ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 13, April 2016

User Authentication using One Class Learning Algorithm in Mouse Behavior

Ms.S.HemaMalini,M.Tech
Dept. of Computer Science and Engineering
SRM University
Kattankulathur, Chennai
hemamalinicse@gmail.com

Mr.C.SanthanaKrishnan,Assistant Professor
Dept. of Computer Science and Engineering
SRM University
Kattankulathur, Chennai
santhanakrishnan.c@ktr.srmuniv.ac.in

***Abstract*—the quest for a reliable and convenient security mechanism to authenticate a computer user has existed since the inadequacy of conventional password mechanism was realized, first by the security community, and then gradually by the public. As data are moved from traditional localized computing environment to the new Cloud Computing paradigm, the need for better authentication has become more pressing. Recently, several large scale password leakages exposed users to an unprecedented risk of disclosure and abuse of their information. These incidents seriously shook public confidence in the security of the current information infrastructure; the inadequacy of password-based authentication mechanisms is becoming a major concern for the entire information society. Various potential solutions to this problem, a particularly promising technique is Mouse Dynamics. Mouse Dynamics measures and assesses a user's Mouse Behavior characteristics for use as a Biometric. Compared with other biometrics such as face, finger point and voice, Mouse Dynamics is less intrusive, and requires no specialized hardware to capture biometric information. Hence it is suitable for the current internet environment. When a user tries to log into a computer system, Mouse Dynamics only requires user to provide the login name and to perform a certain sequence of Mouse operations. Extracted behavioral feature's, based on mouse movements and clicks are compared to a legitimate user's profile. A match authenticates the user; otherwise user access is denied. Furthermore, a user's mouse behavior characteristics can be continually analyzed during user subsequent usage of a computer system for identity monitoring or intrusion detection.**

***Keywords*—Data leakage, Mouse Dynamics, Authentication.**

I. INTRODUCTION

Several large-scale password leakages expose users to an unprecedented risk of disclosure and abuse of their information. The inadequacy of password- based authentication mechanisms is becoming a major concern for the entire information society. Consist of three major phases(1) Mouse-Behavior Capture, (2) Feature Construction and (3) Training/Classification. The first module serves to create a mouse-operation task, and to capture and interpret mouse behavior data. The second module is used to extract holistic and procedural features to characterize mouse behavior and to map the raw features into distance based features by using various distance metrics. The third phase is training phase, applies neural network on the distance based feature vectors to compute the predominant feature components, and then builds the user's profile using a one class classifier. In the classification phase, it determines the user's identity using the trained classifier in the distance based feature using Neural Network. A 4 digit OTP is generated to the user's email ID. The user will be giving the „2" digit OTP and the server will be giving balance „2" digit OTP. Users „2" digit OTP is verified by the server and vice versa.

II. ONLINE SIGNATURE VERIFICATION

This section describes the main stages involved in the online signature verification algorithm proposed in this paper, which can be divided into: template, statistical and structural matching.

A. Template matching techniques:

Effective method for Template Matching is dynamic time warping (DTW) algorithm. DTW becoming one of the most popular methods used in signature competitions. Aims to minimize the effects of distortion and time shift between two signatures collected in different sessions.

B. Statistical matching techniques:

This technique consists of estimating a statistical model that represents the most salient behavioral features of a particular signature using Neural Networks.

C. Structural matching techniques:

It"s based on syntactic approaches for representing signatures that are compared through graph or tree matching techniques.

III. PRE PROCESSING STAGES:

The aim of a preprocessing stage is to reduce noise and normalize the signature stroke. This process is widely documented. So that it will be only briefly reviewed here. The preprocessing stage is carried out following four steps:

3.1) Filtering: Signals acquired by the electronic device are smoothed by applying a low-pass filter that reduces noise introduced in the capturing process.

3.2) Equally-spacing: The average signals are transformed to an equally-spaced 256-point temporal sequence by using a linear interpolation.

3.3) Location and time normalization: The x-axis and y-axis temporal functions are normalized by centering the origin of coordinates at the signature centre of mass with a specific rotation.

3.4) Size normalization: The x and y strokes of the signature are normalized by using the norm of the 2 dimension vector [x,y]. Moreover, the dynamic characteristics such as pressure and inclination are also normalized by their maximum value.

An example of signature is shown in Fig. 2, where the x and y positions related to the pen stroke and the pen pressure are presented. The same signature, acquired by a different electronic device, can be captured ranging from different values or disagreeing with the number of points depending on the sample frequency. But, even using the same device, the signature is often affected by other variations like translations, rotations or duration time of writing. As can be seen in Fig. 3, the preprocessing stage removes these preliminary differences creating a normalized signature unaffected by these factors.

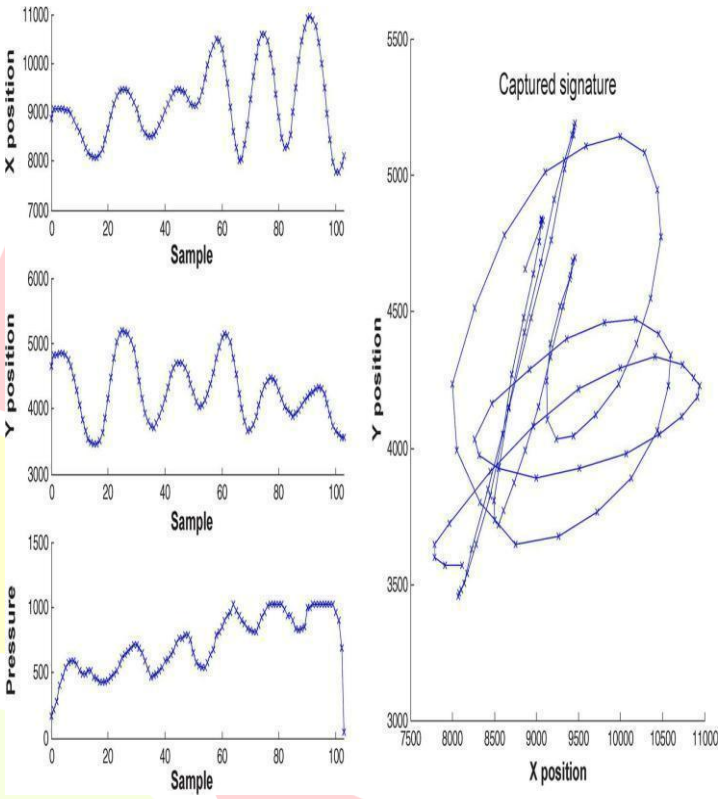


Fig.1. Captured signature after applying the preprocessing stage.

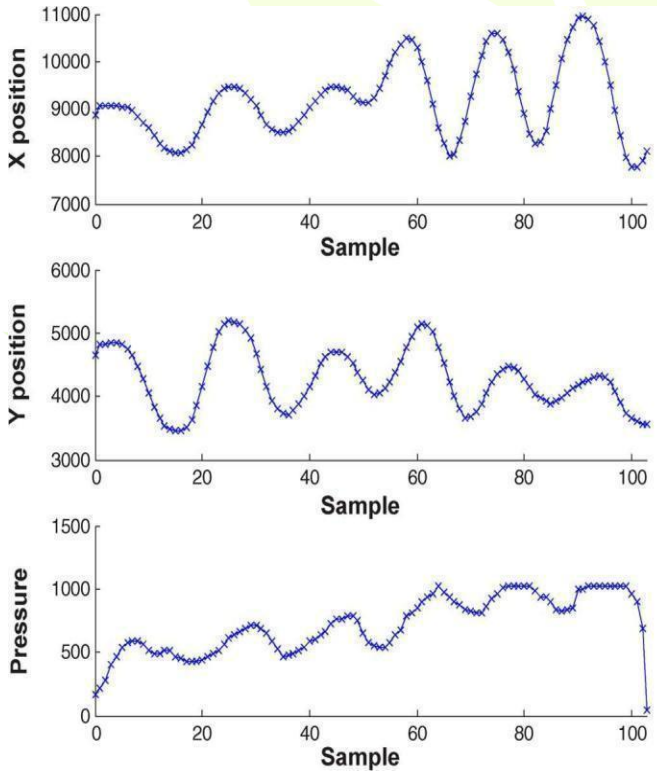
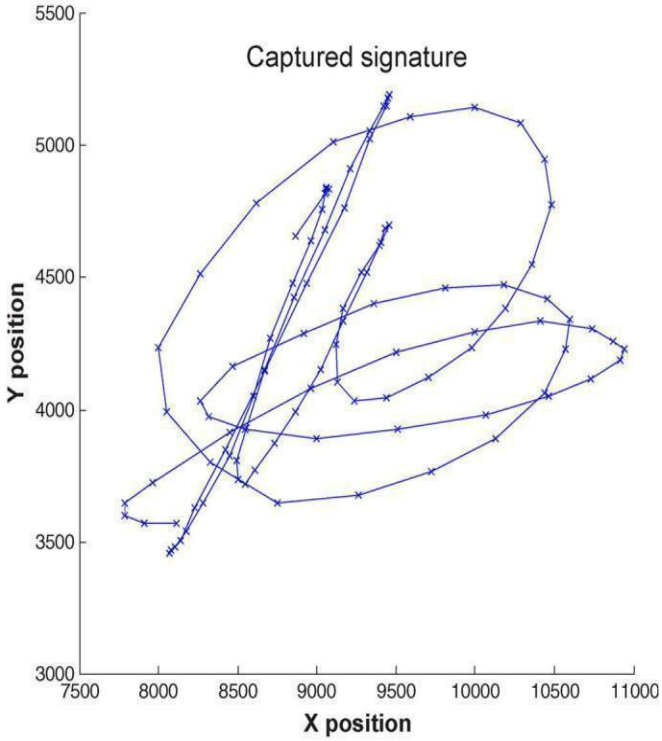


Fig. 2. a)Representation of signature captured by a commercial device.



b)Captured signature using commercial device.

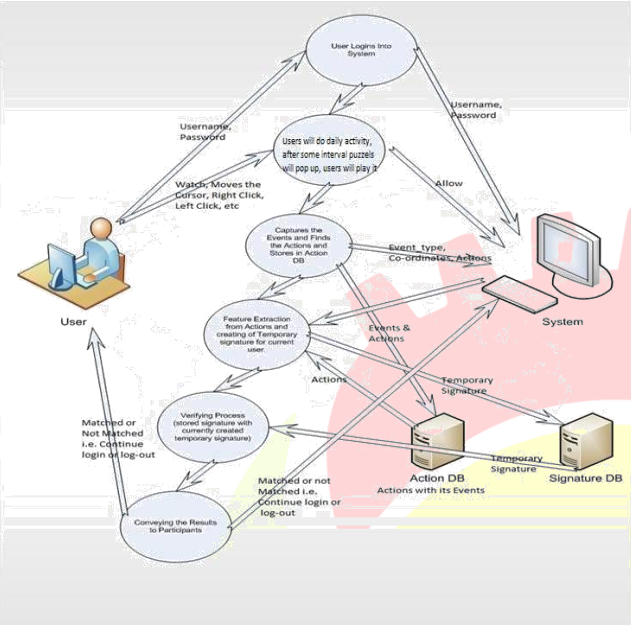


Fig:3, shows the Generic architecture for enrolment and matching processes.

PROBLEMS FACED:

Password leakages will reduce the customer's transaction in the Banking Environment. Password Guessing attacking will allow hackers to access the account easily. Sometimes, the password guessing attacks will led to the blocking of the customer's account and that will led the customer's unsatisfaction. Also the user can enter into phishing site, So that user's Sensitive information will be access by the hacker and they can use it on behalf of yourself.

IV. SYSTEM ANALYSIS

4.1) EXISTING SYSTEM

Many password leakages exposed users to an unprecedented risk of disclosure and misuse their information. These types of password-based authentication mechanisms is becoming a major concern for varieties of Security based applications. Also some attacks namely called, password guessing attacking has become more concern for the users, while accessing the some of the sensitive application like Bank transaction, Train Booking and Online Shopping.

4.2) *PROPOSED SYSTEM*

We are implementing the proposed System which is consisting of three major phases: (1) Mouse–Behavior Capture,(2) Feature Construction, and (3) Training / Classification. In the First Module, we'll create a mouse-operation task, and to capture and interpret mouse-behavior data. The second module is used to extract holistic and procedural features to characterize mouse behavior and to map the raw features into distance-based features by using various distance metrics. The third module, in the training phase, applies neural network on the distance-based feature vectors to compute the predominant feature components, and then builds the user's profile using a one-class classifier. In the classification phase, it determines the user's identity using the trained classifier in the distance-based feature using NN.

V.FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

5.1) ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.2) TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3) SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

5.4) OPERATIONAL FEASIBILITY

Operational feasibility is a test of feasibility that will check whether the systems are working when it is developed and installed in place of the existing system. The Proposed system is beneficial only if it can be turned into information system that will meet the organization's operational requirements

VI. SYSTEM TESTING

Software testing is an important element of Software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of S/W as a system element and the costs associated with Software failure are motivating forces for well planned, through testing.

1) *Fig:4) white box testing:*

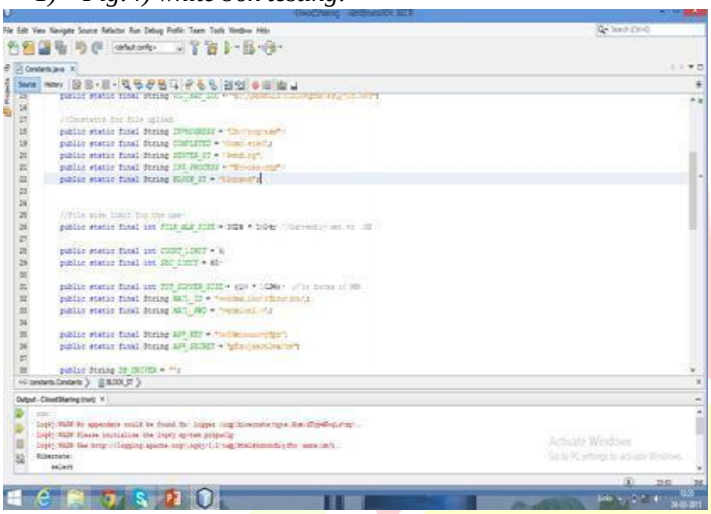


TABLE-I

2) Black Box testing

Test Module	Purpose	Input	Expected Output	Actual output	Status
1.Registration	Register for database	User details	Successful registration	Successful registration	Pass
1.Registration	Register for database	User details	Successful registration	User name already exist	Fail
2.Authentication	Login	User name, password & signature	Successful login	Successful login	Pass
2.Authentication	Login	User name, password & signature	Successful login	Invalid User Name, password or signature not matched	Fail
3.Transaction	Transact money for user	Transaction id and OTP	Successful transaction	Successful transaction	Pass
3.Transaction	Transact money for user	Transaction id and OTP	Successful transaction	OTP mismatch	Fail

VII. DATA BASE CONNECTIVITY

```
Coding importjava.sql.*;
importcom.microsoft.sqlserver.jdbc.*;

publicclassSQLDatabaseTest {

publicstaticvoid main(String[] args) { StringconnectionString
                +     "jdbc:sqlserver://your_server.database.windows.net:
                      1433;"

+ "database=your_database;"
+ "user=your_user@your_server;"
+ "password=your_password;"
+ "encrypt=true;"
+ "trustServerCertificate=false;"
+ "hostNameInCertificate=*.database.windows.net;
  "
+ "loginTimeout=30;";
      =     Declare the JDBC objects.
Connectionconnection = null; Statementstatement =
null; ResultSetresultSet = null;
PreparedStatementprepsInsertPerson = null;
PreparedStatementprepsUpdateAge = null; try
{
            connection =
            DriverManager.getConnection(connectionString);
             //INSERT two rows into the table.
             //TRANSACTION and commit for an UPDATE.
             //SELECT rows from the table.
    }
catch (Exception e) {
e.printStackTrace();    }
finally {
            //Close the connections after the data has been
handled.
if (prepsInsertPerson != null) try {
prepsInsertPerson.close(); } catch(Exception e) {}
if (prepsUpdateAge != null) try { prepsUpdateAge.close(); }
catch(Exception e) {}
if (resultSet != null) try { resultSet.close(); }
catch(Exception e) {}
if (statement != null) try { statement.close(); }
catch(Exception e) {}
if (connection != null) try { connection.close(); }
catch(Exception e) {}
          }          }
```

VIII. MACHINE LEARING PROBLEMS

8.1) Machine Learning Problems :

Machine Learning problems are abound. They make up core or difficult parts of the software you use on the web or on your desktop every day. Think of the "do you want to follow" suggestions on twitter and the speech understanding in Apple‟s Siri.

8.2) Spam Detection: Given email in an inbox, identify those email messages that are spam and those that are not. Having a model of this problem would allow a program to leave non-spam emails in the inbox and move spam emails to a spam folder. We should all be familiar with this example.

8.3) Credit Card Fraud Detection: Given credit card transactions for a customer in a month, identify those transactions that were made by the customer and those that were not.

8.4) Medical Diagnosis: Given the symptoms exhibited in a patient and a database of anonym zed patient records, predict whether the patient is likely to have an illness. A model of this decision problem could be used by a program to provide decision support to medical professionals.

8.5) Stock Trading: Given the current and past price movements for a stock, determine whether the stock should be bought, held or sold. A model of this decision problem could provide.

8.6) Customer Segmentation: Given the pattern of behavior by a user during a trial period and the past behaviors of all users, identify those users that will convert to the paid version of the product and those that will not. A model of this decision problem would allow a program to trigger customer interventions to persuade the customer to covert early or better engage in the trial. A program with a model of this decision could refund those transactions that were fraudulent.

8.7) Digit Recognition: Given a zip codes hand written on envelops, identify the digit for each hand written character. A model of this problem would allow a computer program to read and understand handwritten zip codes and sort envelops by geographic region.

8.8) Speech Understanding: Given an utterance from a user, identify the specific request made by the user. A model of this problem would allow a program to understand and make an attempt to fulfil that request. The iPhone with Siri has this capability.

8.9) Face Detection: Given a digital photo album of many hundreds of digital photographs, identify those photos that include a given person. A model of this decision process would allow a program to organize photos by person. Some cameras and software like iPhoto has this capability.

8.10) Product Recommendation: Given a purchase history for a customer and a large inventory of products, identify those products in which that customer will be interested and likely to purchase. A model of this decision process would allow a program to make recommendations to a customer and motivate product purchases. Amazon has this capability. Also think of Facebook, GooglePlus and Facebook that recommend users to connect with you after you sign.

8.11) Shape Detection:

Given a user hand drawing a shape on a touch screen and a database of known shapes, determine which shape the user was trying to draw. A model of this decision would allow a program to show the platonic version of that shape the user drew to make crisp diagrams. The Instaviz iPhone app does this.

These 10 examples give a good sense of what a machine learning problem looks like. There is a corpus of historic examples, there is a decision that needs to be modeled and a business or domain benefit to having that decision modeled and efficaciously made automatically.

Some of these problems are some of the hardest problems in Artificial Intelligence, such as Natural Language Processing and Machine Vision (doing things that humans do easily). Others are still difficult, but are classic examples of machine learning such as spam detection and credit card fraud detection.

Think about some of your interactions with online and offline software in the last week. I‟m sure you could easily guess at other ten or twenty examples of machine learning you have directly or indirectly used.

Types of Machine Learning Problems

Reading through the list of example machine learning problems above, I‟m sure you can start to see similarities.

This is a valuable skill, because being good at extracting the essence of a problem will allow you to think effectively about what data you need and what types of algorithms you should try.

There are common classes of problem in Machine Learning. The problem classes below are archetypes for most of the problems we refer to when we are *doing* Machine Learning.

1) ***Classification****:* Data is labeled meaning it is assigned a class, for example spam/non-spam or fraud/non-fraud. The decision being modeled is to assign labels to new unlabelled pieces of data. This can be thought of as a discrimination problem, modeling the differences or similarities between groups.

2) ***Regression****:* Data is labeled with a real value (think floating point) rather than a label. Examples that are easy to understand are time series data like the price of a stock over time, The decision being modeled is what value to predict for new unpredicted data.

3) ***Clustering****:* Data is not labeled, but can be divided into groups based on similarity and other measures of natural structure in the data. An example from the above list would be organizing pictures by faces without names, where the human user has to assign names to groups, like iPhoto on the Mac.

4) ***Rule Extraction****:* Data is used as the basis for the extraction of propositional rules (antecedent/consequent aka if-then). Such rules may, but are typically not directed, meaning that the methods discover statistically supportable relationships between attributes in the data, not necessarily involving something that is being predicted. An example is the discovery of the relationship between the purchase of beer and

diapers (this is data mining folk-law, true or not, it's illustrative of the desire and opportunity).

When you think a problem is a machine learning problem (a decision problem that needs to be modelled from data), think next of what type of problem you could phrase it as easily or what type of outcome the client or requirement is asking for and work backwards.

IX. INTRODUCTION OF ONE-CLASS LEARNING

One-Class Learning details

Koby Crammer & Gal Chechik, 2004

Gunjan Gupta & Joydeep Ghosh, 2005
Moshe Koppel & Jonathan Schler, 2004

Systems Biology aims at understanding a biological system as a whole. For instance, a cell or an organ as one unit artificial organ construction, Neural regeneration.

Bioinformatics = Biology + Informatics

Bioinformatics makes life sciences data more understandable and useful. A typical dictionary definition of Science: "The observation, identification, description, experimental investigation (scientific method), and theoretical explanation of phenomena.Such activities restricted to a class of natural phenomena...."(Excepted from The American Heritage Dictionary of the English Language, Third Edition 1996.)

Biology vs. Bioinformatics

Biology: Hypothesis driven and finding evidences, Inferring Science: Learn from a model system; try to explain a target system. Biologist may believe 'data' from wet lab if the data make a sense biologically.

Bioinformatics: With the Broad Definition: Any technical supporting activity toward to Biology.

Current Trend: Only accuracy is important. Do not care even , if the results tell a wrong story. Claiming doing Bioinformatics! With the consideration of Bioinformatics as a part of science: Much bigger setting than the traditional Biology. But, it is still hypothesis driven from existing data.

Machine Learning: Learning from a known data set, try to predict unknown instances."Inferring"-the most attractive word to biologists. So, Machine Learning is a perfect match for Biology nothing can be so sure on data. Some assumptions in MachineLearning.

Underlying assumptions in Machine Learning are not suitable for Biology Data Training data is not much diferent from the data to test.This is not always true in Biology (heterogeneous data).

Number of positive and negative examples are roughly same. This is not always true in Biology (imbalance data). Possible general approaches to overcome thses problems Transductive learning for heterogeneous data, Under sampling or Oversampling for imbalanced data.

General Setting of One-Class Learning
: Decision Process:

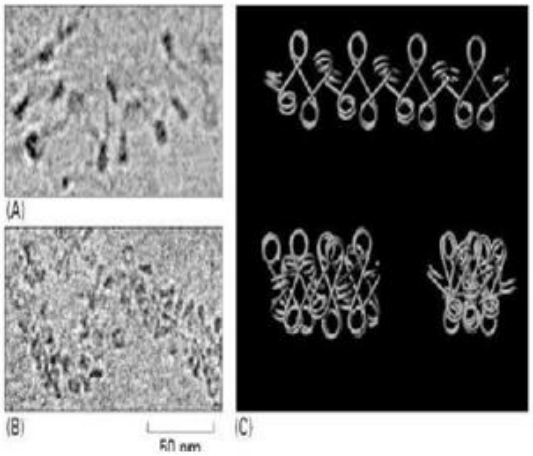
A measure for the distance d(x) or resemblance r (x) of an object x to target class, A threshold on this distance or resemblance new objects are accepted:

f (x) = I (d(x) < d
) or
f (x) = I (r (x) > r)

General Setting of One-Class Learning:

Error Definition: A method which obtains the lowest outlier rejection rate, fO, is to be preferred. For a target acceptance rate fT+ with respect to the threshold.

A Needle in a Haystack: Local One-Class Optimization



1: **Input:** A series of strongly convex functions $f_1, \ldots, f_T$.
2: **Initialize:** $\theta_1 = 0$
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Receive $\boldsymbol{x}_t$
5: Set $w_t = \nabla f_t^*(\theta_t)$
6: Predict $\hat{y}_t = sign(w_t^\top \boldsymbol{x}_t)$
7: Receive y_t
8: **if** $\ell_t(w_t) > 0$ **then**
9: $z_t = \partial \ell_t(w_t)$
10: $\theta_{t+1} = \theta_t - \eta_t z_t$
11: **else**
12: $\theta_{t+1} = \theta_t$
13: **end if**
14: **end for**

Corollary 1. *Define* $B = \sum_{t=1}^{T} (f_t^*$ *the hypothesis of Lemma 1, if* ℓ *is co the 0/1 loss, and* $\eta_t = \eta$*, then for a in Fig. 1 has the following bound on mistakes* M*,*

$$M \leq \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) + \frac{f_T(\boldsymbol{u})}{\eta} + \eta \sum_{t}$$

Moreover if $f_t(\boldsymbol{x}) \leq f_{t-1}(\boldsymbol{x}), \forall \boldsymbol{x} \in$ $B \leq 0$.

A similar bound has been recently gret bound. Yet, there are two differ bounds the number of mistakes, a classification setting, rather than of

Fig:5) Molecular Biology of the cell, 4th Edition

implementing these algorithms in microprocessors or FPGAs, roughly aims at two different approaches.

The first one solves the algorithm using software, by implementing a microprocessor that contains a floating-point unit (FPU) that carries out operations using scalar numbers. This proposal is quite flexible, since the same hardware architecture is used for resolving different parts of the algorithm. However, often mathematical operations are performed on vectors whose length of a vector led to an increase of the resolution time due to several factors. cases, due to the inherent functioning of the FPU, augmenting the dimension is variable and depends on the processed data. In these

1) Signatures are represented as a sequence of bidimensional points that represent the horizontal x and vertical y pen position:

$$\begin{cases} S = s_1, s_2, \ldots, s_j, \ldots, s_N; \text{ with } s_j = (s_x[j], s_y[j]) \\ T = t_1, t_2, \ldots, t_i, \ldots, t_N; \text{ with } t_i = (t_x[i], t_y[i]) \end{cases} i, j \in [1:N] \quad (1)$$

where S and T denote the captured and template signatures to be aligned, respectively. In this particular case, pro-vided that signatures have been previously preprocessed, the length of both sequences N is equal to 256.

2) From these two sequences a distance matrix $C \in \mathbb{R}^{N \times N}$ is built. This matrix represents the Euclidean distance be-tween each pair of elements of S and T according to the following expression:

$$C(t_i, s_j) = \sqrt{(t_x[i] - s_x[j])^2 + (t_y[i] - s_y[j])^2}; i, j \in [1:N].$$

The second approach proposes designing individual dedicated hardware units that minimize memory accesses and parallelize operations. In this approach the substitution of floating-point computations by fixed-point is a usual procedure, in order to simplify the data- path design and to reduce the hardware resources needed by its implementation.

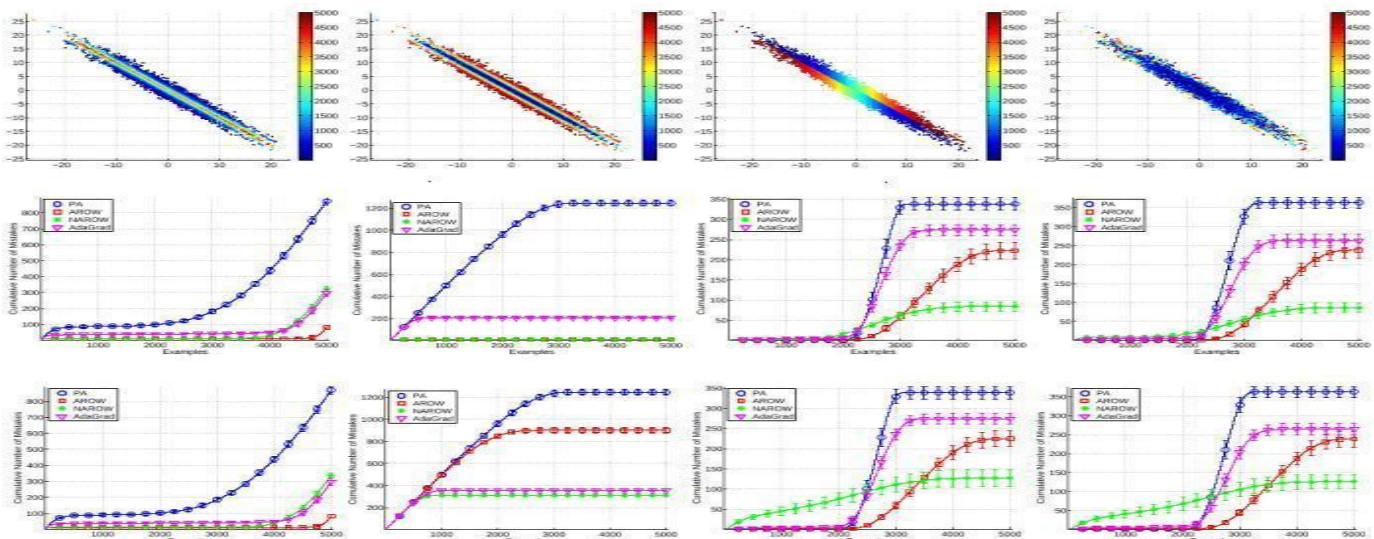
X. Previous Works

Most calculations used in biometric algorithms are solved by mathematical operations involving floating- point numbers. Essentially, the use of this type of arithmetic is necessary, because the resolution of these algorithms requires representing the intermediate and final results with a high accuracy or with a wide dynamic range. The solution adopted in previous works, when

TABLE -II
Signals Used to Calculate the Feature Vector

Feature number	Description	Mathematical Expression
C_1, T_1	Coordinate x	$x(t)$
C_2, T_2	Coordinate y	$y(t)$
C_3, T_3	Velocity in x	$v_x(t) = \text{reg}(x(t),2)$
C_4, T_4	Velocity in y	$v_y(t) = \text{reg}(y(t),2)$
C_5, T_5	Absolute velocity	$v(t) = \sqrt{v_x^2(t) + v_y^2(t)}$
C_6, T_6	Acceleration in x	$a_x(t) = \text{reg}(v_x(t),2)$
C_7, T_7	Acceleration in y	$a_y(t) = \text{reg}(v_y(t),2)$
C_8, T_8	Absolute acceleration	$a(t) = \sqrt{a_x^2(t) + a_y^2(t)}$
C_9, T_9	Tangential acceleration	$a_t(t) = \text{reg}(\lVert v(t) \rVert, 2)$
C_{10}, T_{10}	Angle α	$\alpha(t) = \arcsin\left(v_y(t) / \lVert v(t) \rVert\right)$
C_{11}, T_{11}	Cosine of α	$\cos\alpha(t) = v_x(t) / \lVert v(t) \rVert$
C_{12}, T_{12}	Sine of α	$\sin\alpha(t) = v_y(t) / \lVert v(t) \rVert$
C_{13}, T_{13}	Angle β	$\beta(t) = \text{reg}(\alpha(t),2)$
C_{14}, T_{14}	Cosine of β	$\cos\beta(t)$
C_{15}, T_{15}	Sine of β	$\sin\beta(t)$
C_{16}, T_{16}	Pressure	$p(t)$
C_{17}, T_{17}	Velocity of p	$v_p(t) = \text{reg}(p(t),2)$
C_{18}, T_{18}	Azimuth angle	$az(t)$
C_{19}, T_{19}	Inclination angle	$in(t)$
C_{20}, T_{20}	Velocity of azimuth angle	$v_{az}(t) = \text{reg}(az(t),2)$
C_{21}, T_{21}	Velocity of inclination angle	$v_{in}(t) = \text{reg}(in(t),2)$
C_{22}, T_{22}	Curvature radius	$r(t) = a_t(t) / \text{reg}(\beta(t),2)$
C_{23}, T_{23}	The length to width ratio for window of size 5, centered at the current position	$l_{w5}(t) = \dfrac{\sum_{i=t-2}^{t+1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}}{\lvert \max\{y_{t-2}..y_{t+2}\} - \min\{y_{t-2}..y_{t+2}\} \rvert}$
C_{24}, T_{24}	The length to width ratio for window of size 7, centered at the current position	$l_{w7}(t) = \dfrac{\sum_{i=t-3}^{t+2} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}}{\lvert \max\{y_{t-3}..y_{t+3}\} - \min\{y_{t-3}..y_{t+3}\} \rvert}$
C_{25}, T_{25}	Ratio of the minimum over the maximum velocity (w=5)	$\text{minmax}(t) = \left\lvert \dfrac{\min(v_{t-2}..v_{t+2})}{\max(v_{t-2}..v_{t+2})} \right\rvert$

Fig: 6) show the matching of given signature using mouse behavior



Unfortunately, this approach has also some important draw-backs.

XI.SYSTEM DEVELOPMENT

1) USER REGISTRATION:

In this module we are implementing the Client interface by which the Client can interact with the Application. To access the Application, the Client wants to the register their details with Application Server. They have to provide their information like Name, Password, Date Of birth, Mobile Number and etc. This information will stored in the database of the Application Server. The User is allowed to the access the application only by their provided Interface.

2) DATA VERIFICATION:

The Server will monitor all the User's information in their database and verify the required. Also the Server will store the entire User's information in their database. Also the Server has to establish the connection to communicate with the Users. The Server will update the each User's activities in its database. The Server will authenticate each user before they access the Application. So that the Server will prevent the Unauthorized User from accessing the Application.

3) LEARNING PHASE:

We'll train the system according to identify the User's Signature by using the following modules. (1) Mouse–Behavior Capture, (2) Feature Construction, and (3) Training / Classification. The first module serves to create a mouse-operation task, and to capture and interpret mouse-behavior data. The second module is used to extract holistic and procedural features to characterize mouse behavior and to map the raw features into distance-based features by using various distance metrics. The third module, in the training phase, applies neural network on the distance-based feature vectors to compute the predominant feature components, and then builds the user's profile using a one-class classifier. In the classification phase, it determines the user's identity using the trained classifier in the distance-based feature.

4) SIGNACTURE VERIFICATON PHASE:

Server will verify the User when they are login into their account. The Server will verify the signature provided by the User while login with the Signature provided by the User when they provided during the Training Phase. If the signature is not matched, then the Server will not allow the User to access their account.

5) OTP VERIFICATION:

Once the User provided their signature correctly, the Server will generate the Session Key using Secure Random Number generation algorithm and send it to the User Email id. Once the User received their session key in their Email id, they have to provide the first 2 digits of the session key and the server will verify the next 2 digits of the session key. Once the Session key is verified by the Server, the User is allowed to access their account.

XII. CONCULSION

Mouse dynamics is a newly emerging behavioral biometric, which offers a capability for identifying computer users on the basis of extracting and analyzing mouse click and movement features when users are interacting with a graphical user interface. Many prior studies have demonstrated that mouse dynamics has a rich potential as a biometric for user authentication. In this study, we highlighted the challenges faced by mouse-dynamics-based user authentication, and we developed a simple and efficient approach that can perform the user authentication task in a short time while maintaining high accuracy. Holistic features and procedural features are extracted from the fixed mouse-operation task to accurately characterize a user's unique behavior data. Then distance-based feature construction and parametric Eigen space transformation are applied to obtain the predominant feature components for efficiently representing the original mouse feature space. Finally, a one-class classification technique is used for performing the user authentication task.

A traditional, static password is usually only changed when necessary: either when it has expired or when the user has forgotten it and needs to reset it. Because passwords are cached on computer hard drives and stored on servers, they are susceptible to cracking. This is especially a concern for laptops since they can be easily stolen.

Once the User provided their signature correctly, the Server will generate the Session Key using Secure Random Number generation algorithm and send it to the User Email id. Once the User received their session key in their Email id, they have to provide the first „2" digits of the session key and the server will verify the next „2" digits of the session key.

Once the Session key is verified by the Server, the User is allowed to access their account. To provide the security for the Users and Sensitive applications using Mouse Dynamics Mechanism.

REFERENCES

[1] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE Trans. Syst., Man, Cybern.—Part C: Appl. Rev.*, vol. 38, no. 5, pp. 609–635, Sep. 2008.

[2] S. Impedovo and G. Pirlo, "Verification of handwritten signatures: An overview," in *Proc. 14th Int. Conf. Image Anal. Process.*, Sep. 2007, pp. 191–196.

[3] J. J. Igarza, L. Gómez, I. Hernáez, and I. Goirizelaia, *Searching for an Optimal Reference System for On-Line Signature Verification Based on (x, y) Alignment*, D. Zhang and A. K. Jain, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 519– 525, ICBA 2004, LNCS 3072.

[4] D. Impedovo and G. Pirlo, "On-line signature verification by stroke-de-pendent representation domains," in *Proc. 12th ICFHR*, Kolkata, India, Nov. 2010, pp. 623–627, 16–18.

[5] G. Pirlo, "Algorithms for Signature Verification," in *Proc. NATO-ASI Series Fund. Handwriting Recognit.*, S. Impedovo, Ed., Berlin, Ger-many, 1994, pp. 433–454, Springer-Verlag.

[6] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, and A. Salzo, "A multi expert system for dynamic signature verification," in *Proc. 1st Int. Workshop, Multiple Classifier Syst. (MCS 2000)*, J. Kittler and F. Roli, Eds., Cagliari, Italy, Jun. 2000, vol. 1857, Series: Lecture Notes Comput. Sci., pp. 320–329, Springer-Verlag Berlin Heidelberg.

[7] G. Dimauro, S. Impedovo, M. G. Lucchese, R. Modugno, and G. Pirlo, "Recent advancements in automatic signature verification," in *Proc. 9th Int. Workshop Frontier Handwriting Recognit.*, Oct. 2004, pp. 179–184, IEEE Comput. Society Press.

[8] S. Nabeshima, S. Yamamoto, K. Agusa, and T. Taguchi, "MEMO-PEN: A new input device," in *Proc. Int. Conf. Companion Human Fac-tors Comput. Syst. (CHI'95)*, 1995, pp. 256–257.