On developing a login form using Cyber Profiling and Machine Learning Algorithms

University ID: 001134942

Contents

Abstract

The internet has been a prominent aspect of the human daily life. However, it was not always a safe place to begin with. This is due to the countless attacks attempted on the platform throughout the years to the point where there are certain times when it had cost the lives of victims as well as resources by organizations. These attacks can range from as complex as a DDos or a Virus attack to as simple as Phishing attacks. Some of these attacks tend to go for organization servers. But most of them will usually try to steal personal information and use the victim's account for malicious purposes. Thus, a countermeasure must be made. This can be in the form of a verification that uses not only the username and password, but also brings the user's typing and mouse movement behavior into account. Therefore, even with the correct username and password, if the user types or moves the mouse differently, they will be considered a fake user.

1. Introduction

In recent years, the development of the internet has been rapidly increasing as time goes on. Along with this development comes new methods of committing cybercrimes which could potentially cause significant damage to either just organizations or even towards the civilians. This can be seen during the WannaCry attack in which even medical equipment at the time were affected and nearly cost the livelihood of many patients.

Fortunately, methods of cybersecurity have been implemented from time to time, although they are considered traditional and thus have been outdated as time goes by. These methods include password inputs as well as PIN numbers. The reason that these methods have been outdated can be broken down into the fact that they can be bypassed very easily either through certain software or by knowing the user's personal information. If by software, the inputs can be bypassed through brute-force or through a keylogger that can track the user's typing. This has been an issue for a long time as organizations have been quite susceptible to said attacks which tends to occur as an inside job. It is because of this very reason that cybersecurity has been heavily researched upon in order to reduce the number of ingenuine access into an individual's account.

Cybersecurity has always been a main concern and still is in demand, especially by corporate organizations. Although the mentioned attacks have been resolved soon after it is deployed, concerns are still raised as to how it is possible to prevent the same attacks from happening again. For this reason, research has been poured into this industry to combat future potential attacks. One of these research concerns the topic of Cyber Profiling. Cyber profiling can be defined as the use of an individual's behavior on the Internet to predict their behavior or even just produce a classification on danger levels. The importance of this aspect is due to the usage of tools used by attackers. Tools such as rootkits or phishing tend to leave behind digital footprints during attacks, and it is the same digital footprints that will be used to identify the type of attack and its source too. The topic of cyber profiling has been prevalent since the 90s and since then has been heavily researched upon to produce better accuracy when identifying individuals.

In this paper, the method to create an individual's cyber-profile would be through the use of keystroke behavior, known as keystroke dynamics as well as mouse movements, known as mouse dynamics

1.1 Goal

This paper was mainly inspired from attempted phishing attacks that have occurred throughout the years. According to Khonji et al., Phishing attacks can be defined as a type of cyberattack that communicates with victims electronically to convince them to act for the attacker's benefit via social engineering. In this case, the communication can be through emails or creating a convincing URL to enter their information (Khonji, 2013) . It is true that nowadays, these attacks are reduced due to staff training regarding the attack and how to avoid it. However, there will always still be a few individuals who will still fall for it. Due to this, the paper proposes the concept of using a pattern-based approach in entering personal information. This approach will make it more difficult for the attacker to attempt logging in as they will have to replicate exactly the victim's device usage behavior, thus further reinforcing the security of the platform.

2. Relevant Studies

The following concerns the relevant topics that are used during the development of the project.

2.1 Cyber profiling and Social Engineering

To understand cyber profiling, we first understand cybercrime. Cybercrime is essentially criminal activities that involves using a computer or the Internet and can include piracy, identity theft and cyberstalking. Regardless of the skill level of the attacker, these attacks tend to leave behind digital footprints that can lead to their identity. This is where cyber profiling comes into place. Cyber profiling is a method of criminal profiling that explores cyber data to determine user activities at the time of an internet access. Said cyber data typically involves the user's internet history, e-receipts, mails etc. (Garcia, 2018). This method is still recent and can be considered to be a part of digital profiling which helps investigators establish the behavior and traits of the attacker. However, criticisms naturally arise concerning the validity of this method. One of them is that building a digital profile tends to be difficult when using separate cybercrimes as the motivation and approach involved are different, making it hard to establish a behavior pattern. Another concerns the consistency of profiling framework due to the rapid development rate of technology, leading to questionable validity of old digital profiles. (UKEssays, 2018)

Social engineering can be defined as gathering information of an individual through exploiting human weaknesses inherent in organizations. In the sense of cybersecurity, it involves manipulation of the target to disclose organizational information by making the attack seem like it is legitimate and is from the superiors. One such example of social engineering is a phishing attack which commonly happens. Social engineering can be done within 2 methods which are hunting and farming. Hunting attacks the target with minimal interaction. The interaction lasts until the intended data is achieved. This is the method most frequently used by cybercriminals and typically, organizations only encounter these once. The next method, farming, involves establishing a connection with the victim. Through manipulation, the attacker can trick the victim into revealing information for longer periods of time. This method is not often practiced because it includes the risk of the attacker being found out when interacting with the victim and thus is usually conducted in specific situations (Breda, 2017).

2.2 Biometrics

Biometrics authentication can be defined as a method that utilizes an individual's genetic traits or behavioral characteristics to combine with prior data. This newly formed data is then enrolled into a database to be associated to a specific staff (Boatwright, 2007). Biometrics can be separated into 2 categories based on how they are applied. These are authentication and identification. Authentication determines whether the individual is a genuine user or just an impostor and tends to take into account certain factors such as something the genuine user possesses or their personal knowledge; Identification is associating a user with an identity that the system will verify as either unknown or is in its database (Banerjee, 2012).

The advantages of having biometrics as a form of authentication is that it can be used to protect very sensitive information found in organizations like the bank. It also allows for another method of user verification without needing passwords or PIN numbers. Biometrics are also known to be particularly difficult to replicated as each individual has a unique feature in physical appearance that although may look similar in general, can provide a distinct difference in the eyes of the authenticator (Boatwright, 2007).

There are also disadvantages relating to this method. One of which is that the biometrics system can be compromised by internal hackers and external hackers, the former which is more likely able to conduct the activity. Socially, as mentioned previously, people are still reluctant to give up their biological information because they are uncertain of whether it will be shared among different organizations (Boatwright, 2007).

2.3 Keystroke Dynamics

Monrose defines Keystroke Dynamics as a process of analyzing a user's typing behavior on a device through monitoring the user's keyboard inputs at a very fast rate. After monitoring, the system will attempt to identify the user based on their typing rhythm. Keystroke dynamics take into account a main factor to verify a user as they start typing into mediums such as a physical keyboard (Monrose, 2000). This factor is the keyboard event that occurs during typing which are the time between 2 keypresses (Press-Press), between a keypress and a key release (Press-Release), between a key release and a next keypress (Release-Press) and between 2 successive key releases (Release-Release) (Hocquet, 2007). Most of the following papers in the Literature Review follow this principle for authentication with their own twists and improvements to further increase the system's accuracy such as POHMM, K-nearest Neighbor and Support Vector Machine. An advantage of this method is that the system is non-invasive to the user as it can obtain their typing patterns passively without their knowledge and it is also fairly cost-effective to implement (Banerjee, 2012).

2.4 Mouse Dynamics

Mouse dynamics is a method of authentication that measures and assess a user's behavior on using pointer devices as verification (Ahmed, 2007)**.**General mouse dynamics verification asks users to perform a certain sequence of mouse operations such as clicking certain objects et cetera. This method takes into account the user's behavioral features such as mouse movements, dragging and clicking, as opposed to keystroke dynamic's slightly more complex mechanism (Ahmed, 2007).

This topic was further explored by Margit Antal, in an article written by Ingrid Fadelli. According to Antal, evaluation of mouse Dynamics proved to be initially tough as they realized that research of mouse dynamics had no reproducibility due to how little it was researched about, especially since the required data for their research was raw data instead of preprocessed data. To compromise, Antal used the Balabit Dataset, a type of preprocessed Mouse data instead to evaluate the effectiveness of mouse Dynamics. In their finding, Antal saw that among the 3 proposed actions, Movements ending with a Mouse Click, Movements not ending in a mouse Click, and Drag & Drop, it was Drag & Drop that proved the most effective in detecting intruders (Fadelli, 2018)

Mouse Dynamics has been seen to be quite advantageous as it is easy to implement and does not require any specialized equipment to capture the data (Shen, 2012). One of its limitations, however, is that unlike Keystroke dynamics, Mouse Dynamics is not so often researched as frequently despite the advantages it provides. Another limitation is that mouse dynamics tend to take longer to verify. This can be said in terms of the amount of data the mouse requires in order provide accurate results. Although the authentication time is only a few mere minutes, it is more than enough for a system to be compromised on the off chance that an impostor attempts to use it (Jorgensen, 2011). Lastly, Mouse dynamics is not immune to environmental factors. This can include factors such as the mood of the user, the type of hardware, as well how powerful the machine is which affects pointer speed or certain acceleration settings. These can sometimes affect the result produced and deny entry for the genuine user (Jorgensen, 2011).

Despite limitations, it can be assumed that under ideal conditions, the implementation of more authentication software can slightly increase the security of a program. Thus, the research on mouse dynamics will still be considered into implementation to act as an extra layer of security even with said disadvantages.

3. Literature Review

From what can be understood on prior works, cyber-profiling has always been the topic of research for cybersecurity, although papers covering both mouse and keystroke dynamics have not been implemented as one before. The main focus of the features to be extracted during the cyber profiling will be keystroke dynamic and mouse dynamics. Fabian Monrose et al. has covered the background of keystroke dynamics in sufficient detail. According to Monrose, the methodology of collecting data is through the span of 11 months with 63 users. Participants were to download the experiment and the results were emailed back to the author. To analyze the data, a toolkit was developed using an xview library. The toolkit was able to analyze the data using the K-Nearest Neighbour algorithm, where an output was generated using Matlab and Gnuplot (Monrose, 2000). Identification of the user was verified through statistical decision theory, which followed the research proposed by Joyce and Gupta. Joyce and Gupta represent the classification as a Means Reference Signature, where it will verify the user by comparing test signatures and magnitude of difference between the reference signature with the user's signature. For recognition of the user, Joseph and Gupta applied a replacement method by applying classifiers such as Euclidean distance and weighted probability (Joyce, 1990).

Another methodology was proposed by Joseph Roth et al, who decided to approach the aspect through a camera and reinforcement learning. For Roth, the developed system incorporates an improvement of the "Bag of Words" algorithm, which was referred as the "Bag of Multi-Dimensional Phrases" algorithm. This algorithm is able to select more than one feature of different types by using multiple input words to determine the feature, which will allow a phrase to be learnt across both words and other domains. This means that with multiple features represented in a video frame, a codebook can be made for each individual feature. Roth had implemented this algorithm by using K-means clustering 4 times to allow the system to represent a typing video in more detail. The experiment was conducted in 2 phases, one with the individual typing a paragraph from a book (Static Text) and the other with the user typing random content into a HTML Form (Free Text). The results for both phases were analyzed. In the results, Roth uses an ROC Curve which contains the False Positive Rate (FPR) and True Positive Rate (TPR) axes to evaluate the performance of the system. The system is evaluated on

3 metrics, one of which is the verification time. This metric is important for decision making on unauthorized access with minimal delays (Roth, 2014).

Thirdly, the topic of keystroke dynamics was further explored by Ioannis Tsimperidis under the basis of gender recognition. According to Tsimperidis, the method approached for this was first obtaining the dataset for freetext by a number of individuals. The author developed a program to log the freetext of the user, which is IRecu which can be found on the internet to consider in the design of the project. For analysis of the data, the author used machine-learning models such as SVM, NB as well as Random Forest (RF) to evaluate the features extracted from the keystrokes. This paper is able to provide the potential software to be used in the project to obtain keystroke data from the analyser, which makes it possible to implement the feature and work in synergy with other extracted features (Tsimperidis, 2018).

One common algorithm that is used throughout all 3 of these papers is the Naïve Bayes method, which is a reinforcement learning method based on Bayes' Theorem. This algorithm assumes that one feature of a data is independent of another feature of the same data. The advantage of this is that the algorithm is fast and simple enough to be (Ray, 2017). This particular algorithm has many types which are the Gaussian Naïve Bayes and Multinomial Naïve Bayes (Scikit-Learn, n.d.).

Another feature to extract for cyber-profiling would be Mouse dynamics. Mouse dynamics can be defined as natural occurrences of movement patterns in mouse or pointer-based devices during interaction (plurilock, n.d.). Merylin Monaro et al. has conducted an approach that incorporates mouse dynamics and unexpected question to determine if the user is genuine. To do this, they created a program that logs movement factors of the mouse such as trajectories, and velocity of the mouse movement. According to Monaro, the "liars" tend to over-learn certain standard question to provide a convincing answer. Thus, the questions made were to be only Yes/No questions in order to limit them and catch them off-guard during answering. Besides that, some of the questions were also unexpected to further provide this effect. As a result, the model was able to provide a classification with 90% accuracy on average. (Monaro, 2017) . This is proven to be effective as the approach was done on the assumption that if the liars are not ready to answer an unexpected question, they would tend to frantically move the cursor around before deciding.

Bassam Sayed et al. decided on creating a framework that can authenticate the user in a short time, allowing it for use in scenarios with static authentication such as during logins. According to the author, the framework breaks down into 4 modules which are the gesture creation module, data acquisition and preparation module, feature extraction module and classification module. Gesture creation mainly tasks the user to freely draw a predefined set of gestures with a size of 64 data points per stroke. The gestures must be completed in 1 stroke and is repeated 20 times. Data acquisition and preparation module uses the obtained data to process and filter it, removing noises in the process. The processing is done via K-means algorithm and the data smoothing is done by using Peirce's criterion to eliminate outliers in the data. The framework was applied to 39 users and it provided a False Acceptance Rate of 8.65% with one gesture, 8.57 with 2 gestures and so on. This means that the developed framework was a success as using more gestures made the authenticator more accurate (Sayed, 2013).

Mouse dynamics has also been implemented alongside Deep Learning techniques involving recurrent Neural Networks (RNN) as well as Convolutional Neural Networks (CNN). Penny Chong et al has researched this topic by applying techniques such as Long Short-Term Memory as a method of RNN and 1-Dimensional CNN. Another model was also implemented on the assumption that it could outperform the other models, this is the 2-Dimensional CNN model. This model can capture temporal data and is said to allow transfer learning, making it able to learn from prior pre-trained models to function. Chong implemented the GoogLeNet architecture as an example of this model. For this research, the BalaBit and TWOS dataset were used. As a result of this research, it was found that the 2D-CNN model was able to outperform the other models, with it having 0.08% in False acceptance rate with weighted learning on the Balabit Dataset and 0.191% FAR without weighted learning in the TWOS dataset (Chong, 2019).

An attempt at further improving Mouse Dynamics was proposed by Nakkabi, who decided that the feature can be improved via variance reduction through extractors of separate features. In their approach, Nakkabi proposes a LAMDA Classifier as well as fuzzy logic into development. Fuzzy logic was chosen to account for human uncertainty such as changes in behavior etc. and the LAMDA Classifier was chosen as a knowledge expert for the topic was not available at the time of the research. As a result of this approach, the LAMDA Classifier was shown to outperform the existing approaches used in evaluation. Besides that, the classifier was also able to produce an FAR or 0% as well as an FRR of 0.36%, although this was produced through fixed session lengths of data acquisition. The author states that this is one main limitation of the paper as the session length is important in continuous authentication (Nakkabi, 2010).

The closest paper that relates to the object of the research can be found in a conference paper written by Patrick Bours and Christopher Johnsrud Fullu. In the paper, Bours et al. proposed a login system that utilizes the mechanisms of Mouse dynamics to authenticate its users. To gather data, the authors gave a task called “follow the maze”. The maze consists of 18 tracks and participants had to perform the task 5 times per session. During the preprocessing phase, the position data is converted to velocity data following the formula:

$$v_i^x = (x_i - x_{i-1})/(t_i - t_{i-1})$$

The authors used Moving Average (MA) as a noise reduction filter. This filter will have a window size of 5. The formula is as follows:

$$v_i' = (v_{i-2} + v_{i-1} + v_i + v_{i+1} + v_{i+2})/5.$$

Feature extraction was then implemented as the next step. This process splits the data into horizontal and vertical movements. The author further modified this process by adding a transition vector that indicates the point where horizontal tracks and vertical tracks start and end. Various distance metrics were used as evaluation such as Euclidean Distance and Manhattan Distance. The results however, proved to be somewhat ambiguous as the EER value was over 40%, which was not a very promising basis for the paper. The author suggests modifying the proposed feature extraction methods such as splitting sessions into separate tracks in order to reduce this value (Bours, 2009).

4. Design

The following will concern the initial design concept of the project. The idea of the project is to create an application that contains the written program. This application will ask for a user's login details as well as provide a task involving mouse movements, and only allows them access if the typing and mouse behavior reaches a certain confidence value separately. The main program will be stored in a .py file called "module1.py".

Based on the literature review, it can be assumed that the keystroke dynamics aspect will implement the Naïve Bayes algorithm as it seems to be able to be implemented easily and also works fast on a given small sample. The program will follow the code written by tongplw as a basis for further modifications. The code can be found on Github under the user **"tongplw/Keystroke Biometrics"** (Tongplw, 2020). Besides that, another algorithm that will be used for this would be the Manhattan Distance algorithm. These 2 algorithms will be compared to each other to evaluate which is the better algorithm to be used.

For mouse dynamics, the program will have to train based on the registered user's data instead of a given dataset. This is due to every user's movement pattern being unique to one another. The program will process this data using Fuzzy Logic and Euclidean Distance and compared in the "Results" section.

During the login phase, essentially the same functions will still be used. However, they will be called as a separate function to avoid any overlap in functionality. For example, there will be 2 functions for naïve bayes which are "naiveBayes" and "naiveBayesVeri" which is used for verification. For the user to log in, they would have to go through the same phase as the registration phase. However this time, any processed data will be stored into a list called "verivariable" and it will then be used to compare the database's registered values. If the similarity between the 2 values exceed a certain percentage, only then the user will log in successfully.

4.1 Naïve Bayes Method

The Naïve Bayes algorithm is an algorithm based on Bayes' Theorem with a twist. This twist is that rather than calculating the probability of an event based on the probability of other events, Naïve bayes will instead assume that all of the probabilities are independent to the presence of each other. This assumption is considered as class conditional independence (Vembandasamy, 2015).

Naive Bayes has been applied in many scenarios due to its ease of implementation. Despite this however, the algorithm is surprisingly powerful. An example of its application can be seen in Catal et al. who made a fault prediction tool with said algorithm (Catal, 2011).

Shenglei Chen et al. improved upon this algorithm by implementing a selective method into it. This new method involves letting the algorithm pass through the training data twice, once to form a frequency table and the other to validate the result after all squared errors are accumulated. As a result of this improvement, Selective Naïve Bayes was said to outperform algorithms with the Gain Ratio-Based Feature Weighting Method (GRFW) in the Nemenyi test, which states that 2 classifiers are significantly different if the critical value between them is more than a certain value (Chen, 2020).

A novel approach to the algorithm was proposed by Liangxiao Jiang et al. in the form of a Hidden Naïve Bayes (HNB) model. The motivation behind this was the fact that Naïve Bayes could not learn arbitrary attribute from data, and also that even though learning-restricted structures could counter this factor, only one parent could be assigned to a single attribute. The Hidden Naïve Bayes model aims to combat this by avoiding the fault produced by traditional Naïve bayes, as well as being able to consider influences from all attributes into account during implementation. As a result, the HNB model seems to have a better overall performance relative to other methods of Naïve Bayes, thus proving its reliability for future implementations. (Jiang, 2008)

An example of implementation for Naïve Bayes can be found in a paper by Jiacang Ho and Dae-Ki Kang. In their paper, the authors proposed an authentication method that uses a variant of the classic Naïve Bayes method. This is called One-Class Naïve Bayes algorithm. Contrary to classic Naïve Bayes, that assumes that each feature of an object is independent of one another, the One-Class Classifier instead joins the features into one single class label and rejects instances below a certain threshold. The basic formula of the classifier is as follows, according to the paper:

accept X if j p(vj) >τ ; and reject otherwise.

The type of data collected from the keystrokes are in time format. This is due to time's property to reach infinity and thus provide frequency estimates for each letter typed. The interval between the frequency is then calculated and discretized by converting them into letters based on the amino acid code which are "ACDEFGHIKLMNPQRSTVWY". After the mapping phase, the frequency of the attribute is calculated via the following formula into a logarithmic value:

$$L(output) = \sum_{j=1}^{N} \begin{cases} log(\frac{1}{M+20}), & \text{if } letter_j \text{ is out of the 20 categories} \\ log(P(letter_j)), & \text{if } letter_j \text{ is within the 20 categories,} \end{cases}$$

Where L is the log value. This same value is then compared to the threshold value specified by the user, where it will determine if the user is genuine or an imposter (Jiacang, 2018).

4.2 Fuzzy Logic

Fuzzy logic can be defined as logic that aims at modeling approximate modes of reasoning rather than exact. This logic is unique in the sense that it can model rational human decision in uncertain situations (Zadeh, 1988). In other words, it depends on the human's ability to act based on ambiguous knowledge domains.

Fuzzy logic functions on the basis of fuzzy inference which first maps a given input to an output, then provides a threshold where certain answers or patterns above it will be considered said output. Fuzzy inferences involves membership functions, fuzzy logic operators as well as a certain amount of if-then statements (Kalogirou, 2014).

A Fuzzy logic system was implemented as well by Khubaib Khawar et al. The system was developed for the purpose of evaluating the course performance of e-learning students. According to Khawar, the logic was developed with 3 defined linguistic words as the bounds. Values within a certain range can be considered as "good" and so on. (Khawar, 2020)

The mechanism of the logic starts with the definition of the input and output variables. In this case, the input variables are assigned with the linguistic words "Good", "Average", and "Bad". Next, the variables are assigned a specific range as well as provided a membership function. The equation for the membership function is as shown:

$$f(x,a,b,c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

Otherwise, it can be defined in another manner:

$$f(x,a,b,c) = max\left(min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$$

This process is called fuzzification, as it is allows for the output to achieve a value that is not static. Static values are typically Boolean values such as 0 and 1, which corresponds to True or False. After this process, the output is then calculated based on a certain set of rules set by the developer. This set is called the Fuzzy Inference Rules set and comprises of several IF-THEN statements. The rules are based on the formula:

$$r = m^n$$

Where **r** is the number of rules, **m** is the number of bounds which in this case, are the linguistic words, and **n** is the number of input variables to be used for the process.

The next process in Fuzzy Logic is Defuzzification. This is where the linguistic words are now assigned to the output variable. Unlike Fuzzification, where the given input produces a vague answer to be processed, Defuzzification essentially converts this answer into a more decisive output. This works due to decision-making algorithms that chooses the best value given from the fuzzy set (Masoum, 2015)

A general example of a fuzzy logic system can be the prediction of weathers. Traditional logic cannot necessarily predict that tomorrow will rain with complete confidence due to certain environmental factors and the spontaneous behavior of nature in general. In the perspective of Fuzzy Logic, it does not measure the same factor with precision, rather providing a result with a confidence value. Thus, the result is still much up to interpretation of the user.

4.3: Manhattan Distance

The Manhattan Distance, also known as Taxicab Geometry Algorithm is an algorithm that calculates the distance between two points that are measured along right angles (Black, 2019). This algorithm is fairly straightforward as it provides the distance taken to get from one point to another.

$$D(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

Manhattan distance

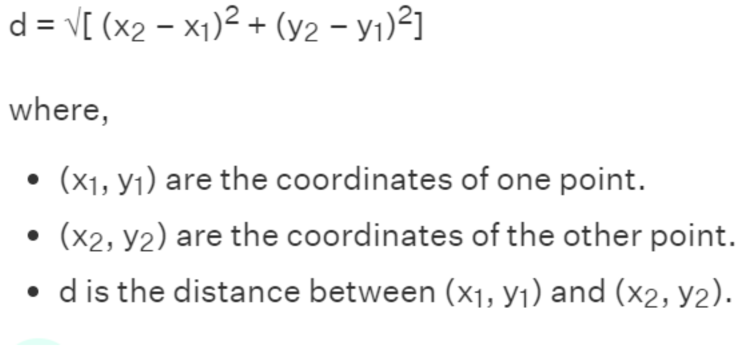
Figure 1: Formula for Manhattan Distance

The advantage of applying this algorithm is that it is very effective when discrete attributes are present in the dataset. This is because the algorithm will then be able to account for paths that are practical enough to be processed from the given values. It is also due to this advantage however, that makes Manhattan Distance not a good candidate to be used for data with high dimensions, namely data with many factors involved into a dataset. The algorithm will take up huge amounts of resources to compute and may result in errors at times. Besides that, Manhattan Distance is not optimal as well for floating point numbers in the dataset (Maheshwari, 2021).

A paper was written by Yu Zhong et al where the algorithm was used as a comparison alongside Mahalanobis Distance to point out limitations and advantages. In the paper, the author states that Manhattan Distances tend to perform better even when outliers are present in the dataset (Zhong, 2012)This is especially crucial for this metric as users attempting to log in or register, undoubtedly tend to press a wrong key or have different typing speed due to uncontrollable environmental factors. Despite the advantages of the Manhattan Distance, the author proposes to utilize its robustness as well as Mahalanobis Distance's better correlation recognition to form a new distance metric. This metric will be able to decorrelate features with Mahalanobis Distance and then normalize them before calculating the Manhattan Distance between any 2 data points. This method allows for lesser outlier influence as well as a better performance on the system. The system was evaluated using the CMU benchmark dataset for Keystroke Dynamics and naturally, managed to outperform its predecessor algorithms. The new metric managed to achieve an EER of 8.7% and has its ERR reduced by 8.4% (Zhong, 2012).

4.4: Euclidean Distance

Euclidean Distance can be defined as the shortest distance between two points. This algorithm calculates the distance in a straight line unlike Manhattan Distance which calculates the distance along right angles.



$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

where,

- (x_1, y_1) are the coordinates of one point.
- (x_2, y_2) are the coordinates of the other point.
- d is the distance between (x_1, y_1) and (x_2, y_2).

Figure 2: Formula for Euclidean Distance

Upon inspection of the formula, Euclidean Distance follows the same behavior as the Pythagoras' Theorem. One advantage of this algorithm is that it performs better in situations with compact or isolated clusters. Besides that, due to its simplistic nature, Euclidean distance is the more favored algorithm compared to Manhattan Distance when it comes to low-dimensional data, the use of K-Nearest Neighbor also facilitates the robustness of this algorithm further when integrated together. (Grootendorst, 2021)

Depending on the units to be used for computation, Euclidean distance will not be suitable as the final value may be skewed. The dataset would also need to be normalized before implementing it into this algorithm (Grootendorst, 2021). Besides that, the data vectors must have a common attribute value, otherwise, the algorithm would produce a smaller distance value (Shirkhorshidi, 2015)

Despite the disadvantages that may come with its implementation, this algorithm is still one of the most commonly used metrics as there are ways to work around the cons. Euclidean distance works best when low-dimension data is involved and the magnitude is significant enough to be measured. An example of its application would be from Jing Li et al. who decided on Euclidean distances when developing a pattern recognition system. The approach done was by implementing an adaptive image Euclidean distance algorithm that would factor the space between pixels with different gray scale values and would be embedded in existing image recognition systems based on the same algorithm. The motivation behind this paper was due to the Riemannian Geometry mentioned inside the paper that says images like a face would have a specific manifold in image space. This is because the pixels in the image would have a closer relationship if on the same object compared to pixels on different object despite the former being further apart from each other than the latter. Due to this, the author proposes an adaptive algorithm that considers either the distance of the pixels (AIMED-D) or the gray scale value between any 2 adjacent pixels (AIMED-C) depending on certain circumstances. For evaluation, the traditional Euclidean distance, algorithms IMED, AIMED-D and AIMED-C were embedded in 2 image metrics which are the nearest-neighbor classifier and the support vector machines. The results for the AIMED algorithms were successful as among the 4 algorithms, AIMED-C has the best performance in both metrics, with AIMED-D being the second best, followed by IMED and finally traditional Euclidean Distance. With Nearest-neighbor, AIMED-D increases performance by 0.95% whereas AIMED-C increases it by 1.37% ; with SVM, the former has an increased performance of 0.72% and the latter with an increase in 0.98% (Li, 2009)

4.5:MySQL

MySQL is a database management system based on the Structured Query Language (SQL). This application follows a relational model which allows it to be able to store data points that are related to one another. Relational-type databases are usually good in maintaining data consistency, this is because when one change is made, the database would ensure that every part of the table can commit to the change before it does the update. This type of commitment to change follows 4 rules that define it which are Atomicity, Consistency, Isolation, and Durability. Atomicity defines the elements that makes up a database. Consistency comprises of rules to maintain data points in their correct state after any changes. Isolation keeps the change hidden until it is available to commit, and Durability ensures that the change becomes permanent after commitment (Oracle, n.d.).

For this paper, a relational database is used as there are indeed parts of data which are derived from other existing data in the .CSV file. In the case of Mouse Dynamics, it is the Distance and Velocity that are related to the positions and time of the mouse movement, and so on. Other relational database systems exist as well, such as SQLite. However, SQLite is not so favorable due to its properties as an embedded relational database. This means that any data to be stored is kept locally in a file that can only be accessed by the application that kept it. This means that it would not be suitable for scaling in the future. Besides that, this property of SQLite means that it can only handle one connection at ta time, unlike MySQL. Thus, MySQL is chosen as the ideal database for the paper as it can be scaled for any future work, and It can also handle multiple connections at once, allowing it to be accessed by many other applications if needed. (Wolfe, 2021)

Modules used

Module name	Version	Used for
tkinter	-	GUI
pandas	1.3.4	Keylogger, Mouselogger
datetime	-	Keylogger
pynput	1.7.6	Keylogger, Mouselogger
math	-	Mouselogger
pyautogui	0.9.53	Mouselogger
time	-	Mouselogger
numpy	1.21.2	Mouselogger
threading	-	Mouselogger
sklearn	0.0	Naïve Bayes
skfuzzy	0.4.2	Fuzzy Logic
scipy	1.7.3	Manhattan Distance
pyodbc	4.0.32	Database(MySQL)

5. Implementation

The approach to this project would first be the development of the keylogger. This keylogger was taken from GitHub by the user Tongplw. This keylogger does not meet the requirement for the project however, as it only logs the key presses and releases and also the epoch at the start of the press with a label for the events respectively. The code has thus been modified to log not only the prior factors, but also the duration between the release and press of a key. The time of the keystrokes have been changed from using epoch to using milliseconds to be more easily understood.

Next implementation would be the mouse logger. As there are limited research about mouse dynamics, there are not many source codes nor repositories to modify from. Thus, the code for this aspect was made from scratch. The mouse logger is meant to provide a data sheet containing the x-coordinate, y-coordinate as well as the velocity of the mouse at that time. A Login UI will be made to facilitate these activities.

Finally, the last implementation would be the database. Firstly, a database should be created under the name "**userinfo"** as the program will be unable to function if the database does not exists. Next, it is best to change the connection type in the program to one that the user will be implementing on their own as it is currently only connected to a local server named "**DESKTOP-6VAEH1I**". Creating the database can be done in a Database Management System such as Microsoft SQL Server Management Studio.

5.1: Data Acquisition

There are 4 types of data to be acquired for this project which are Username, Password, Keystroke data and Mouse movement data.

Username and password will be acquired by the user typing in their credentials. These are constant in the database and will restrict access if the input does not match the database during login. This type of acquisition is called "static text entry". This type of entry typically requires the user to type a pre-determined text, to which the program will then compare to the enrollment data. Examples of static text entry can be usernames and passwords (Crawford, 2010). Contrary to dynamic text, which simulates real-time entry more due to its flexibility in allowing space for errors to happen and still able to process properly, Static text is the more desired type of entry to be used as this will allow the program to narrow the threshold for being a genuine user more, thus further distinguishing a fake user and a genuine user (Hu, 2008).

In the case of keystroke data, the UI will provide a paragraph in which the user will have to type out in the textbox. The "on_release" function is called as the user starts to type. During this process, any input made is appended to the data frame, "df", where it will then be saved into a .csv file with the "savetocsv" function. The factors saved into the file would be the down-time of keypress(**timeP**), up-time of the keypress**(timeR)**, key pressed(**key**), whether it is pressed or released (**event**), and the down-up time in between keypresses(**duration**).

Keystroke data will follow the dynamic text entry which allows the user to type any text freely as long as the program is able to process the inputs. Although the user is given the freedom to type anything into the textbox, it is still highly suggested that they follow the given text as a guideline to gauge how much input is needed for processing. The text to follow will be based on parts of the first paragraph of the novel "If on a winter's night, a traveler" by Italo Calvino. This text is used because it provides enough input for the keystroke algorithms to train and start processing the value. In the login phase however, another text will be used which will contain numbers and punctuation. This text may be smaller but still provides more than enough data to classify the user thanks to the presence of strings that are not entirely alphabets.

For mouse dynamics, a UI will be provided in which 5 buttons are placed randomly. Each of these buttons provide a different function, with “Button 1” starting the mouse-logger, “Button 2” to stop it, “Button 3” to get the average velocity and distance in the log, “Button 4” to start processing the data and finally, “Button 5” to save all the given inputs from usernames and passwords up to the Euclidean Distance of the movement into a database. As the time taken to press button 1 to button 2 will undoubtedly be short, the mouse logger will be designed to take as much data as possible during that short window. This also means that the 2 buttons will have to be placed as far apart as possible as well to prolong the acquisition.

5.2: Data Processing

Once all the data is acquired, the program will then start processing the values using Naïve Bayes and Fuzzy Logic for Keystroke dynamics and Mouse Dynamics respectfully. During this, certain features will be extracted from the .csv file to be processed by the algorithm. These features should have significance in providing an accuracy value as they will form the basis of the classification when a user logs in.

In the case of Keystroke Dynamics, the features to be extracted will be the “Time” and “Duration” features. These factors are chosen as they represent the Down-time and Down-Up time of the keypresses. Because different individuals type at different speed, these 2 factors will be the determinant for the classifier. The algorithm will use some rows for training and the rest will be for real-time. The algorithm uses 78% of the data for the training phase, and the rest will be used for the testing phase. The model of Naïve Bayes algorithm used for this implementation would be the Gaussian NB model. Gaussian NB essentially is a version of Naïve Bayes that still uses the normal distribution method of the standard NB model, with the difference that it can support continuous data, meaning that it is very suitable for Keystroke dynamics.

For Manhattan Distance, the same .csv file will be used. The features to be extracted from the file for this algorithm would be the **timeP** and **timeR** columns. This is because the algorithm can only function properly when it is provided a 2-dimensional array. This means that only using the **duration** column will not work, hence the features mentioned. The data processing is split into 2 phases, “timepress” and “timerelease”, each only doing one of the columns respectively. Both of these phases involve stripping the column of any empty cells, then appending them to a list. For timepress, it will be appended to timePress, and timerelease will be timeRelease. After those phases, the lists will be called into the algorithm where they will return the raw Manhattan Distance Value.

For mouse dynamics, the features to be extracted are the average Velocity and average Distance. These factors will be used in the Fuzzy Logic Algorithm in which a result value will be provided as the threshold required to exceed to log in. Fuzzy Logic systems typically contains 3 boundaries which are basically “Good”, “Neutral”, and “Poor”. For this particular system, the algorithm will only contain 2 boundaries, “Good” and “Poor” as a Neutral boundary does not pose any significance in this classification system. The “Poor” boundary will be set from 0 to 1300 units with a peak at 700 units; whereas the “Good” boundary will be set from 1300 units to 2000units with a peak at 1600 units. Upon setting the rules, the algorithm will then read the .csv file for the mouse logger and compute the resultant value.

In Euclidean Distance, the features to be extracted are the **x** and **y** coordinates in the .csv file. The program will first strip these columns of any empty cells. After that, a loop will be called where it will process the Euclidean Distance of the first 2 rows together before appending them to a list and move on to the next 2 rows. This will continue until the last non-empty cell, in which case it will then give an average value of the Distance based on the list and return it to the program.

Besides the algorithm, the program has been tested for the following situations

Use-case	Result
Logging in with correct username but wrong password and vice versa	Works as expected. User is restricted from access
During login phase, not typing in the following text correctly	User is restricted access as processed input value does not match with registered value to a certain extent.
Authenticating user with mouse behavior	Program is at a consistent value even with different movement patterns. Possibly not reliable.
During authentication, text is typed correctly and mouse movement matches	Program allows the user access to the statistics of their movement pattern, displaying processed values given by the different algorithms.

Results

The result will be split into 2 parts due to the nature of the evaluation for Mouse Dynamics. Thus, 2 usernames, Trackpad and Mouse will be used throughout this section. The evaluation will be based on an average value in 6 iterations and compared to the value taken during the Registration Phase (Registered Value).

In Trackpad, the Registered value for all the features are as Follows:

Feature	Registered Value
Naïve Bayes	0.522935779816514
Manhattan distance	3966.175
Fuzzy Logic **(Trackpad)**	1666.80128632955
Euclidean Distance **(Trackpad)**	6.99788175513195
Fuzzy Logic **(Mouse)**	1666.68071599009
Euclidean Distance **(Mouse)**	16.9895867741198

Based on these values, the evaluation starts during the login phase. The average value is taken over 6 iterations, and then compared to the Registered Value to see how close they are together.

For naïve Bayes, if the similarity value is above 100%, the boundary will be set to 12.5% over it. This is based on the fact that throughout the test with the same user, the extra value always seems to average out into that border. The similarity values of the Naïve Bayes algorithm are as follows:

Iteration (NB)	Similarity Value(%)	Processed value
1	119.517543859649123	19.517543859649123
2	117.078410311493016	17.078410311493016
3	115.53362573099415	15.53362573099415
4	91.63011695906434	91.63011695906434
5	95.6140350877193	95.6140350877193
6	91.63011695906434	91.63011695906434
Average	-	54.89888813557027

Based on the average processed value, the closeness between it and the registered value is 55%, which proves to be somewhat acceptable, but leaves room to improve.

In the case of Manhattan Distance, the features “down-time” (timeP) and “up-time” (timeR) were used. The values for this algorithm is as follows:

Iteration (MD)	Similarity Value(%)	Processed Value
1	72.91965180558094	72.91965180558094
2	16.5717599445035204	16.5717599445035204
3	40.59200615202303	40.59200615202303
4	66.86301033111246	66.86301033111246
5	39.93530290519201	39.93530290519201
6	41.47623339867755	41.47623339867755
Average	-	46.39299408951492

Based on the given average value, the Manhattan Distance gives a value of 46.4%, which means that Naïve Bayes proves to be better at keystroke Dynamics classification compared to Manhattan distance.

Mouse dynamics would work differently as mentioned before. Thus, there will be 2 tables for each algorithm, one for trackpad, another for the mouse device as different devices will provide different values. The trackpad used is on a Dell Inspiron 5480 Laptop and the mouse used will be a Microsoft Wireless Mobile Mouse 1850.

For Fuzzy Logic, the registered value for the trackpad falls on the "good" threshold, even proving to be really close to the peak. For evaluation of this algorithm, we will see if the provided value falls under the "Good" threshold, as well as checking how close it is towards the Peak Value, 1500 units. The iteration values for the algorithm are as follows:

Iteration **(Trackpad)**	Value
1	1667.0011178942823
2	1667.0111286870256
3	1667.0414119998181
4	1667.0217157366374
5	1667.0034362299073
6	1667.057794997027
Average Value	1667.0227675907827

Figure 3a: Fuzzy Logic Values by Trackpad

Iteration **(Mouse)**	Value
1	1667.1865494566412
2	1666.939291857596
3	1666.9007926314646
4	1666.9452171918356
5	1667.0528234348392
6	1667.054570163088
Average Value	1667.0132074559108

Figure 3b: Fuzzy Logic Values by Mouse

As can be seen above, the values are always consistent at 1667, in the "Good" Threshold. Especially since the value is considered quite close to the peak value, the algorithm can be considered to be feasible enough for implementation for future works.

Even with using the Mouse device, it seems that the average is not so different from when using the trackpad. Although it is now slightly closer to the peak value, the difference can be considered negligible in this case.

In Euclidean distance, the features of the mouse to be extracted would be the coordinates of the mouse, namely the x and y coordinates. Similar to keystroke dynamics, a Percentage of Similarity to the Registered Value will be used as an accuracy metric. The iteration value is as follows:

Iteration**(Trackpad)**	Similarity Value(%)	Processed Value
1	138.64557691264926	38.64557691264926
2	12.4162901613882	12.4162901613882
3	8.067571411469308	8.067571411469308
4	103.06368101213698	3.06368101213698
5	93.19078087466006	93.19078087466006
6	126.92596473754531	26.92596473754531
Average Value	-	30.38497751830819

Figure 4a: Euclidean Distance Value by Trackpad

Iteration**(Mouse)**	Similarity Value(%)	Processed Value
1	95.66651037949156	95.66651037949156
2	62.35822818974592	62.35822818974592
3	83.49250452257725	83.49250452257725
4	53.855817447455365	53.855817447455365
5	79.79913528564916	79.79913528564916
6	74.15763441647762	74.15763441647762
Average Value	-	74.88830504023282

Figure 4b: Euclidean Distance Value by Mouse

Based on the above table, it seems that a trackpad is not suitable to be used for classification due to its low average similarity value of 30%, thus proving the device's unreliability. However, Euclidean Distance seems to be highly suitable if the user is using a Mouse pointing device instead. This is because the similarity value is increased drastically to approximately 74% when using the device. This means that the algorithm is indeed capable enough for classification, but it may have to depend on the devices being used at the time. In summary, using Fuzzy Logic still proves to be better due to its flexibility in being able to provide a consistent enough classification for both devices, but it will have to compromise slightly on security due to the same flexibility as the fake user can still be able to bypass the program if they have the same manner of movement as the genuine user.

Limitations and future work

Undoubtedly, this program will have certain limitations. One of them is that the program can only work as an application itself and that it could not be implemented into websites just yet. Besides that, some of the algorithms were evaluated based on how similar the login values are to the registered values, thus raising concerns to its reliability for accuracy. Also, certain algorithms such as the Naïve bayes algorithm have the potential to become an authentication method, however modifications must be applied to increase its accuracy value. Future works will include implementing more complex algorithms to give a more proper accuracy with the same basis for registered values and also writing the program to be scalable enough to be implemented onto websites by using frameworks such as Django or Flask.

Conclusion

In conclusion, we have explored the potential of using cyber-profiling to verify users in a login platform using the 2 main factors which are Keystroke Dynamics and Mouse Dynamics. We have also explored 2 types of algorithms for each of these factors to authenticate the user. Naïve Bayes and Manhattan Distance for Keystroke Dynamics; Fuzzy Logic and Euclidean Distance for Mouse Dynamics, namely their mechanisms, reliability to be used as well as in what scenario are they practical enough to be used by providing a similarity value based on their registered data in the database.

Self-Evaluation

Throughout the course of this project, I have undoubtedly came across certain obstacles. Some more major than the other such as designing the keylogger and mouselogger. However, I have managed to overcome these with the proper research as well as determination. Although simple, I nevertheless have learnt how to properly search for a solution to problems as well as having gained the skills necessary to design a program.

Acknowledgement

I would like to thank my supervisor Mr Manos Panaousis, who provided the motivation and guides necessary to conduct the project.

References

Ahmed, A. a. T. I., 2007. A new biometric technology based on mouse dynamics. *IEEE Transactions on dependable and secure computing,,* 4(3), pp. 165-179.

Banerjee, S. a. W. D., 2012. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research,* 7(1), pp. 116-139.

Black, P. E., 2019. *Manhattan distance.* [Online]
Available at: https://www.nist.gov/dads/HTML/manhattanDistance.html
[Accessed 1 April 2022].

Boatwright, M. a. L. X., 2007. What do we know about biometrics authentication?. *Proceedings of the 4th annual conference on Information security curriculum development ,* pp. 1-5.

Bours, P. a. F. C., 2009. *A login system using mouse dynamics..* s.l., IEEE.

Breda, F. B. H. a. M. T., 2017. Social engineering and cyber security. *International Technology, Education and Development Conference,* 3(3), pp. 106-108.

Catal, C. S. U. a. D. B., 2011. Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm. *Expert Systems with Applications,,* 38(3), pp. 2347-2353.

Chen, S. W. G. L. L. a. M. X., 2020. A novel selective naïve Bayes algorithm. *Knowledge-Based Systems,* Volume 192, p. 105361.

Chong, P. E. Y. a. B. A., 2019. User authentication based on mouse dynamics using deep neural networks: A comprehensive study. *EEE Transactions on Information Forensics and Security,* Volume 15, pp. 1086-1101.

Crawford, H., 2010. Keystroke dynamics: Characteristics and opportunities. *2010 Eighth International Conference on Privacy, Security and Trust,* pp. 205-212.

Fadelli, I., 2018. *An evaluation of mouse dynamics for intrusion detection.* [Online]
Available at: https://techxplore.com/news/2018-10-mouse-dynamics-intrusion.html
[Accessed 21 February 2022].

Garcia, N., 2018. *The use of criminal profiling in cybercrime investigations.* s.l.:s.n.

Grootendorst, M., 2021. *9 Distance Measures in Data Science.* [Online]
Available at: https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa
[Accessed 28 March 2022].

Grootendorst, M., 2021. *9 Distance Measures in Data Science.* [Online]
Available at: https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa
[Accessed 12 January 2022].

Hocquet, S. R. J. a. C. H., 2007. *User classification for keystroke dynamics authentication.* Berlin, Springer.

Hu, J., 2008. A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. *2008 IEEE International Conference on Communications,* pp. 1556-1560.

Jiacang, H., 2018. One-class naïve Bayes with duration feature ranking for accurate user authentication using keystroke dynamics. *Applied Intelligence,* 48(6), pp. 1547-1564.

Jiang, L., 2008. A novel Bayes model: Hidden naive Bayes. *IEEE Transactions on knowledge and data engineering,* 21(10), pp. 1361-1371.

Jorgensen, Z. a. Y. T., 2011. *On mouse dynamics as a behavioral biometric for authentication.* s.l., ACM, pp. 476-482.

Joyce, R. a. G. G., 1990. Identity authentication based on keystroke latencies. *Communications of the ACM,* 33(2), pp. 168-176.

Kalogirou, S. A., 2014. Designing and Modeling Solar Energy Systems. *Solar Energy Engineering (Second Edition),* pp. 583-699.

Khawar, K., 2020. Fuzzy Logic-based Expert System for Assessing Programming Course Performance of E-learning Students. *Journal of Information Communication Technologies and Robotic Applications,* pp. 54-64.

Khonji, M., 2013. Phishing Detection: A Literature Survey. *IEEE Communications Surveys & Tutorials,* 15(4), pp. 2091-2121.

Li, J., 2009. An adaptive image Euclidean distance. Pattern Recognition. *Pattern Recognition,* 42(3), pp. 349-357.

Maheshwari, H., 2021. *How to decide the perfect distance metric for your machine learning model.* [Online]
Available at: https://towardsdatascience.com/how-to-decide-the-perfect-distance-metric-for-your-machine-learning-model-2fa6e5810f11
[Accessed 1 April 2022].

Masoum, M., 2015. Optimal placement and sizing of shunt capacitor banks in the presence of harmonics. *Power Quality in Power Systems and Electrical Machines,* pp. 887-959.

Monaro, M. G. L. a. S. G., 2017. The detection of faked identity using unexpected questions and mouse dynamics. *PloS one,* 12(5), p. e0177851..

Monrose, F. a. R. A., 2000. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems,* 16(4), pp. 351-359.

Nakkabi, Y. T. I. a. A. A., 2010. Improving mouse dynamics biometric performance using variance reduction via extractors with separate features.. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* 40(6), pp. 1345-1353.

Oracle, n.d. *What is a Relational Database (RDBMS)?.* [Online]
Available at: https://www.oracle.com/uk/database/what-is-a-relational-database/
[Accessed 2 April 2022].

plurilock, n.d. *Mouse Dynamics.* [Online]
Available at: https://plurilock.com/answers/mouse-dynamics-what-does-mouse-dynamics-mean/
[Accessed 15 December 2021].

Ray, S., 2017. *6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
[Accessed 10 December 2021].

Roth, J. L. X. a. M. D., 2014. On continuous user authentication via typing behavior. *IEEE Transactions on Image Processing,* 23(10), pp. 4611-4624.

Sayed, B. T. I. W. I. a. O. M., 2013. Biometric authentication using mouse gesture dynamics. *IEEE systems journal,* 7(2), pp. 262-274.

Scikit-Learn, n.d. *Naive Bayes.* [Online]
Available at: https://scikit-learn.org/stable/modules/naive_bayes.html
[Accessed 12 December 2021].

Shen, C. C. Z. G. X. D. Y. a. M. R., 2012. User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security,* 8(1), pp. 16-30.

Shirkhorshidi, A., 2015. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one,* 10(12), p. e0144059.

Tongplw, 2020. *Keystroke Biometrics.* [Online]
Available at: https://github.com/tongplw/keystroke-biometric
[Accessed 30 November 2021].

Tsimperidis, I. A. A. a. K. A., 2018. Keystroke dynamics features for gender recognition. *Digital Investigation,* Volume 24, pp. 4-10.

UKEssays, 2018. *The Art of Cybercriminal Profiling..* [Online]
Available at: https://www.ukessays.com/essays/criminology/the-art-cybercriminal-profiling-7972.php?vref=1
[Accessed 12 February 2022].

Vembandasamy, K. S. R. a. D. E., 2015. Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology,,* 2(9), pp. 441-444.

Wolfe, M., 2021. *MySQL vs. SQLite.* [Online]
Available at: https://towardsdatascience.com/mysql-vs-sqlite-ba40997d88c5
[Accessed 2 April 2022].

Zadeh, L. A., 1988. Fuzzy logic. *Computer,* 21(4), pp. 83-93.

Zhong, Y., 2012. Keystroke Dynamics for User Authentication. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 117-123.