**AN AI BASED KEYSTROKE DYNAMICS FOR USER AUTHENTICATION**

**ABSTRACT:**

Authentication is frequently referred as the most critical part of a computer system security. Users commonly identify themselves using a combination of username and password, but sometimes this is not enough. Concerning web-based services, attacks like phishing or social engineering can easily result in identity theft. In addition, the widespread use of single sign-on services can seriously increase the consequences of such attacks. In these circumstances strong authentication is mandatory.

Strong authentication is often implemented using additional authentication steps or specialized hardware modules, which is not suitable for web-based systems. However, biometrics can used to overcome these limitations. More specifically, behavioural biometrics based on keyboard typing patterns can provide an extra security layer on top of conventional authentication methods, with no additional cost and no impact to the user experience.

This work aims to evaluate the feasibility of the implementation of strong authentication on the web using keystroke dynamics. This is carried out through the creation of a application prototype using python environment.

**1. INTRODUCTION**

Our dependence on computers and digital platforms has been observed to be overwhelmingly increased to simplify our lives. The use of such automated information systems together has resulted in improved performance of the available networking services in the form of reliability and computational costs. With such efficient utilization the advances in technology have generated a collective interest in global access to such online platforms. However, at the same time there is a rise in the threats regarding to the security of computers. Usage of advanced methodologies to safeguard the system from attacks and frauds come under the topmost concerning priority of many research scientists. Hence there is a need to generate foolproof measures to prevent such unauthorized access is being worked upon. One such preventive method to give access to individuals by detecting their unique and behavioral pattern is an individual’s typing rhythm. This unique typing rhythm tends to become a natural choice for security of computers and is commonly known as Keystroke Dynamics. It is observed that when a person types; the placement of his fingers, applied pressure on the keys and the regularly typed strings appears to be consistent for a specific individual. Hence this concept is used to differentiate between an intruder and a legit user as they will be typing on the keyboard anyway. Therefore, making such kind of typing rhythms easily accessible to track computer activities

**1.1 Background**

The initial step to prevent any form of unauthorized control into a system is user authentication. This is the only operation that confirms the identity of an individual. This identity is matched with a pre-generated code or a registration number with the legit individual and is used in the process of identification. Once this identification is accomplished some form of indicator is shared to give access control to the user.

Knowledge-based encryption is based on the idea of two people sharing their secrets. Username and Login credentials are two well-known examples. Whereas object-based authorization is characterized based on possession of specific things and relies on the concept of something which someone has. On the other hand, biometric based authentication is characterized by behavioral features an individual would have.

In real time, the classifications of object and knowledge based are merged to fulfil the process of authentications such bank passwords and their PINS. One major disadvantage with regards to this classification-based authentication is the ability to memorize and manage multiple such PINS and recalling them. Therefore, the usage of biometrics authentication is preferred as it overcomes these issues and makes use of automated methods to identify and verify the individual. Also, this form of authentication is gaining worldwide popularity as they provide an extra level of security.



**Figure 1.1: Classification of user authentications**

**1.2 Motivation**

User authentication has always been a fundamental component of access control. Nowadays, the Internet is the standard platform for communication and a tremendous amount of sensitive data is transmitted between computer systems. This makes the process of identity verification more important than ever. However, basic username and password combination is used the same way we did a decade ago. Attacks like man-in-the-middle, phishing or social engineering can easily result in identity theft. This can lead to access to private information, exploitation of trust relationships and other criminal activities. Furthermore, with the widespread use of single sign-on services, where a single set of credentials can authenticate users in multiple websites, consequences of identity theft can be devastating. Even though in most cases simple password-based authentication will suffice, other environments like online commerce and banking platforms can definitely benefit from strong authentication to ensure proper protection of sensitive, private and confidential data.

Strong authentication doesn’t have a standard definition, but it generally aims to deliver increased security beyond standard authentication methods. It is often implemented with two-factor authentication and usually based in knowledge or token-based approaches. Although these methods can effectively improve security in authentication, what is really validated are passwords, tokens and keys. Also, user credentials can be stolen, lost, shared or manipulated, resulting in compromised security.

Biometrics can be used to overcome the limitations imposed by classical solutions by allowing identity verification based on the user himself. Unfortunately, authentication solutions based on biometrics may not always be easily deployed, since most of them require specific hardware (e.g. fingerprint reader).

Keystroke dynamics, a behavioural biometric, presents itself as a viable contender to implement strong authentication on the web. It has the distinct advantage of not requiring specific hardware, while maintaining the desirable properties of the more mainstream biometric solutions. Also, as we will explore in this work, it can be seamlessly integrated with existing password-based authentication.

**1.3 Problem Statement**

One of the major issues or a problem the digital world face is with regards to passwords used. Although, a few organizations do exist that provide secured authentication for users when they login but on the other hand there are still most sites that skip the process of identification leading to major crimes across the globe. Hence the research work in this thesis aims to suggest better security systems that could be analysed using the typing behaviour of individuals using keystroke dynamics. Main emphasis is being put on to detect the typing behaviour of individuals using the concepts of validate the keystrokes timing features of users and give them a more secure and efficient system than ever before.

**1.4 Literature Review**

Keystroke dynamics is observed to be an emerging field of interest in terms of security that is responsible to validate the authenticity of the system based on users typing rhythm. This authentication process has been overpowering other fields of technology due to the following reasons:

• Easy implementation as only the typing data is required

• Zero hardware requirement

• Low computational cost

• Does not require special permission from the user

Various statistical models have been proposed to implement the studies of keystroke dynamics [1-5]. Classifiers such as [6-10] have been built using machine learning approaches. Apart from this hybrid models are also taken into consideration [11-13] to build the model with accuracy. These models can distinguish and establish a certain pattern of typing frequency and are also capable to access multitudes of data. The argument of keystroke dynamics is observed to differ in certain studies with regards to neuro-physiological behavior of an individual.

This behavior exhibits a user’s specific typing pattern [14]. In [15] the concept of keystroke was further classified using MLP and clustering algorithms. Alternatively, [16] designed a system that incorporates the features of applied pressure and typing latency time difference to create and organize a typing pattern for a user. To authenticate individual users, they made use of Artificial Neural Network as classifiers. With an average training time of 0.9094 seconds, the classification rate was 100%. Using the RBFN method, the same concept of pressure and time latency was established in [17]. [18] Made use of MLP with Radial Basis Function and achieved an authentication of 97%.

The contribution in [19] achieved an accuracy of 97.5%. [20] Performed an experiment with a 17-digit password, where each individual typed in the password several times in each session and features such as the size of the finger and timing frames were taken into consideration to reduce the error rate.

Similarly [21] made use of the time intervals between keystrokes utilizing the concepts of MLP and fundamentals of principal component analysis (PCA). This experimentation resulted in an accuracy of 80%. [22] Examined the dataset of keystrokes without the touchscreen features where the data was collected from Android devices and classification algorithms such as SVM, Naïve Bayes were used. This implementation resulted in a 10% percent increase than a normal authentication process.

**1.5 Objectives**

This work aims to evaluate the viability of keystroke dynamics as a method for implementing strong authentication on a web-based environment. It should result in an implementation capable of enrolment and identity verification, applying typing behaviour on top of basic username and password credentials.

The goal of this study is to compare the performance of some classical matching algorithms proposed by previous research. Although numerous experiments have achieved encouraging results, there is significant variation in the reported effectiveness, which makes it hard to assess the full potential of keystroke dynamics as a biometric for authentication.

We’ll also look at how adaptive techniques can influence the performance of keystroke dynamics. Finally, most studies on keystroke dynamics tend to use a single passphrase for all users; this work’s experiment should account for individual authentication credentials for every user, which is arguably a better representation of the real world.

**2. SYSTEM STUDY**

System analysis will be performed to determine if it is flexible to design information based on policies and plans of organization and on user requirements and to eliminate the weakness of present system. This chapter discusses the existing system, proposed system and highlights of the system requirements.

**2.1 EXISTING SYSTEM**

Distortion in keystroke dynamics is not a well researched field, and we did not find many results when doing literature search. However, it measured two typing samples of keystroke dynamics data and used two different measures to compare the samples. However, one method that could be used for detecting distorted timing information is Benford’s Law and ZIPF’s Law. Benford’s Law, or the first-digit law, is an observation in a set of numerical data where the first digit, or leading digit, is more likely to be small. In a balanced distribution of numbers between 1 and 9 there would be exactly 11% for each number to be the leading digit. However, if Benford’s Law is obeyed then the change of the leading bits to be small increases.

**DISADVANTAGES OF EXISTING SYSTEM**

* However, the results showed that only latency values from keystroke dynamics timing information followed the law.
* While duration values did not follow the law.More resources used for re-keying because it is done for each join or leave operations.

**2.2 PROPOSED SYSTEM**

In order to authenticate a user using keystroke dynamics we need to create a reference average value for each user that will represent, as accurately as possible, their specific typing behaviour. This template varies a lot depending on whether static or continuous authentication is used. In authentication we want to create a template that reflects the typing rhythm that the genuine user uses in order to type the password.

This average value is created based on enrolment samples, where the user would get requested to type their password a number of times. The features, such as duration and latency, are then extracted and the average typing rhythm is calculated and stored as a reference average. When a user tries to authenticate, the system will check their typing rhythm, which is referred to as a probe, against the reference template and then either reject or accept the user based on a criterion. This criteria for decision making are decided by a threshold which is created by a minimum and maximum average time in milliseconds.

**ADVANTAGES:**

* The proposed Keystroke dynamics is based on the assumption that each user can be authenticated because of their unique typing manner. This is because keystroke dynamics performs on a millisecond’s precision level meaning it is impossible to accurately recreate the way another user types.
* This is true even for a user who is typing their own password, as they would not be able to type exactly the same way they did last time. Even though the user might type one of the keys or key pairs the same, there are still other keys that they could type in a different way. It is because of this reason keystroke dynamics works as an authentication method.

**3. SYSTEM SPECIFICATION**

**3.1 HARDWARE SPECIFICATION**

Processor : i3 and above

RAM : 2 GB and above

HDD : 500GB and Above

**3.2 SOFTWARE SPECIFICATION**

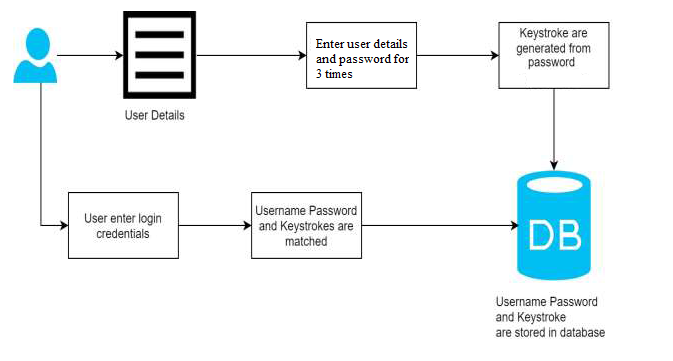
Operating System : Windows XP/8/10

Programming Language : Phyton

IDE : Sql Lite

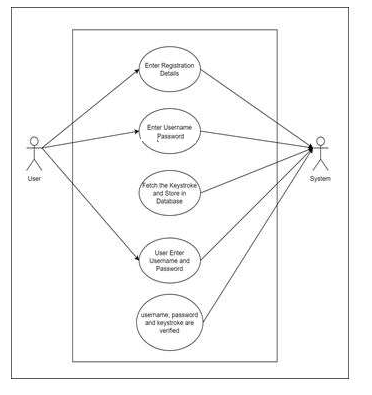
**4. SYSTEM ANALYSIS AND DESIGN**

**Architecture diagram**

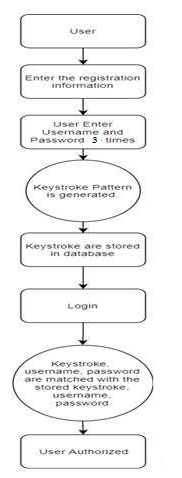
****

**Use Case Diagram:**

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system’s behavior.

****

**DFD:**

****

**4.2 MODULE DESIGN**

**Keystroke timing:**

From the KeyDown and KeyUp time of each keystroke we can calculate the duration and latency of a key. The duration of a key is how long the key was held down, this can also be referred to as dwell or hold time. While the latency of a key is the time between releasing one key and pressing another key, and this can sometimes be referred to as flight time. We differentiate between 4 different latencies, given as:

* **pp-latency:** The timing it takes to press down one key and the next key.
* **rr-latency:** The timing it takes to release one key and the next key.
* **rp-latency:** The timing it takes to release one key and press the next key.
* **pr-latency:** The timing it takes to press down one key and release the next key.

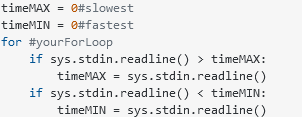
In order to get the pp-, rr- and pr-latency we have to use the timing information from duration and rp-latency.We can calculate pp-latency as *latpp* = *durA*+*latrp*, rr-latency as *latr r* = *latrp* + *durB* and pr-latency as *latpr* = *durA* + *latrp* + *durB* where *durA* and *durB* represents the duration of two different keys. The following figure shows the timing values we can extract if a user types the keys *A* and *B*.



**Figure : Timing information extracted from two keystrokes**

From these latencies, only the rr-latency and rp-latency can be negative. For example, for the rr-latency we can press the shift key, followed by pressing the C key, and then release the C key before releasing the shift key. The same can be said with rp-latency as we can press the next key before releasing the previous key, for example we can press the C key before releasing the shift key.

By using pp-latency, rp-latency and duration for our timing values. The naming of pp-latency refers to press-press-latency, however, we will refer to this as KeyDown-KeyDown latency (DD). While rplatency refers to press-release-latency, which will be called KeyUp-KeyDown latency (UD).



**Matching module:**

In this module, extracted timing features are compared against those stored in the database, using a matching algorithm from statisticalt approaches. Ultimately, this process results in a matching score, that represents a similarity measure between the extracted features and the ones previously stored in the database.

**Decision module:**

Given the matching score and a decision threshold, this module is responsible for either accepting or rejecting the claimed user identity, based on the matching score and a predefined threshold.

**4.3 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

**4.4 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**4.5 DATABASE DESIGN**

Databases are normally implemented by using a package called a Data Base Management System (DBMS). Each particular DBMS has somewhat unique characteristics, and so such, general techniques for the design of database are limited. One of the most useful methods of analyzing the data required by the system for the data dictionary has developed from research into relational database, particularly the work of E.F.Codd. This method of analyzing data is called “Normalization”. Unnormalized data are converted into normalized data by three stages. Each stage has a procedure to follow.

**NORMALIZATION:**

The first stage is normalization is to reduce the data to its first normal form, by removing repeating items showing them as separate records but including in them the key fields of the original record.

The next stage of reduction to the second normal form is to check that the record, which one is first normal form, all the items in each record are entirely dependent on the key of the record. If a data item is not dependent on the key of the record, but on the other data item, then it is removed with its key to form another record. This is done until each record contains data items, which are entirely dependent on the key of their record.

The final stage of the analysis, the reduction of third normal form involves examining each record, which one is in second normal form to see whether any items are mutually dependent. If there are any item there are removed to a separate record leaving one of the items behind in the original record and using that as the key in the newly created record.

**BUSINESS MODELING:**

The information flow among business function is modeled in a way that answers the following questions: what information drives the business process? What information is generated? What generate it? Where does the information go? Who process it?

**DATA MODELING:**

The information flow defined as a process of the business modeling is refined into a set of data objects that are needed to support the business. The characteristics (called attributes) of each object are identified and relationships between these objects are defined.

**PROCESS MODELING:**

The data objects defined in the data-modeling phase are transformed to achieve the information flow necessary to implement a business function. Processing description is created for addition, modifying, deleting, or retrieving a data object.

**5. SYSTEM TESTING**

Software testing is a critical element if software quality assurance represents the ultimate reviews of specification, design and coding. Testing is vital of the system.

Errors can be injected at any stage during development. During testing, the program is executed with correctness. A series of testing are performed for the proposed systems before the system is delivered to the user.

**5.1 Unit Testing**

In the unit testing the testing is performed on each module and this module is known as module testing. This testing was carried out during programming state itself. In this testing all the modules working satisfactorily as regard to the expected output from the module.Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. Unit tests are created by programmers or occasionally by white box testers.

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development. Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.

**5.2 Acceptance Testing**

Acceptance testing is black-box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. It is also known as functional testing, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, or factory acceptance testing.

Acceptance testing generally involves running a suite of tests on the completed system. Each individual test, known as a case, exercises a particular operating condition of the user's environment or feature of the system, and will result in a pass or fail, or Boolean, outcome. There is generally no degree of success or failure. The test environment is usually designed to be identical, or as close as possible, to the anticipated user's environment, including extremes of such. These test cases must each be accompanied by test case input data or a formal description of the operational activities (or both) to be performed—intended to thoroughly exercise the specific case—and a formal description of the expected results.

**5.2.1 Types of Acceptance Testing**

Typical types of acceptance testing include the following

**User acceptance testing**

This may include factory acceptance testing, i.e. the testing done by factory users before the factory is moved to its own site, after which site acceptance testing may be performed by the users at the site.

**Operational acceptance testing**

Also known as operational readiness testing, this refers to the checking done to a system to ensure that processes and procedures are in place to allow the system to be used and maintained.

**Contract and regulation acceptance testing**

In contract acceptance testing, a system is tested against acceptance criteria as documented in a contract, before the system is accepted. In regulation acceptance testing, a system is tested to ensure it meets governmental, legal and safety standards.

**Alpha and beta testing**

Alpha testing takes place at developers' sites, and involves testing of the operational system by internal staff, before it is released to external customers. Beta testing takes place at customers' sites, and involves testing by a group of customers who use the system at their own locations and provide feedback, before the system is released to other customers. The latter is often called “field testing”.

**5.3 Integration Testing**

One module can have adverse effect on another such functions when combined may not produce the desired results. Integration testing is a systematic technique for constructing the program structure and conducting test to uncover errors associated with interface. All the modules are combined in this testing step. The entire program is tested as the whole. The errors uncovered are corrected for the next testing step.

**5.4 Black Box Testing**

The black box approach is attesting method in which test data are delivered from the functional requirement without regard to the final program structure. Because only functionality of the software is concerned.

In black box testing, only the functionality is determined by observing the outputs to the corresponding input. In this testing various input images are exercised and the output images are compared as required by the content retriever.

**5.5 White Box Testing**

White box testing are the software predicates on close examination of procedure details. It provides test cases that exercise specific test for conditions and loops. White box testing was carried out in the order to guarantee that

* All independent parts within a module exercised at least once.
* All logical decision on this true and false side was exercised

**5.6 Validation Testing**

Computer input procedures are designed to detect errors in the data at the lower level of detail which is beyond the capability of the control procedures. The validation succeeds when the software functions in the manner that can be reasonably expected by the customer.

**6. IMPLEMENTATION**

The implementation phase focuses how the engineer attempts to develop the system. It also deals with how data are to be structured, how procedural details are to be implemented, how interfaces are characterized, how the design will be translated into programming and hoe the testing will be performed. The methods applied during the development phase will vary but three specific technical tasks should always occur.

* The software design
* Code generation
* Software testing

The system group has changed with responsibility to develop a new system to meet requirements and design and development of new information system. The source of these study facts is variety of users at all level throughout the organization.

**6.1 Stage of Development of a System**

* Feasibility assessment
* Requirement analysis
* External assessment
* Architectural design
* Detailed design
* Coding
* Debugging
* Maintenance

**6.2 Feasibility Assessment**

In Feasibility this stage problem was defined. Criteria for choosing solution were developed, proposed possible solution, estimated costs and benefits of the system and recommended the course of action to be taken.

**6.3 Requirement Analysis**

During requirement analysis high-level requirement like the capabilities of the system must provide in order to solve a problem. Function requirements, performance requirements for the hardware specified during the initial planning were elaborated and made more specific in order to characterize features and the proposed system will incorporate.

**6.4 External Design**

External design of any software development involves conceiving, planning out and specifying the externally observable characteristic of the software product. These characteristics include user displays, report formats, external data source and data links and the functional characteristics.

**6.5 Internal Design Architectural and Detailed Design**

Internal design involved conceiving, planning out and specifying the internal structure and processing details in order to record the design decisions and to be able to indicate why certain alternations were chosen in preference to others. These phases also include elaboration of the test plans and provide blue prints of implementation, testing and maintenance activities. The product of internal design is architectural structure specification.

The work products of internal design are architectural structure specification, the details of the algorithm, data structure and test plan. In architectural design the conceptual view is refined.

**6.6 Detailed Design**

Detailed design involved specifying the algorithmic details concerned with data representation, interconnections among data structures and packaging of the software product. This phase emphasizes more on semantic issues and less synthetic details.

**6.7 Coding**

This phase involves actual programming, i.e, transacting detailed design into source code using appropriate programming language.

**6.8 Debugging**

This stage was related with removing errors from programs and making them completely error free.

**6.9 Maintenance**

During this stage the systems are loaded and put into use. They also get modified accordingly to the requirements of the user. These modifications included making enhancements to system and removing problems.

**7. CONCLUSION AND FUTURE ENHANCEMENT**

With the conclusion of this work, and regarding the original research goals, keystroke dynamics can be considered a viable choice to implement strong authentication on the web. We’ve shown that this technology can be integrated on top of a tradition authentication procedure, taking advantage of user input to seamlessly capture and classify the associated typing behaviour.

Keystroke dynamics presented some notable advantages over other alternatives. No specialized hardware was needed. Biometric data acquisition was completely transparent and multifactor authentication was effectively achieved with a single conscious action from the user. In addition, even simple matching algorithms, as the ones used in this study, can yield reasonable accuracy on authentication.

Although the proposed goals of this work were successfully met, some points could have been improved. An assessment of the system usability should have been done by the users that participated in the study. Furthermore, the target population was greatly biased, mainly computer science students, which are probably better typists than the average individual. Although more complex, applying this study to a more diverse population would have been a closer representation of a real environment.

**SCOPE FOR FUTHURE ENHANCEMENT**

For instance, it would be interesting to develop a complementary study, focused on the evaluation of more sophisticated algorithms, in order to further improve matching accuracy.

**8. BIBLIOGRAPHY**

[1] J. V. Monaco and C. C. Tappert, “The partially observable hidden Markov model and its application to keystroke dynamics,” Pattern Recognition, 2018.

[2] S. Roy, U. Roy, and D. D. Sinha, “Security enhancement of knowledge-based user authentication through keystroke dynamics,” in Proc. MATEC Web Conf., 2016, vol. 57.

[3] K. S. Killourhy. (2012). A scientific understanding of keystroke dynamics. [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/2012/CMU-CS-12-100.pd f

[4] A. Messerman, T. Mustafić, S. A. Camtepe, and S. Albayrak, “Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics,” in Proc. Int. Jt. Conf. Biometrics, 2011.

[5] P. S. Teh, B. J. Andrew Teoh, T. S. Ong, and H. F. Neo, “Statistical fusion approach on keystroke dynamics,” in Proc. Int. Conf. Signal Image Technol. Internet Based Syst., January 2007, pp. 918–923.

[6] J. Ho and D. K. Kang, “Mini-batch bagging and attribute ranking for accurate user authentication in keystroke dynamics,” Pattern Recognit., vol. 70, pp. 139–151, 2017.

[7] S. Maheshwary and V. Pudi, “Mining keystroke timing pattern for user authentication,” Lect. Notes Comput. Sci., vol. 10312, pp. 213–227, 2017.

[8] A. Darabseh and A. S. Namin, “On accuracy of classification-based keystroke dynamics for continuous user authentication,” in Proc. 2015 Int. Conf. Cyberworlds, 2016, pp. 321–324.

[9] P. H. Pisani, A. C. Lorena, and A. C. P. L. F. de Carvalho, “Adaptive Positive Selection for Keystroke Dynamics,” J. Intell. Robot. Syst. Theory Appl., vol. 80, pp. 277–293, 2015.

[10] R. Giot, M. El-Abed, and C. Rosenberger, “Keystroke dynamics with low constraints SVM based passphrase enrollment,” in Proc. IEEE 3rd Int. Conf. Biometrics Theory, Appl. Syst., 2009.

[11] H. Mohabeer and S. K. M. Soyjaudah, “Application of predictive coding in neuroevolution,” Int. J. Comput. Appl., vol. 114, no. 2, pp. 41–47, 2015.

[12] J. Nisha and R. P. Kumar, “User authentication based on keystroke dynamics analysis,” Int. J. Eng. Res. Appl., vol. 4, no. 3, pp. 345–349, 2014.

[13] H. B. K. Bharadi, P. S. Shah, and A. Ambardekar, “Keystroke dynamic analysis using relative entropy & timing sequence euclidian distance,” in Proc. Int. Conf. Work. Emerg. Trends Technol., 2011, p. 220

[14] R. Joyce, G. Gupta, Identity authorization based on keystroke latencies, Commun. ACM 33 (2) (1990) 168–176.

[15] L. K. Maisuria , C. S. Ong and W. K. Lai, “ A comparison of artificial neural network and cluster analysis for typing biometrics authentication”, International Joint Conference on Neural Network, IJCNN’99, vol.5, pp 3295-3299, 1999.

[16] H. Ali, Wahyudi and Momoh J.E Salami, “Intelligent Keystroke Pressure-Based Typing Biometrics Authentication System by Combining ANN and ANFIS-Based Classifiers”, International Colloquium on Signal Processing & Its Applications (CSPA), pp198, 2009.

[17] A. Sulong, Wahyudi and M.U Siddiqi, “Intelligent Keystroke Pressure-Based Typing Biometrics Authentication System by Using Radial Basis Function Network”, International Colloquium on Signal Processing & Its Applications (CSPA), pp151, 2009.

[18] N. Capuano, M.Marsella, S.Miranda and S. Salerno, “User Authentication with Neural networks”, Univerity of Salerno Italy. http://www.capuano.biz/Papers/EANN\_99.pdf

[19] M.S. Obaidat and D.T Macchairolo, “A multilayer neural network system for computer access security”, IEEE transactions on Systems, Machine and Cybernetics, vol 24(5), 1994.

[20] R. Gaines, W. Lisowski, S. Press, N. Shapiro, Authentication by keystroke timing: some prelimary results. Rand Rep. R-2560-NSF, Rand Corporation, 1980.

[21] J. Leggett, G. Williams, Verifying identity via keystroke characteristics, Int. J. Man-Mach. Stud. 28 (1) (1988) 67–76.

[22] Trojahn, M., Arndt, F., & Ortmeier, F. (2013). Authentication with keystroke dynamics on Touchscreen Keypads—effect of different N-Graph combinations. In MOBILITY 2013, The third international conference on mobile services, resources, and users (pp. 114–119).

[23] Harun, N., Dlay, S.S., & Woo, W.L. (2010). Performance of keystroke biometrics authentication system using multilayer perceptron neural network (mlp nn). In 7th International symposium on communication systems networks and digital signal processing (CSNDSP’ 2010) (pp. 711–714). IEEE, Newcastle upon Tyne.

[24] Antal, M., Szabo, L.Z., & Laszlo, I. (2014). Keystroke dynamics on android platform. In INTER-ENG 2014, 8th international conference inerdisciplinarity in engineering (pp. 114–119). Tirgu Mures: Elsevier

**9. APPENDIX**

**SAMPLE CODE:**

**SCREEN SHOTS:**