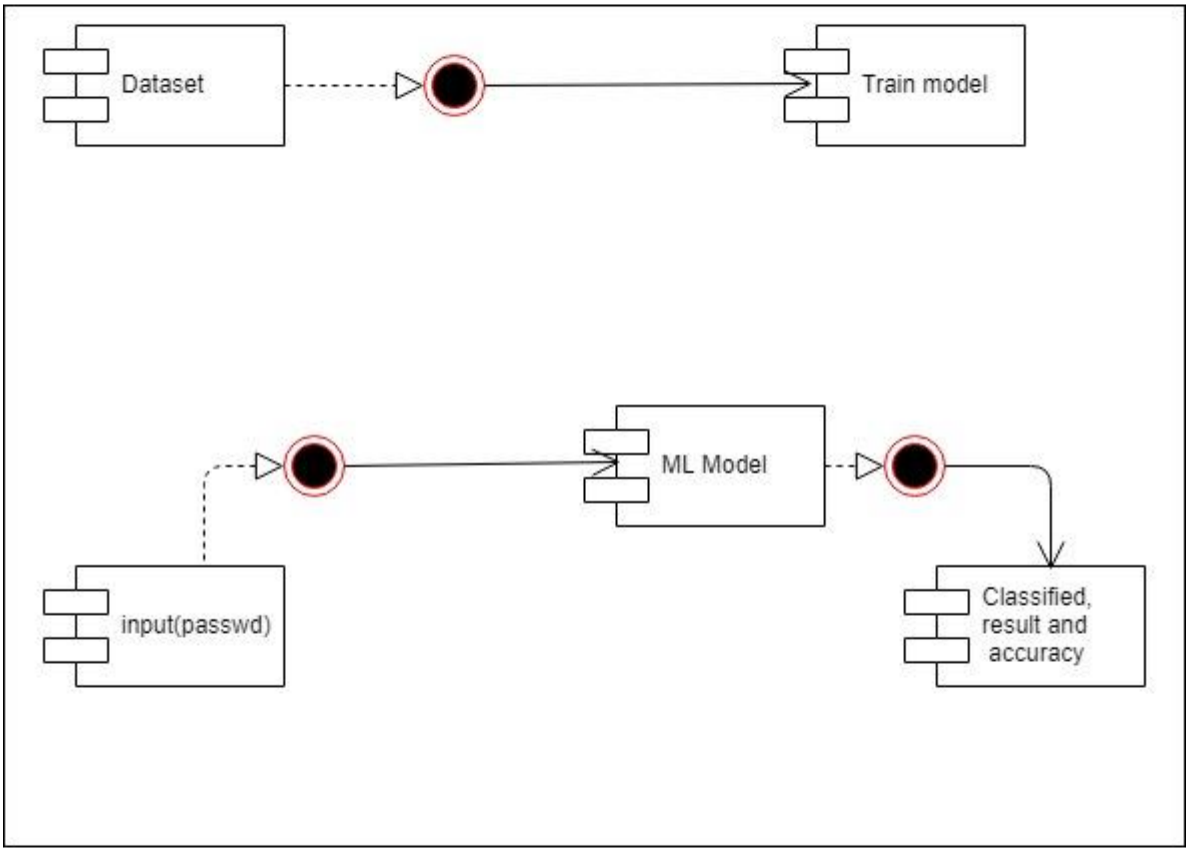


Figure 3. Software Interface

The above component diagram describes all the interfaces that present in the software system.

- Machine learning model
- The system must have NodeJS installed.
- Npm version is >= 5.6.0 and node version is >= 9.4.0

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

- The software should take minimum time for detecting the correct user.
- The User Interface should be very high so as to utilize majority of the processors resources.
- Availability of servers should be increased. So that no problem will arrive if one of the server goes down.

3.4.2 Security Requirements

- Effective encryption algorithms should be used while the data is being transmitted from client to server and vice versa.

3.4.3 Software Quality Attributes

Following Software quality attributes should be achieved:

- Adaptability
- Availability
- Correctness
- Flexibility
- Interoperability
- Maintainability
- Portability
- Reliability
- Reusability
- Robustness
- Testability
- Usability

3.5 System Implementation Plan

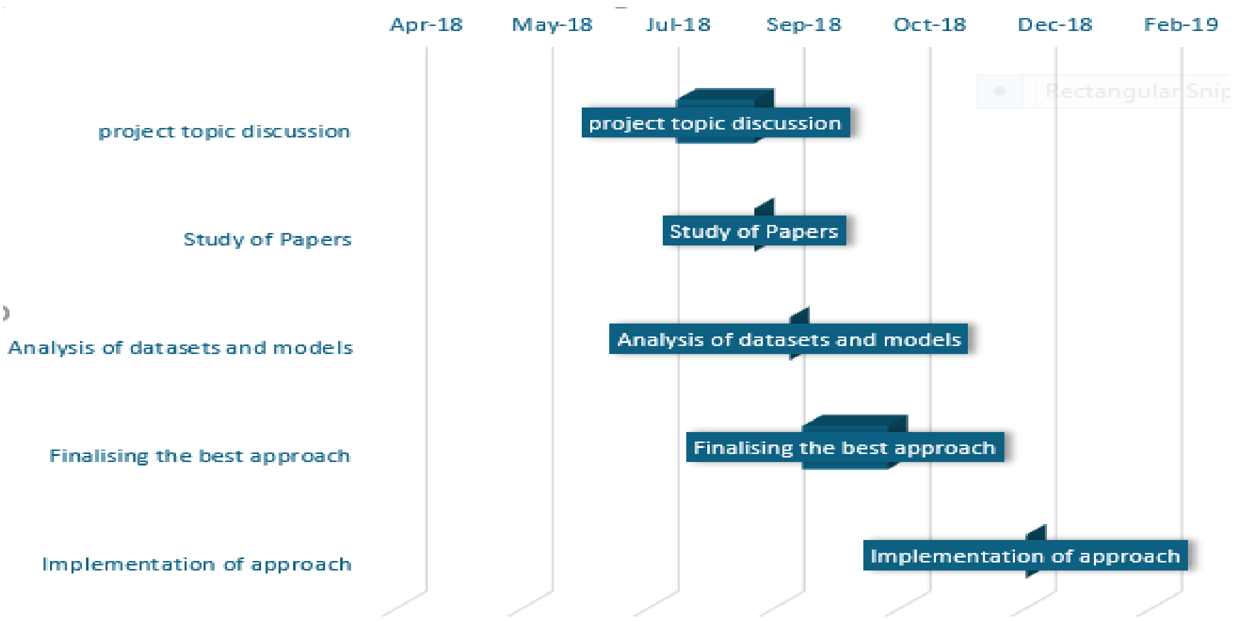
3.5.1 Cost Estimate

All the software used are open source. Extra cost might incur if the project is expanded further for heavy computations.

3.5.2 Time Estimates

PHASE	Time
Conceptualization and research	1 month
Requirement Analysis	1 month
System Design	2-3 month
Implementation	4-5months
Testing	1 month
Improvement	2 weeks
Deployment	1 week

Table 1. Time Estimate



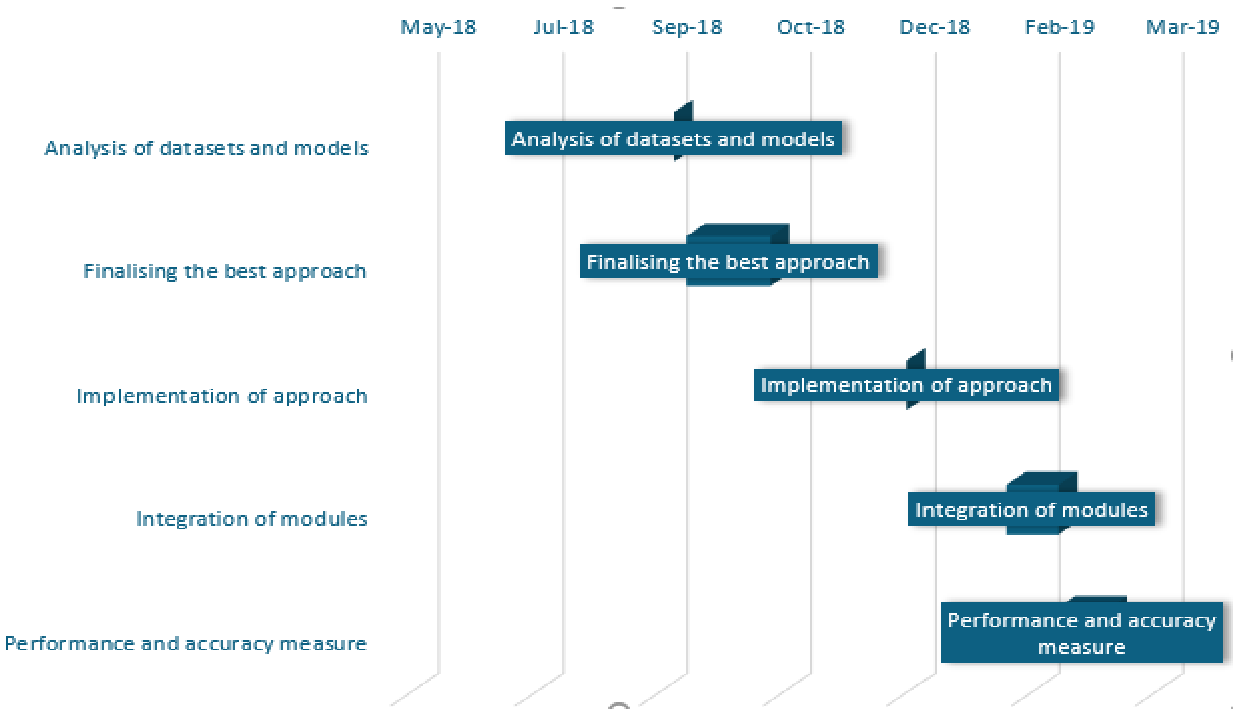


Figure 4. Timeline

3.5.3 Project Resources

1. Development team: 4 members.
2. Hardware Resources:

   Window/Linux based system, Network connectivity.

   The minimum hardware specification are as follows:

   - CPU -Pentium dual core processor or higher
   - RAM-4GB ram or more
   - Storage- 500GB hard disk or more
   - LAN setup

3. Software Resources:

NodeJs

python

3.6 **RISK MANAGEMENT**

3.6.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned below. Please refer table below for all the risks. You can have refereed following risk identification questionnaire.

1. Have top software and customers formally committed to support the project.
2. Are end-users enthusiastically committed to the project and the system/product to be built?
3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirement?
5. Do end-users have realistic expectations?
6. Does the software engineering team has the right mix of skills?
7. Are project requirements stable?
8. Is the number of people on the project team adequate to do the job?
9. Do all customers/user constituencies agree on the importance of the project and on the requirements for the system/product to be build?

3.6.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Estimated Project Schedule	High	High	High	High
2	Project Scope Creep	Low	Low	Medium	Medium
3	Project Team Availability	Medium	Medium	High	High
4	Person Hours	High	High	High	High
5	Timeline Estimates Unrealistic	Medium	Medium	Medium	Medium
6	Poor Functional Match of Package to Initial System Requirements	Low	Low	Low	Low

Table 2. Risk Description

Probability	Value	Description
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 - 75%
Low	Probability of occurrence is	< 25%

Table 3. Risk Probability Definition

Impact	Value	Description
Very high	> 10%	Schedule impact or Unacceptable quality
High	5 - 10%	Schedule impact or Some parts of the project have low quality
Medium	< 5%	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 4. Risk Impact Definition

3.6.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Risk ID	1
Risk Description	Estimated Project Schedule
Category	Development Environment
Source	Software requirement Specification document
Probability	High
Impact	High
Response	Mitigate
Strategy	Created comprehensive project timeline with frequent baseline reviews
Risk Status	Anticipated

Risk ID	2
Risk Description	Project Scope Creep
Category	Requirements
Source	Software Design Specification documentation review
Probability	Low
Impact	Medium
Response	Mitigate
Strategy	Scope initially defined in project plan, reviewed monthly by project guide
Risk Status	Identified

Risk ID	3
Risk Description	Project Team Availability
Category	Technology
Source	This was identified during early development and testing
Probability	Medium
Impact	Very High
Response	Accept
Strategy	Continuous review of project momentum by all levels.If necessary, increase committment by participation to full time status
Risk Status	Identified

Risk ID	4
Risk Description	Person Hours
Category	Requirements
Source	Software Design Specification documentation review
Probability	High
Impact	High
Response	Mitigate
Strategy	Comprehensive project management approach and communications plan
Risk Status	Identified

Risk ID	5
Risk Description	Timeline Estimates Unrealistic
Category	Requirements
Source	Software Design Specification documentation review
Probability	Medium
Impact	Medium
Response	Mitigate
Strategy	Timeline reviewed monthly by the members of the group and Project guide to prevent undetected timeline departures
Risk Status	Identified

Risk ID	6
Risk Description	Poor Functional Match of Package to Initial System Requirements
Category	Requirements
Source	Software Design Specification documentation review
Probability	Low
Impact	Low
Response	Mitigate
Strategy	Use of Intranet project website, comprehensive Communication plan
Risk Status	Identified

Table 5. Overview of Risks

3.7 PROJECT SCHEDULE

3.7.1 Project task set

Tasks in the Project stages are:

- Project Title Selection
- Discussion & Proposal
- Proposal submission
- Literature Survey
- Proposal Presentation
- Proposal Design
- Software Development
- Test Plan
- Testing and QA

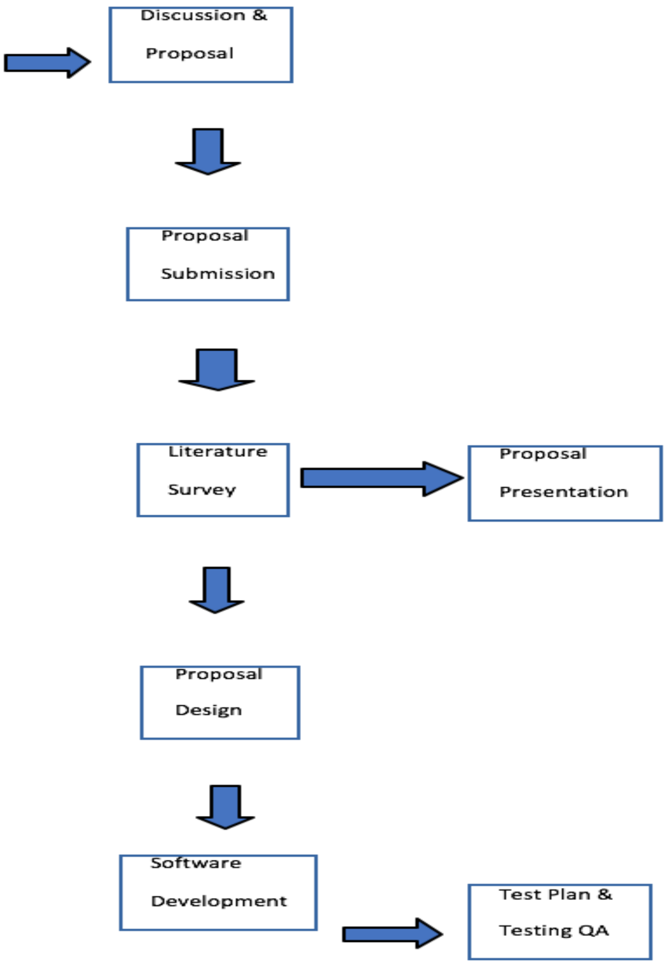


Figure 5. SDLC Model

4 System Design

4.1 System Architecture

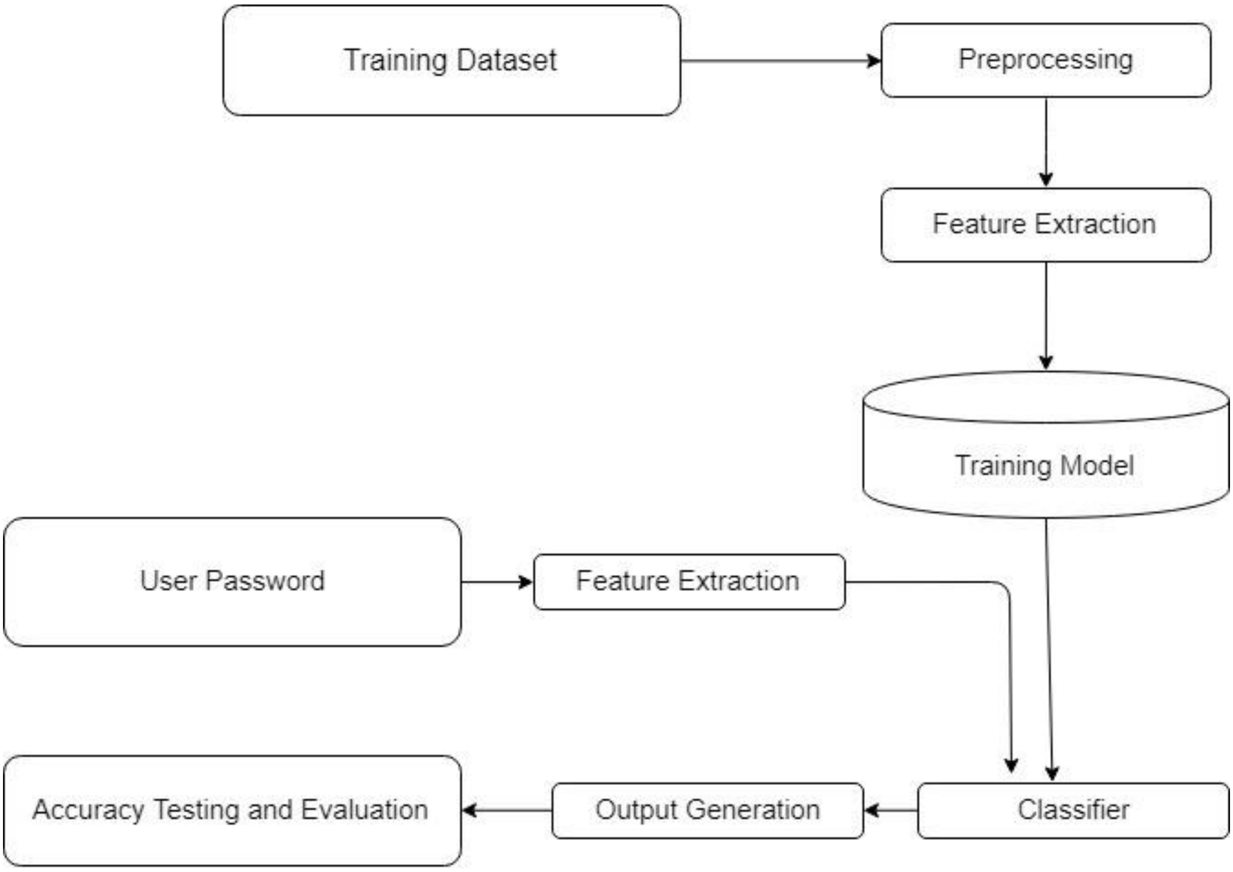


Figure 6. System Architecture

4.2 Data Flow Diagram

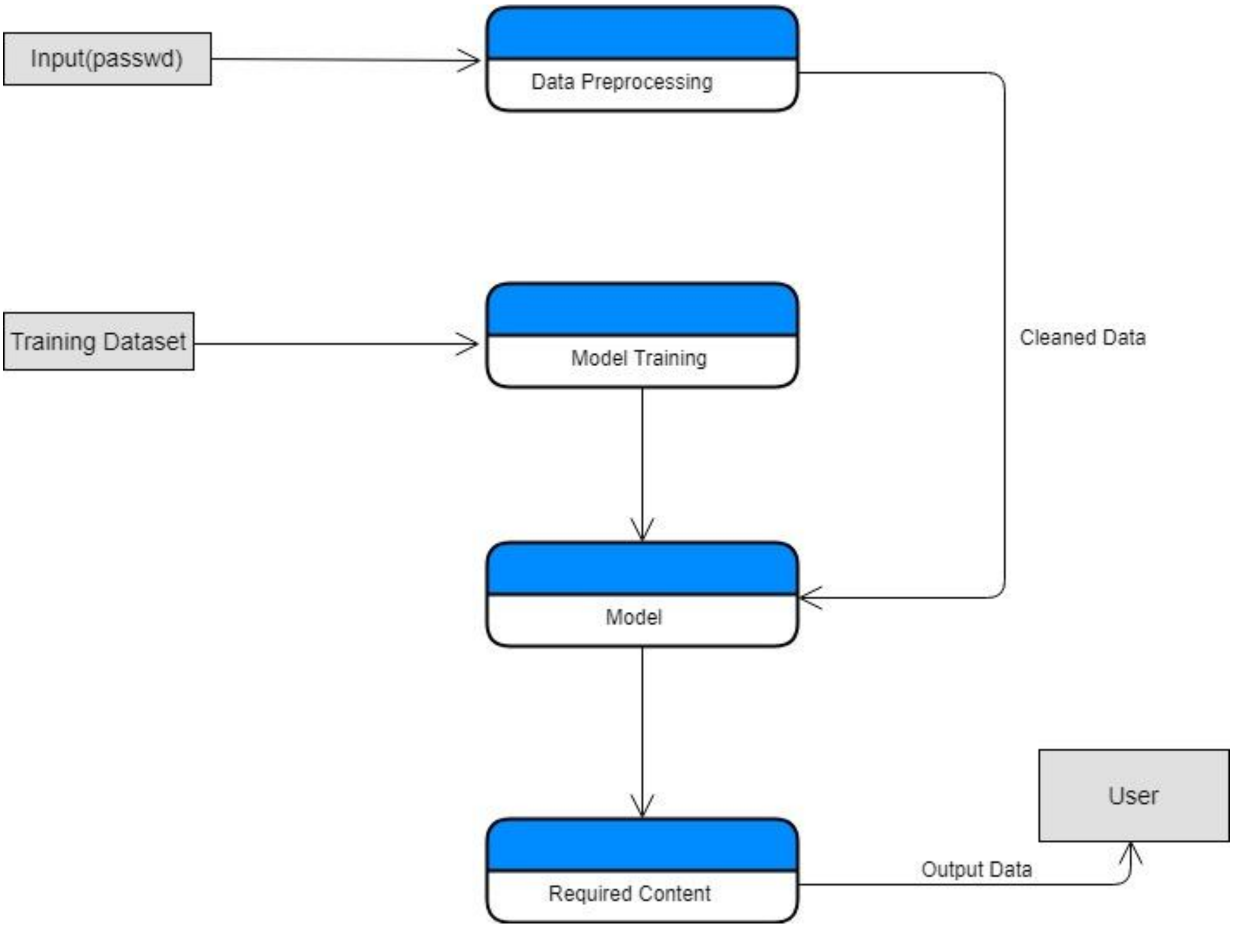


Figure 7. Data Flow Diagram

4.3 **UML DIAGRAMS**

4.3.1 Use case Diagram

Use Case ID	01
Use Case Name	Authentication System(registration)
Description	Data to be inputted to the web system
Actors	User(Customer)
Precondition	Dataset not trained for the user
Primary Sequence	1.User enters detail in the form 2.Data is sent to server 4.Server preprocess the data and trains the model.
Post Condition	Training is successfully done
Priority	High
Frequency of Use	High
Normal Steps	1.Register 2.Enter details 4.Click Submit

Use Case ID	02
Use Case Name	Authentication System(verification)
Description	user data to be verified by the model for prediction
Actors	User
Precondition	User enters the password
Primary Sequence	1.password sequence with speed captured 2.prediction by the model
Post Condition	Prediction successfully done

Priority	High
Frequency of Use	High
Normal Steps	1. Login 2.Authenticate

Table 6. Use Case Tables

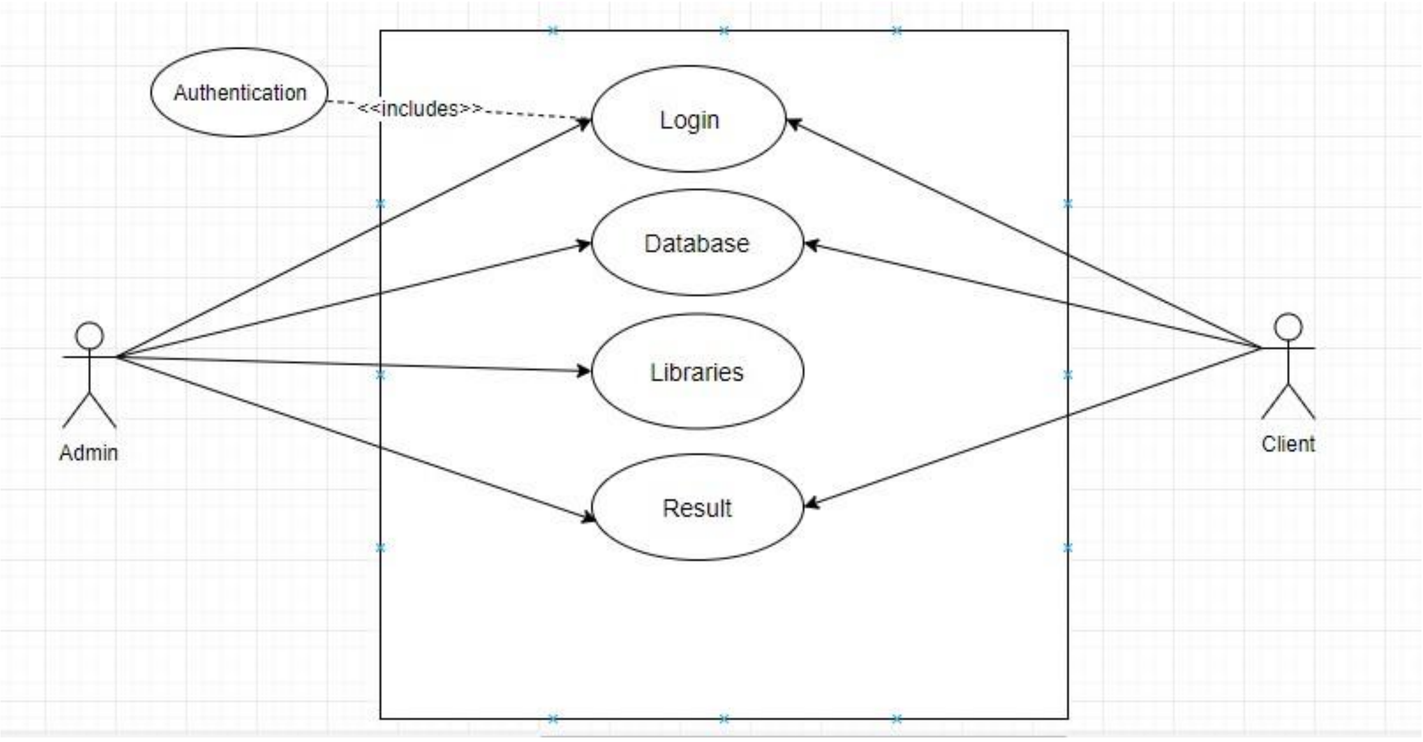


Figure 8. Use case Diagram

4.3.2 Sequence Diagram

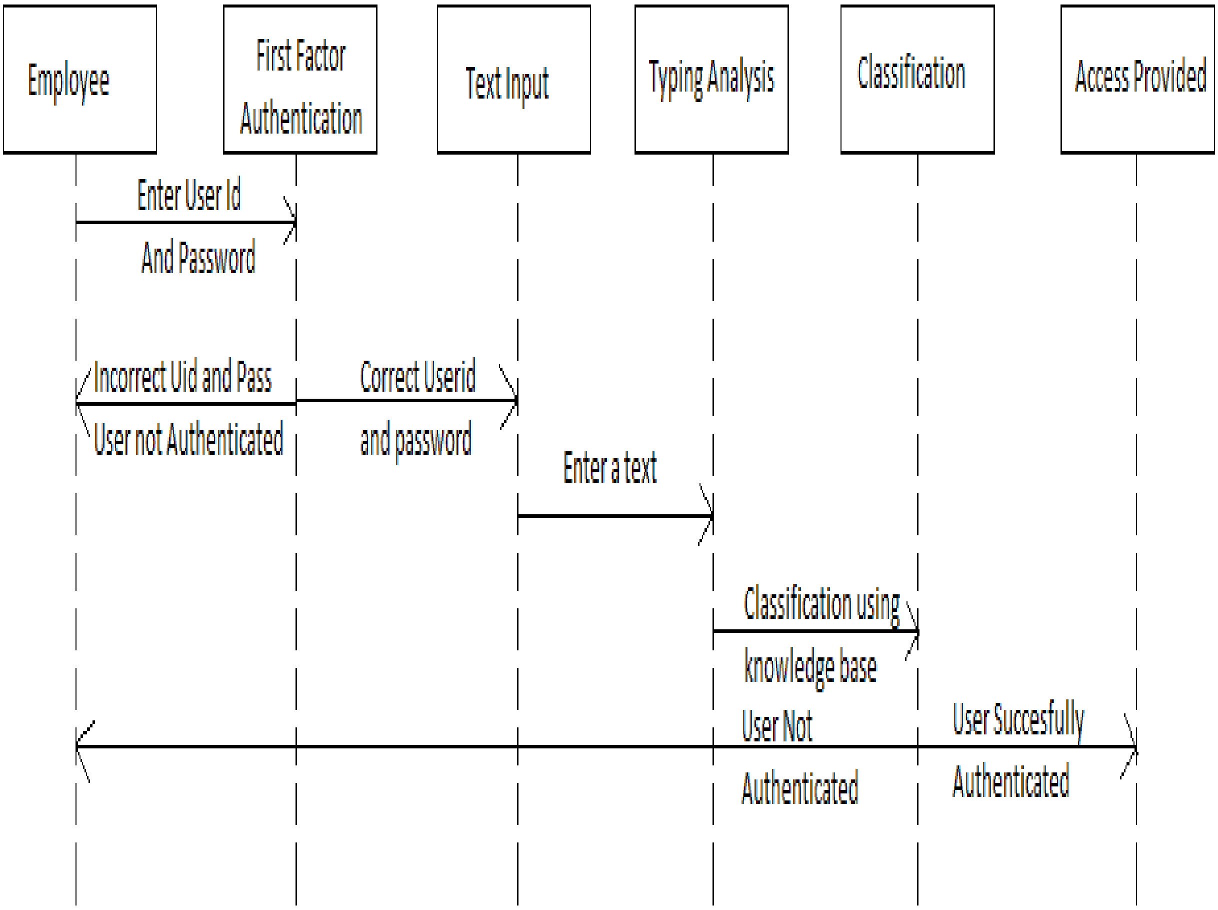

Employee
First Factor Authentication
Text Input
Typing Analysis
Classification
Access Provided
Enter User Id And Password
Incorrect Uid and Pass User not Authenticated
Correct Userid and password
Enter a text
Classification using knowledge base
User Not Authenticated
User Succesfully Authenticated

Figure 9. Sequence Diagram

4.3.3 State Transition Diagram

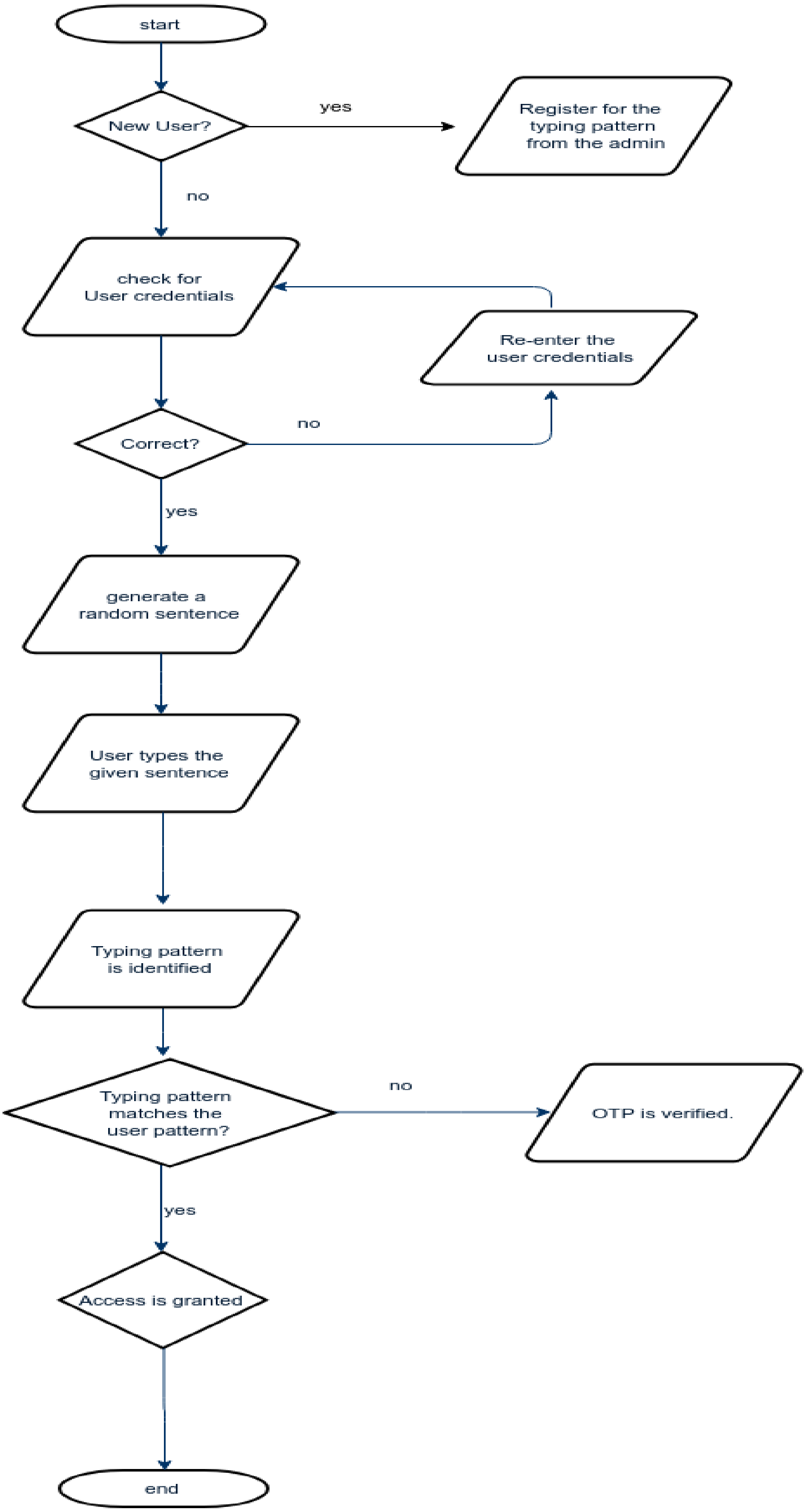

start
New User?
yes
Register for the typing pattern from the admin
no
check for User credentials
Re-enter the user credentials
Correct?
no
yes
generate a random sentence
User types the given sentence
Typing pattern is identified
Typing pattern matches the user pattern?
no
OTP is verified.
yes
Access is granted
end

Figure 10. State Transition Diagram

5 IMPLEMENTATION

5.1 Overview Of Project Modules

The project modules are divided into various categories based on various functionalities delivered by these modules :

Signup() : In this module we are taking the new user name and password from the user and while he is typing his password we collect the data and put it in a csv file.

Train() : In this module we are taking the user name and password from the user and while he is typing his password we collect the data and put it in a csv file.

Test() : : In this module testing the password and username typed by the person against the data collected.

GetTimeDifference() : In this module we will be collecting the time difference of each key letter user types.

Predict() : : In this module the password and the typing pattern corresponds to the user name is predicted.

5.2 Tools And Technologies used

Various tools and technologies used in our project are :

1. **NodeJs :Node.js** is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

2. **Python :Python** is an interpreted, high-level, general-purpose programming language. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

3. **Numpy :**NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4. **Pandas :**In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

5.3 Algorithm Details

Authentication using Behavioural Analytics can be implemented using various approaches like like considering their typing behaviour, the way people interact with the system just after login like someone will open his mail while some other person used to check where he left, other approach could be using sensors such as gyroscope in mobile that tells us the orientation of the phone so when we walk each user have different walking style so different readings; these can be used to detect as well as authenticate the right user with minimal efforts from user. In our case we will be focusing on keyboard parameter where we will use data regarding the way a person types in his unconscious mind to authenticate the user.

Keyboard typing speed differs from person to person when he his typing in his usual manner that is to say when he is typing with his unconscious mind. We will be taking parameter such as typing difference between pressing of two key letters. To say if a person types ‘My name is Ram’ , we will be collecting the time difference of each key letter he type the difference between M and y to be noted then y and n similarly for the whole sentence. Each persons data is collected by asking the person to type his password say 70 times so that we could have enough dataset for prediction. There may be case when system or model to say is unable to predict automatically at that time it will use second factor that is email authentication to authenticate the user.

5.4 Basic Approach

5.4.1 Web Page for typing:
First, we will take input password with the help of web page.

5.4.2 Pre-processing:
Steps involves computing the time differences of each letter the user types except for the keys ‘Enter’ and ‘Shift’.

5.4.3 Building the SVM model :
At last the SVM model is build which is a classifier that finds a hyperplane or a function g(x)=w^t*x+b that correctly separates two classes with a maximum margin.

5.4.4 Prediction From model :
Here we are predicting the user seeing his password pattern if it has less accuracy then second factor authentication activates which then asks for an otp sended to his email.

6 SOFTWARE TESTING

6.1 Type of testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc.

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

6.2 Testing applied in our project:

- Unit Testing : In this type of testing we have applied testing over the individual modules of our project which are
  a. Signup module : We took new user data from the user and its password.
  b. Email module : This module checks whether the email is being sent to user on authentication failure.
  c. GetTimeDifference module : This module stores the time differences of each letter being pressed by user.

- Integration testing : All the above modules were integrated together and then testing was performed by giving username and password as input and related content along with successful authentication as output.

- System testing : This is final testing of the working in which the algorithm is tested on a fully functioning system connected to server.

7 OUTCOMES/RESULTS

7.1 Outcomes:

Throughout our research and implementation there were a few observations for our models:

1.We take the typing pattern for each user atleast 70 times.

2. Comparison of various algorithm used to implemented and tried :

LEARNING METHODS	TRAINING	TEST ACCURACY
Decision Tree	95.6%	93.3%
Naïve Bayes	93.3%	90.8%
KNN	91.3%	75.6%
SVM	97.2%	96.8%

Table 7. Comparisons

7.2 Screenshots:

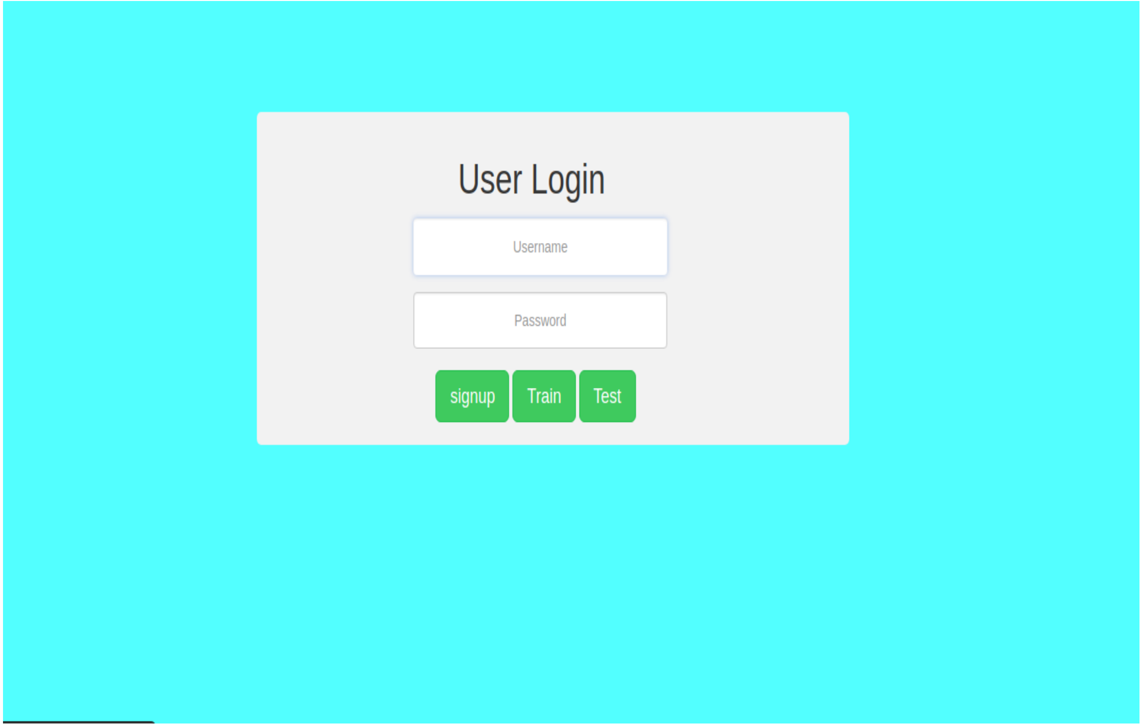
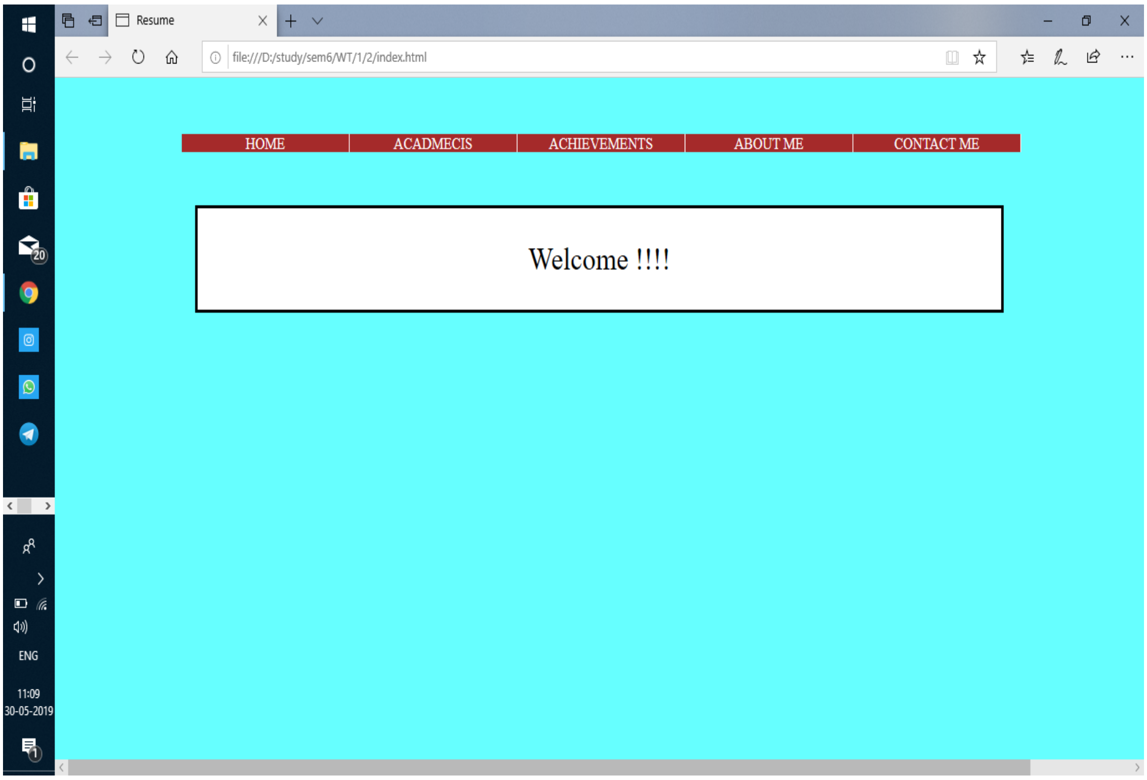


Figure 11 Front Page



Welcome !!!!

Figure 12 Login Successful Page

8 CONCLUSIONS AND FUTURE WORK

User Behavior analytics is the current boom in the field of research and IT. The Proposed system is an appropriate replacement for rule based authentication system. It is expected that our approach will achieve much better results. The proposed system provides all the necessary features of a security system. This system is a cost-effective as no external hardware is used such as biometrics for a reliable authentication system. The proposed system is cost-efficient as compared to other security architectures. The architecture also benefits the users in terms of usability, and trust. We might get high performance of our classifier but there is definitely scope for improvement. We evaluated our models using absolute probability thresholds, which may not be the most reliable for models where probability scoring is not well calibrated.

APPENDIX A

In this appendix, we focused on problem statement feasibility considering its complexities, its advantages disadvantages over other to solve the problem. Basically, two algorithms that we focused on are: -

Support Vector Machine(SVM)

The SVM technique is a classifier that finds a hyperplane or a function g(x)=w^t*x+b that correctly separates two classes with a maximum margin .Mathematically speaking, given a set of points *x*i that belong to two linearly separable classes ω1, ω2, the distance of any instance from the hyperplane is equal to |g(x)|/PwP . SVM aims to find *w*, *b*, such that the value of g(*x*) equals 1 for the nearest data points belonging to class ω1 and –1 for the nearest ones of ω2. This can be viewed as having a margin of

$$\frac{1}{\text{P}w\text{ P}} + \frac{1}{\text{P}w\text{ P}} = \frac{2}{\text{P}w\text{ P}},$$

Whereas

$$w^T x + b = 1 \text{ for } x \in \omega_1,$$

$$w^T x + b = -1 \text{ for } x \in \omega_2$$

This leads to an optimization problem that minimizes the objective function

$$J(w) = \frac{1}{2}\text{P}w\text{ P}^2,$$

subject to the constraint

$$y_i\left(w_i^T x + b\right) \geq 1,\ i = 1, 2, \ldots, N.$$

Decision Tree

CART and decision trees like algorithms work through recursive partitioning of the training set in order to obtain subsets that are as pure as possible to a given target class. Each node of the tree is associated to a particular set of records T that is splitted by a specific test on a feature. For example, a split on a continuous attribute A can be induced by the test $A \leq x$. The set of records T is then partitioned in two subsets that leads to the left branch of the tree and the right one.

$Tl=\{t \in T:t(A) \leq x\}$

and

$Tr=\{t \in T:t(A)>x\}$

Similarly, a categorical feature B can be used to induce splits according to its values. For example, if $B=\{b1,\ldots,bk\}$ each branch i can be induced by the test B=bi. The divide step of the recursive algorithm to induce decision tree takes into account all possible splits for each feature and tries to find the best one according to a chosen quality measure: the splitting criterion. If your dataset is induced on the following scheme

$$A1,\ldots,Am,C$$

where Aj are attributes and C is the target class, all candidates splits are generated and evaluated by the splitting criterion. Splits on continuous attributes and categorical ones are generated as described above. The selection of the best split is usually carried out by impurity measures. *The impurity of the parent node has to be decreased by the split*. Let (E1,E2,…,Ek) be a split induced on the set of records E, a splitting criterion that makes used of the impurity measure I(·)is:

$$\Delta = I(E) - \sum |Ei|/|E| * I(Ei)$$

Standard impurity measures are the Shannon entropy or the Gini index. More specifically, CART uses the Gini index that is defined for the set E as following. Let pj be the fraction of records in E of class *cj*

$$pj=|\{t \in E:t[C]=cj\}|/|E|$$

then

$$Gini(E)=1-\sum (Pj)^2$$

where Q is the number of classes.

It leads to a 0 impurity when all records belong to the same class.

APPENDIX B

Details of Paper Publication

Our paper 'Two Factor Authentication using User Behavioural Analytics' bearing paper id 'IJCSI-2019-16-3-12414' has been accepted for publication in IJCSI Volume 16, Issue 1, March 2019 which is scheduled to be published on Saturday 08th June 2019.

REFERENCES

1. Shepherd, S. J. Continuous authentication by analysis of keyboard typing characteristics.Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." .

2. Panasiuk, P., & Saeed, K.. A modified algorithm for user identification by his typing on the keyboard. In *Image Processing and Communications Challenges 2* (pp. 113-120). Springer, Berlin, Heidelberg.

3. Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.

4. Juola, P., Noecker, J. I., Stolerman, A., Ryan, M. V., Brennan, P., & Greenstadt, R. . Keyboard-behavior-based authentication. *IT Professional*, *15*(4), 8-11.

5. Banerjee, S. P., & Woodard, D. L. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, *7*(1), 116-139.