**Project Report for Detecting Parkinson’s Disease using Keystroke Data**

1. **Introduction:**

Parkinson’s disease is a serious and cruel problem for the world. The advantage of improving lifespans is not realized by its victims, who slowly worsen in physical health and become increasingly dependent on family and friends, while caretakers often give up their independence to take care of someone who no longer retains autonomy over themselves.

Parkinson’s disease is thought to occur due to the impairment or death of neurons. The death of these neurons results in less production of dopamine, which manifests as the movement problems in Parkinson’s disease. Additionally, patients afflicted with Parkinson’s lose nerve endings that produce norepinephrine, which controls many vital body functions and this is thought to explain the non-motor functions of Parkinson’s disease such as fatigue and high blood pressure.[[1]](#footnote-1)

This disease has only become more common, with more than 10 million people worldwide living with Parkinson’s disease.[[2]](#footnote-2)

Worse still, because the symptoms of the disease are usually benign (worsening fine motor skills may be attributed to aging, for instance), it is not easy to detect Alzheimer’s disease. Detecting Parkinson’s disease would not only provide relief to millions of families who have family members afflicted by the disease, but it would also help mitigate the costs in caring for afflicted patients (which has a total economic burden of $51.9 billion dollars as of 2017[[3]](#footnote-3)) and may potentially help in recognizing factors that affect Parkinson’s disease.

1. **Objectives and Implications**

This paper, therefore, aims to detect Parkinson’s disease using machine learning processes. The project will have been successful if:

1) I can obtain keystroke data easily, securely, and quickly

2) I can extract useful information regarding the keystrokes such as the duration of time between keystrokes or the time holding a key.

3) I can train a machine learning algorithm to recognize a pattern between these keystroke data and instances of Parkinson’s Disease

My machine learning algorithm should have an accuracy of at least 70% and a high sensitivity (the ability to correctly identify people with the disease) and specificity (the ability to correctly identify people without the disease). This target was chosen as it is as accurate as professional tests for Parkinson’s disease and thus would make it viable in the clinical environment.

The second aim of this paper is to discuss the viability of using Machine Learning processes as a clinical tool in diagnosing diseases. As mentioned below under Section 4, professional testing of Parkinson’s disease is subject to many errors in the form of bias and in the difficulty of diagnosing the disease entirely as early symptoms closely resemble other diseases.

This paper aims to prove the efficacy of using machine learning in diagnosing these diseases. I believe that such an approach is viable and should be seriously considered due to it being not subject to human biases and able to detect and compute much more information than humans.

This project therefore has many important direct and indirect implications for the real world. It would firstly aim to demonstrate an accurate method of testing for Parkinson’s disease and other similar diseases, making it viable for use by clinicians and secondly, it would demonstrate the efficacy of machine learning solutions to medical problems and thus create more accurate, powerful, and non-intrusive tests for diseases.

1. **Previous instances of detecting Parkinson’s Disease**

Previous instances of detecting Parkinson’s disease using machine learning processes rely on using biomarkers such as Rapid Eye Movement (REM) or olfactory loss[[4]](#footnote-4). Such methods are effective, but rely on having these biomarkers in the database, which may not be readily available and may take valuable time in developing.

1. **Proposed Approach: Advantages and Limitations**

This paper proposes a novel approach using keystroke data to detect the onset of Parkinson’s disease.

This approach was considered after recognising that, since Parkinson’s Disease involves significant motor dysfunction, a person afflicted with Parkinson’s disease would possess different attributes in the manner of their keystrokes compared to a healthy person, for instance they may take longer to key in a sentence, or they may hold a key for longer. As a result, it may be possible to recognise Parkinson’s patients simply through an analysis of their keystroke data,

I specifically chose keystroke data because, in our digital world, one would be hard-pressed to find someone who does not possess a communications device through which we can obtain this data, making it very plentiful.

Finally, this approach is also non-intrusive and can be done remotely, which enables the patient to carry on with their daily activities without obstruction. I had noticed how many patients were unwilling to go to doctors because it would take too much time for them to meet one or they would be subject to uncomfortable tests. This approach would enable doctors to test for Parkinson’s disease without running intrusive and uncomfortable tests, thus increasing the screening for Parkinson’s disease.

A limitation to this approach would be identifying which data is relevant and the loss of other biomarkers such as bradykinesia (slowed movement) and speech changes that could be used to make this test more rigorous and specific to Parkinson’s Disease rather than any motor dysfunction. While this was considered and mitigated through extracting data which is highly pertinent to only Parkinson’s disease such as hold time, this could not be fully eliminated and resulted in an accurate, but not necessarily specific test for Parkinson’s disease. This would be considered in the conclusion.

Another challenge would be found with keylogging. Such an act may result in the sharing of compromised information, which should be avoided. This could be accomplished through anonymizing not only each patient but controlling their keystrokes in a constrained environment in the clinic. If remote use is required, then the data would first be randomized by sending it through a randomly generated hash map and then the contents of each key would be scrambled. For instance, the phrase “Parkinson’s disease” may be randomly assigned to “Ldieyn3yeyd fryi3Wn” in one attempt and “fGYEiU#JUNT IenNwjs” in another. This approach would enable the doctor to still obtain relevant information such as time between keystrokes without any of the security concerns associated with keylogging.

Such an approach would be easily extended towards other neurodegenerative or movement-based diseases and, hence, would be useful beyond Parkinson’s disease, for instance in detecting encephalitis lethargica.

Finally, this method is non-intrusive and thus is easier to implement in a real-world setting. This approach is also likely to be superior to just medical testing by professionals which is subject to bias and human cognitive limitations and do not even have a rigorous test for Parkinson’s disease[[5]](#footnote-5) and frequently misdiagnose it with an error rate of 24%[[6]](#footnote-6).

Additionally, the use of classification models such as Support Vector Machines or random forest classifier provide the opportunity for hyper-parameter tuning, which would enable the classification to become even more accurate. Finally, the use of keystroke data is important because of its prevalence and due to its assurance of being an accurate tool to detect the disease [7,8,9,10].

This paper will make use of keystroke data from the Tappy Keystroke Data[[7]](#footnote-7)11. This data is robust and well-rounded, which makes it a highly reliable source for applying machine learning methods; however, it only has 200 patients.

This may pose a problem because the low amount of individual specimens may result in overfitting the model, i.e. the model only works very well with this specific set of data and no other data. I would aim to rectify this problem by performing feature reduction: a technique used to both reduce computational time while maintaining the meaningful properties of the data, however, there was no solution to the low amount of data. The implications of the low data would be fully considered in the conclusion.

This data would be standardized to prevent biases from infiltrating the classification algorithms. Machine learning methods would also test the efficacy of using keystroke data to assess Parkinson’s disease.

1. **Timeline of Action**

To finish this project quickly and effectively, I created a plan of action with my mentor, Dr Chad Curtis.

Week 1: Perform a literature review of keylogging and RSI

Week 2: Do a deep-dive read into the PD/keystroke papers and dataset

Week 3-5: Perform preliminary feature extraction and normalization of data

Week 6-8: Conclude data normalization and implement machine learning algorithms on datasets obtained from the internet or generate our own data

While we were unable to get our own dataset in the end, we were able to access data procured from Tappy and used that to create this project. Ideally, we would have been able to generate our own data, which would have enabled us to have greater control over how much data we get, what data we can use and which conditions we want to extract information from, the data we used, as mentioned before, is highly accurate and reliable.

Chart, bar chart, waterfall chart

Description automatically generated

Figure: Gantt Chart

1. **Challenges**

Throughout this project, I faced quite a few challenges.

To begin with, starting with the literature review was quite challenging. I did not expect my project to involve such dense reading and, as a result, I spent quite a lot of time trying to both find papers that would support or challenge my proposed approach, so that I could best adapt it to be an effective algorithm and identify the specific features of Parkinson’s disease. As demonstrated in the Gantt Chart above, the literature review took a week longer than I expected and, as a result, I needed to make some changes. This was for the best, however, as I was able to better understand both how to implement my machine learning models.

Furthermore, the content in the literature I read was quite complicated. Identifying key symptoms of Parkinson’s disease such as bradykinesia and evaluating the efficacy of my approach were of vital importance, so that I could create a replicable and powerful approach. While, at first, it was difficult to wrap my head around these concepts, after focusing my attention onto reading the papers until I understood them and referencing both my professor and online and printed sources, I was able to understand the background behind my problem statement.

The next challenge was in choosing the appropriate data from the Tappy dataset. The dataset consisted of people who were unaffected, mildly affected and severely affected by Parkinson’s disease. Additionally, some people took medicine like levodopa, which would mask their symptoms. Clearly, I would need to only select people who possessed mild symptoms of Parkinson’s disease and did not take any medicine to properly predict early incidence of Parkinson’s disease. However, manually going through the data would be challenging. I decided to automate this process using the OS library in Python to extract certain keywords (‘Parkinson’s: Mild’ and ‘Medicine: No’) so that I could quickly obtain this data.

Additionally, extracting the data was not enough. I would need to extract useful characteristics (features) of this data to make meaningful comparisons and obtain useful results. Unfortunately, the data was quite small, but computationally intense. Additionally, it was formatted in a way such that my algorithms could not make direct inferences. I needed to both convert my data to a more legible format for computers and normalise it to prevent noise from interfering with my results. I accomplished this by calculating the mean and median of the data and scaling the data to this. I also discarded any values that proved to be noisy (possessing meaningless additional information) and performed feature reduction to reduce this noise. Finally, to address the issue of the low testing data, I used a technique called k folds cross validation, where I would split my data into 5 separate folds, perform analysis on each fold and aggregate all the results into one result. This effectively increased my testing data 5 fold and reduced computational complexity.

Finally, my initial proposed approach, which was using a machine learning model to assess the instance of Parkinson’s disease was found to be inaccurate as one model would struggle to constrain all the features I had extracted. This would be problematic as it would defeat the purpose of my project if the algorithm was inaccurate entirely. I overcame this inaccuracy by creating an ensemble method, a machine learning model that “pools” all the results from various, different models and weighs them to give a final probability of having Parkinson’s disease. This ensemble method proved to be greater than the sum of its constituent machine learning models and demonstrated the viability of my project.

**Technical Documentation**

1. **Literature Review**

In the first week, my mentor and I discussed the scope of the problem and to what extent we should take this project forward. We settled on using a keylogger to diagnose Parkinson’s Disease. We reviewed the existing literature on using keystroke data in authenticating a user or identifying biological traits about them. We also reviewed the literature surrounding Parkinson’s disease to better understand the disease and identify important features to detect the disease.

1. **Data Filtering**

Using the !wget command, we retrive and download the Parkinson's Data we need to analyse. We proceed to unzip the zipped folder enclosed in the Tappy Keystroke Data file. The first folder we unzip, Archived-Data, contains the file Tappy\_Data. Tappy\_Data is essentially all the keystroke data for an individual user. The second folder, Archived-users, contains the Archived\_users folder which is data about each individual user. While Tappy\_Data is of greater interest to this project considering it contains all the relevant keystroke data, we will first analyse Archived\_users to filter the list of users we need to analyse. Not only would this help us to speed up the process of feature extraction from the keystroke data, but it would also help the sensitivity and specificity of our program as we would get relevant data from more useful users. For ease of reference, a separate directory called data is made. This directory houses both the Tappy\_Data folder and the Archived\_users folder.

In pseudo-code, this is the algorithm used:

*# filter out mild PD users into users\_pd*

*# filter out non-PD users into users\_no*

*create empty features*

*for user in users\_pd:*

*user\_files = files associated with a single user*

*create empty total\_user\_file*

*for file in user\_files:*

*read in file*

*drop columns that aren't needed*

*append file to total\_user\_file*

*calculate user\_features*

*append user\_features to features*

*Repeat the above with users\_no*

At the end of the program we print the i variable and recieve 90. This is the number of users who matched our condition and part of the users whose data we plan on analysing and testing the Machine Learning algorithm on.

1. **Feature Extraction**

We first create two list variables, holds\_names and lat\_names, to store the name of the headings for our two dataframes, holds and latency. This is done to improve the readability of our code.

Next, we create two sets of variables, two lists: holds\_data and latency\_data. holds\_data is created to store an instance of our 9 features (the skewness, kurtosis, standard deviation and mean for L and R, as well as the absolute mean difference between them) for the holds dataframe. latency\_data, much like holds\_data, stores our features (18 rather than 9, them being the skewness, kurtosis, standard deviation and mean for a pair of LL, RR, RL and LR, along with the absolute mean difference between LL and RR, and RL and LR)

We, then, create six DataFrame variables – hdf, ldf, df1, df2, holds, and latency. hdf is created to store holds\_data's values as a DataFrame and ldf has the same function, but instead for latency\_data. df2 stores the an instance of a text file in Archived users who match our criteria while df1 stores every text file associated with a particular user who match our criteria. holds stores all the 9 features for a user who matches our criteria while latency stores 18 features.

We first create a for loop where we cycle through every user in our users list.

We create a variable, filenames, to store the list of files in Tappy Data. We then check them using an if statement.

If they contain a username in userrs, then we proceed onwards. If not, then the loop runs again until a username contained in userrs is found.

If the condition is true, we create a new file, output\_file.txt, using the with open method. This is done to store all the values of the text files in Tappy Data in one location. To access output\_file easily, and read and write to it, we create a TextIOWrapper, outfile, We then create a nested for loop where we iterate every user in filenames who match our condition as another TextIOWrapper called infile. We read the now opened infile using the read method, and write its contents to outfile using the write method. This allows the data to be written concretely in output\_file.txt.

We, now exit this second for loop and append the data present in output\_file to df2. We also crucically add in the argument, header=None, delimiter='\t', usecols=[0,3,4,5,6] for many reasons. Header=None is added as there are no headers in our text files, delimiter='\t' is added as the delimiter in this case is a tab, and usecols is added as we only need the 1st, 4th, 5th, 6th and 7th columns and using the other columns would use a lot of unnecessary memory. df2's data is quickly appened to df1. The reason we can't directly append output\_file's data to df1 directly is because with every increment of the for loop, data would be lost, so we have to store this data in with the help of two DataFrames.

After all the text files associated with a user are appended into df1, we check it's length using another if statement. If the number of rows is atleast 2000, only then do we proceed onwards to the feature extraction process. Otherwise, we clear the data of the DataFrames and start again.

If the data has at least 2000 rows, then we use the to\_numeric method to convert the 5th column of our DataFrame into a numeric type, provided that the 4th column is 'L'. We store this numeric in holdl, a local variable that calculates our required features. We also add in argument, errors=coerce to convert any non numeric value into a NaN (Not a Number) and remove these erroneous rows using the dropna method.

We repeat this process to retrive the features for the Right keystrokes, calculate the absolute mean difference, and finally store these values into holds\_data. We convert this data into a DataFrame by using the read\_csv method and store it in hdf, and append the hdf DataFrame to the holds DataFrame.

Finally, we repeat this process for the latency DataFrame with a few key changes:

1) We check 6th row rather than the 4th row,

2) We convert the 7th row to a numeric value rather than the 5th row, and

3) We store the values in different variables.

Now, finally, we can refresh the value of the df1 DataFrame to null so that no repeat data is accidently added and we are!

1. **Feature Reduction**

Using principle component analysis (PCA) and linear discriminant analysis (LDA), I was able to reduce the dimensions of the data and make it easier to manage. We found that we could capture about 60% of the variation in the data using a single feature from PCA. We resolved some initial confusion surrounding LDA-- for a dataset with two classes of outputs, we project our input data onto a single dimension.

1. **Creation of Machine Learning Algorithms**

Using instances of sci-kit learn’s machine learning I got my machine learning algorithms running! I was able to get a start on working with different train-test splits, and found that different splits gave widely different accuracies, anywhere between 30% to 70%. My model was ready to be wrapped together into a meta-predictor to give a single accuracy measurement per run.

**Conclusion**

1. **Health and Safety Notes and Ethical Considerations**

This experiment involved no health and safety concerns since this relied on computers and data processing and was conducted in the safety of my home. Additionally, there were no ethical considerations with the data. The patients were aware of how their keystroke data would be used and sufficient measures were taken by the original data collector’s author to ensure that no sensitive information was revealed.

1. **Evaluation**

With the conclusion of this project, it is important to evaluate if it managed to succeed at its key objectives and determine the strengths and limitations of the approach chosen.

Firstly, the project was partially successful in the first objective, which was to determine create a high accuracy, highly specific and sensitive algorithm to detect Parkinson’s disease. As could be noted, the accuracy would vary very dramatically between different models, from 30% to 70% and even with the introduction of an ensemble model, the accuracy was equivalent to professional testing. However, the algorithm had a high recall (that is, it would correctly select patients who may have Parkinson’s disease), which means that it would rarely let a false negative happen, which is very useful for our purposes as the disparity between the risk of a false negative and the risk of a false positive is quite high.

Next, this test would work for most motor dysfunction diseases, and this has two interesting implications. Firstly, this means that this project can only be made more accurate to the degree required by doctors by adding more biomarkers or features, meaning that this project has great potential to be used in clinical environments. Secondly, and more importantly, this project can be easily extended to other neuro-degenerative diseases such as RSI, which makes it multifaceted and useful for a variety of situations. This means that while the project was not highly specific, it still is a good predictor for Parkinson’s disease and can be used for other disease prognosis and diagnosis. However, it is still important to note that certain additional biomarkers or features may be required to have full confidence in the results of the algorithm and as such, the algorithm was not as specific as was desired.

This approach also faced a major hurdle in the form of data. As noted in the challenges section, the data obtained was very limited (200 patients) and even though techniques like k folds cross validation were used to increase the data, this was not a complete substitute for the lack of data. In the future, it would be best to obtain greater data, so that the algorithms can extract meaningful information, reduce the noise of the data, and observe a clearer pattern.

Finally, this project was very successful in demonstrating the degree to which machine learning can aid in clinical diagnosis. Even with limited data, the algorithm was able to produce very promising results for predicting the incidence of Parkinson’s disease and this approach also has the benefit of being able to be administered remotely (which is something that is becoming quite important amidst the current pandemic) and administered independently, which could free up the clinician to do more pressing tasks. Additionally, as an algorithm is able to compute much more information than a human, this project could expedite the process of identifying Parkinson’s disease, enabling patients to get better care and mitigate the effects of it.

1. **Personal Reflections and Learnings**

Throughout this project, I learnt a lot about machine learning and about thinking like a data scientist. I had to make many compromises with my data and implementation and had to work long hours to ensure that the implementation was correct and that the algorithms worked properly. I am pleased that I managed to fulfill the goals of my project in a timely timeframe. In retrospect, I now recognize the importance of utilizing biomarkers and other data since they provide unambiguous results relating to the detection of Parkinson’s disease, but this project still succeeded in its core goal of demonstrating that keystroke data is viable to detect Parkinson’s Disease.

The project is easy to use as one would just need to download a software to securely record their keystrokes and my project would give the probability of them having Parkinson's disease upon inputting the data. This project utilises the kurtosis, mean and median of the right handed and left handed latency and hold time between keystrokes. These metrics are accurate indicators to detect abnormalities during keystokes and are a good indicator of someone possessing Parkinson's disease. As a result, I believe that my algorithm could be used by doctors as an alternative test for Parkinson's disease. It has the advantage of being self-administered, allowing the patient to have easy access by simply downloading this algorithm.

1. **Bibliography**

Adams, Warwick. *Tappy Keystroke Data*. physionet.org, 2017. *DOI.org (Datacite)*, <https://doi.org/10.13026/C2K08D>.

*Bartmann D, Bakdi I, Achatz M. On the Design of an Authentication System Based on Keystroke Dynamics Using a Predefined Input Text. Int J Inf Secur Priv. 2007;1(2):1–12.*[[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Int+J+Inf+Secur+Priv&title=On+the+Design+of+an+Authentication+System+Based+on+Keystroke+Dynamics+Using+a+Predefined+Input+Text&author=D+Bartmann&author=I+Bakdi&author=M+Achatz&volume=1&issue=2&publication_year=2007&pages=1-12&)]

“Diagnosis.” *Parkinson’s Foundation*, <https://www.parkinson.org/Understanding-Parkinsons/Diagnosis>. Accessed 21 Sept. 2021.

Giot R, Rosenberger C. A new soft biometric approach for keystroke dynamics based on gender recognition. Int J Inf Technol Manag. 2012;11(August):35. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Int+J+Inf+Technol+Manag&title=A+new+soft+biometric+approach+for+keystroke+dynamics+based+on+gender+recognition&author=R+Giot&author=C+Rosenberger&volume=11&publication_year=2012&pages=35&)]

Kang SJ, Choi JH, Kim YJ, Ma H-I, Lee U. Development of an acquisition and visualization of forearm tremors and pronation/supination motor activities in a smartphone based environment for an early diagnosis of Parkinson’s disease. Adv Sci Technol Lett. 2015;116:209–12. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Adv+Sci+Technol+Lett&title=Development+of+an+acquisition+and+visualization+of+forearm+tremors+and+pronation/supination+motor+activities+in+a+smartphone+based+environment+for+an+early+diagnosis+of+Parkinson%E2%80%99s+disease&author=SJ+Kang&author=JH+Choi&author=YJ+Kim&author=H-I+Ma&author=U+Lee&volume=116&publication_year=2015&pages=209-12&)]

“Parkinson’s Disease.” *National Institute on Aging*, <http://www.nia.nih.gov/health/parkinsons-disease>. Accessed 21 Sept. 2021.

“Statistics.” *Parkinson’s Foundation*, <https://www.parkinson.org/Understanding-Parkinsons/Statistics>. Accessed 21 Sept. 2021.

Teh P, Teoh A, Tee C, Ong T. A multiple layer fusion approach on keystroke dynamics. Pattern Anal Appl. 2011;14(1):23–36. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Pattern+Anal+Appl&title=A+multiple+layer+fusion+approach+on+keystroke+dynamics&author=P+Teh&author=A+Teoh&author=C+Tee&author=T+Ong&volume=14&issue=1&publication_year=2011&pages=23-36&)]

Wang, Wu, et al. “Early Detection of Parkinson’s Disease Using Deep Learning and Machine Learning.” *IEEE Access*, vol. 8, 2020, pp. 147635–46. *IEEE Xplore*, <https://doi.org/10.1109/ACCESS.2020.3016062>.

Yang, Wenya, et al. “Current and Projected Future Economic Burden of Parkinson’s Disease in the U.S.” *Npj Parkinson’s Disease*, vol. 6, no. 1, July 2020, pp. 1–9. *www.nature.com*, <https://doi.org/10.1038/s41531-020-0117-1>.

1. **Acknowledgements**

I would like to thank my parents for encouraging me to do this project and helping me stay motivated during it. I would also like to thank my mentor, without whom this project may have never seen the light of day. His evaluations and suggestions were vital to the success of this project. Finally, I would like to thank Warwick Adams and his work in collecting this data, without which this project would not have been possible.

1. “Parkinson’s Disease.” *National Institute on Aging*, <http://www.nia.nih.gov/health/parkinsons-disease>. Accessed 21 Sept. 2021. [↑](#footnote-ref-1)
2. “Statistics.” Parkinson’s Foundation, <https://www.parkinson.org/Understanding-Parkinsons/Statistics>. Accessed 21 Sept. 2021. [↑](#footnote-ref-2)
3. Yang, Wenya, et al. “Current and Projected Future Economic Burden of Parkinson’s Disease in the U.S.” Npj Parkinson’s Disease, vol. 6, no. 1, July 2020, pp. 1–9. www.nature.com, <https://doi.org/10.1038/s41531-020-0117-1>. [↑](#footnote-ref-3)
4. Wang, Wu, et al. “Early Detection of Parkinson’s Disease Using Deep Learning and Machine Learning.” *IEEE Access*, vol. 8, 2020, pp. 147635–46. *IEEE Xplore*, <https://doi.org/10.1109/ACCESS.2020.3016062>. [↑](#footnote-ref-4)
5. “Diagnosis.” Parkinson’s Foundation, <https://www.parkinson.org/Understanding-Parkinsons/Diagnosis>. Accessed 21 Sept. 2021. [↑](#footnote-ref-5)
6. Pagan FL Am J Manag Care. 2012 Sep; 18(7 Suppl):S176-82. [↑](#footnote-ref-6)
7. 7 Bartmann D, Bakdi I, Achatz M. On the Design of an Authentication System Based on Keystroke Dynamics Using a Predefined Input Text. Int J Inf Secur Priv. 2007;1(2):1–12. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Int+J+Inf+Secur+Priv&title=On+the+Design+of+an+Authentication+System+Based+on+Keystroke+Dynamics+Using+a+Predefined+Input+Text&author=D+Bartmann&author=I+Bakdi&author=M+Achatz&volume=1&issue=2&publication_year=2007&pages=1-12&)]

   8  Giot R, Rosenberger C. A new soft biometric approach for keystroke dynamics based on gender recognition. Int J Inf Technol Manag. 2012;11(August):35. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Int+J+Inf+Technol+Manag&title=A+new+soft+biometric+approach+for+keystroke+dynamics+based+on+gender+recognition&author=R+Giot&author=C+Rosenberger&volume=11&publication_year=2012&pages=35&)]

   9 Kang SJ, Choi JH, Kim YJ, Ma H-I, Lee U. Development of an acquisition and visualization of forearm tremors and pronation/supination motor activities in a smartphone based environment for an early diagnosis of Parkinson’s disease. Adv Sci Technol Lett. 2015;116:209–12. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Adv+Sci+Technol+Lett&title=Development+of+an+acquisition+and+visualization+of+forearm+tremors+and+pronation/supination+motor+activities+in+a+smartphone+based+environment+for+an+early+diagnosis+of+Parkinson%E2%80%99s+disease&author=SJ+Kang&author=JH+Choi&author=YJ+Kim&author=H-I+Ma&author=U+Lee&volume=116&publication_year=2015&pages=209-12&)]

   10 Teh P, Teoh A, Tee C, Ong T. A multiple layer fusion approach on keystroke dynamics. Pattern Anal Appl. 2011;14(1):23–36. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Pattern+Anal+Appl&title=A+multiple+layer+fusion+approach+on+keystroke+dynamics&author=P+Teh&author=A+Teoh&author=C+Tee&author=T+Ong&volume=14&issue=1&publication_year=2011&pages=23-36&)]

   11 Adams, Warwick. *Tappy Keystroke Data*. physionet.org, 2017. *DOI.org (Datacite)*, <https://doi.org/10.13026/C2K08D>. [↑](#footnote-ref-7)