Two-Factor Identification Through Heart Rate and Gait Analysis

Anna Soper, Ethan Carroll, Drew Child, Lauren Le

Abstract— **This research project investigates the possibility of identifying unique heart rate and gait patterns in individuals. Through the use of mathematical analysis and neural networks, these patterns can be used as an alternative authentication method to passcodes and other biometric scanners.**

I. Introduction

This project explores different techniques in which the unique electrocardiography and gait of a person can be interpreted to distinguish them from other people. Electrocardiography data was obtained from a heart rate sensor, and human gait data was obtained from measuring the three-dimensional acceleration movement patterns from an accelerometer. A computational model was used to analyze the data for four different people in order to identify a specific person for a specific heart rate or walking pattern. The purpose of this project was to understand how neural networks can be used, and its applications to online security systems and accuracy of user identification, like in [1]. The notion of user identification based on heart rate or gait is used in companies who prefer that over creating or remembering passwords, as described in [2]. While there are sources such as [3] explaining the utility of heart rate and human gait as security measures, they are not yet popular in everyday use [4]. Thus, part of the project investigates different barriers and accuracy of such security methods, discussed in [5], and can bring more insight as to how innate qualities can be used to enhance overall cyber security.

II. Scientific and Engineering Goals

A. *Accelerometer and Pulse Sensor:*

The two sensors in the project include the Arduino Pulse Sensor XD-58C [1] and accelerometer MPU-6050 [1]. See Section IV for more information on the scientific data it measures. The purpose of these sensors are to generate a photoplethysmograph [1] from the pulse sensor and to obtain the three dimensional components of a person's gait. The importance of this data is to find unique identifying characteristics of a person based on their heart rate and micromovement acceleration patterns during walking. The goal is to utilize the data in a meaningful way, through mathematical analysis, to feed into a neural network and ultimately to correctly identify a person from their heart rate or walking pattern at a given time. Learning goals of using these sensors include circuit design to connect to the Elegoo Arduino board, to run a program that takes samples of data on the order of milliseconds without time delay, and to determine how to record standard data through the serial monitor in text files. Moreover, the sensor measurements will contribute to the learning objective of normalizing, interpreting, and filtering noisy data in order to understand its impact on the precision of the neural network system.

[1]

B. *Feedforward neural net*

The goal of implementing a feedforward neural network is so that a person can input their raw data into it, and have it report back that they've been identified. This concept is based on the architecture of the network, in which there is an input layer, multiple hidden layers, and then an output layer, described in [6]. To achieve the goal, the neural network used in this project was trained with 4 different people, each with 10 training datasets and 5 testing datasets. Distinguishing features of different people from their data were extracted, and then passed into the neural network. The goal of this is to provide sufficient and unique numerical data of each person,and to lessen noise that may contribute to error during the machine learning process, as in [7]. Learning objectives of exploring this computational model is to use Matlab to interpret statistical (e.g. quartile, mean, etc.) and informational (low and high frequencies of sinusoidal patterns) data to pass into build-in machine learning functions. Another learning goal includes understanding the extent to which the feedforward neural network can find patterns [8] within noisy data by comparing its precision results to a simple feature comparison algorithm.

C. *Measurements:*

In order to achieve learning objectives discussed previously about feedforward neural networks, the following combinations were used to create comparison algorithms in 1a-1c, and to train different types of neural networks in 2a-3c:

1a. Parameter comparison, accelerometer

1b. Parameter comparison, heart sensor
1c. Parameter comparison both sensors
2a. Net, raw data, heart sensor
2b. Net, raw data, accelerometer
2c. Net, raw data, both sensors
3a. Net, key features data, heart sensor
3b. Net, key features data, accelerometer
3c. Net, key features data, both sensors

III. Mathematical Model

In this experiment, two forms of data analysis were used to identify a person from their gait and heart rate data. The first type of analysis calculated specific features from the raw data. These features include average magnitude, average distance from the mean, spectral centroid, the first three frequency quartiles, and maximum frequency from the FFT. The characteristic features that most closely matched the test data were used to predict the identity of the person whom the test data came from. The second method of analysis used pattern recognition in a neural network to predict the person whom the data came from.

A. Features

The initial form of analysis was by feature matching between the training data and the test data. Seven features were extracted for each person for both the accelerometer and heart rate sensor. The features used to characterize the gait acceleration were chosen based off of the gait analysis from [1].

i) Magnitude: The Magnitude of the data is defined as the root sum of squared of the acceleration from each axis. It is calculated as shown in Equation (1).

$$Magnitude = \sum_{k=1}^{l} \sqrt{(x_k^2 + y_k^2 + z_k^2)} \quad (1)$$

Where x, y, and z represent the acceleration from all three axes, and k represents the index of the k'th data point.

Magnitude was chosen as a defining features because it the magnitude of acceleration of a person walking is related to their weight [source***] and the kinematics of how they move their body.

ii) Spectral Centroid: In audio processing, spectral centroid is related to the "brightness" of a sound measure, but for a general set of data in the frequency domain, the spectral centroid is the weighted mean of a set of frequency data [9]. It is calculated as shown in Equation (2).

$$Centroid = \sum x_i f_i / l \quad (2)$$

Although there is not a one to one map between spectral centroid and some physical characteristic of an individual, this feature has been used successfully by [10] to distinguish between different groups of people.

iii) Gait Frequency: According to [11], gait frequency is related to a person's height. This is because taller people are more likely to have a longer stride and thus a lower frequency. Therefore the normal walking frequencies can be used to distinguish between different people

iv) Resting Heart Rate: Similar to gait frequency the rest heart rate frequency was used to distinguish between people. This feature was chosen because according to [12] peoples resting heart rate is relatively consistent over time, and can vary between individuals by up to seventy beats per minute.

iv) Frequency Quartiles: Frequency quartiles are a measure of the spread of frequencies within a spectrum. The first quartile frequency is calculated as shown in Equation (3).

$$\frac{1}{4} = \int_0^{f_1} c_f \, df \quad (3)$$

Where f_1 is the quartile 1 frequency- the frequency that is reached when a normalized frequency spectrum has been integrated to a value of ¼. The Third and fourth quartiles are calculated in the same way, with ½ and ¾ replacing ¼ respectively.

v) Average Difference From Mean: The average difference from the mean is a measure of the variation in a data set [13]. It is calculated as shown in Equation (4).

$$Diff = Avg(|x_t - x_{mean}|) \quad (4)$$

B. Neural Net

The main form of data analysis that was necessary in this project was pattern recognition. One way to get accurate results in this type of analysis is using a neural network, and for this project a feedforward neural network was used. Feedforward networks are composed of the input layer, then a number of hidden layers, and finally the output layer. In a neural net each hidden layer is composed of hidden nodes, which are comparable to the neurons in people's brains. Initially, each hidden node is connected to every node in the layer before and after it and after a set of training data is run through the net the weights and biases of the nodes are determined. These weights and biases are used to determine whether a node will be activated depending on the signal it takes in from the prior layer.[14]]

Equation 5 represents the sigmoid function. This is the most common activation function given its shape and that its values are constrained between 0 and 1.

$$S = 1/(1 + e^{-1}) \quad (5)$$

It's also more popular than some other more simple activation functions because the sigmoid function allows for a wide variation of outputs given that it can output any number between 0 and 1. This means that the outputs can represent probabilities as opposed to just binary results.[15]

Matlab has a collection of functions useful to creating a neural network in their deep learning toolbox. The first iteration of the network used a function called "nnstart" which resulted in a self generated neural network function after inputting training data. Unfortunately, this function was only capable of creating single layer neural networks. This was not acceptable given the complexity of the data that the sensors generated. Therefore the final data analysis used the function "patternnet," which is visible in Figure 1, to create the neural net and "train" to train it.

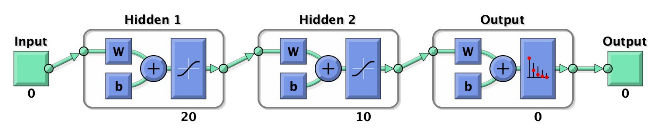

Figure 1. This image represents a basic neural net created in Matlab using the Patternnet function.

Patternnet allows for any number of hidden layers and nodes to be input into the net which is helpful given that multiple networks were created with varying degrees of complexity. The network demonstrated in the first figure is an example of a network with two hidden layers, the first layer with 20 nodes and the second with 10 nodes. This would be good for a data set that has less than 20 inputs and one where the outputs can be easily derived from the inputs. The neural net used in this project required more hidden layers given the complexity of the data and the noise generated from the sensors The networks used in the project had seven layers with 600, 40, or 20 nodes for raw data, combined feature data, and parameter data respectively. Patternnet is also useful because it allows for a reliable and accurate network to be created with very little code

$$Mean = \sum x_i / N \quad (6)$$

$$Average\ Distance\ to\ Mean = \sum (x_i - Mean(x))/N \quad (7)$$

In total seven networks were created using the patternnet function and the inputs were, the raw data for both the heart rate sensor, and accelerometer, combined raw data for both, the FFT of the heart rate sensor data and a collection of data parameters for both the heart rate sensor, accelerometer and then both combined. The parameters included the average magnitude of the data, the average distance from the mean, the centroid, the first three quartiles, and the maximum frequency from the FFT. Equations 4 and 5 represent two of the values included in the data parameters, the mean and average distance to the mean. For each neural net the outputs were either 0001, 0010, 0100, or 1000 depending on which of the four people the data belonged to.

IV. Methods

A. Sensors

In order to record heart rate and accelerometer data, the XD-58C [16] heart rate sensor and the MPU-6050 [17] accelerometer (Figure 2) were chosen due to their low cost, ease of implementation with the Arduino board, and practicality for deployment.

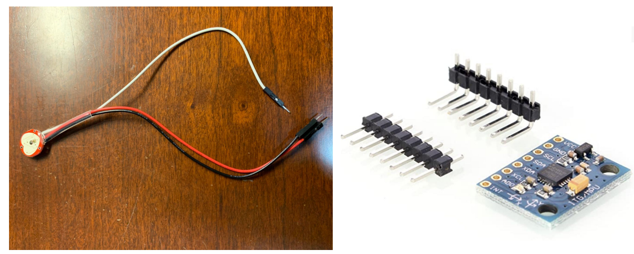

Figure 2. XD-58C heart rate sensor (left) and MPU-6050 accelerometer (right).

The XD-58C heart rate sensor works by transmitting green light through the skin and measures the intensity of reflected light as it varies with blood flow [18]. The MPU-6050 is a 3 axis gyroscope and 3 axis accelerometer that is compatible with the Arduino interface and is capable of recording acceleration up to ±16g.

In discussing the benefits of each sensor, it is also important to recognize the shortcomings. Since the heart rate sensor operates using reflected light intensity readings [18], high amounts of noise are embedded in the output readings. Varying the pressure applied to the heart rate sensor will also change the average output independent of a user's actual blood flow.

B. Procedure

Due to their compatibility with the Arduino, both the MPU-6050 and XD-58C are simple to implement. The XD-58C has 3 pins: 5V, GND, and A0. These pins can be inserted directly into the corresponding ports on the Arduino board as shown in Figure 3.

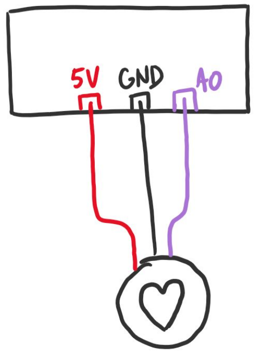

Figure 3. diagram of heart rate sensor connection to Arduino

The MPU-6050 has 8 pins, 7 of which are connected to the Ardunio with 1 pin left unused. The MPU-6050 needs the connecting pins soldered onto the board before connecting them directly to the corresponding ports on the Arduino board according to Figure 4.

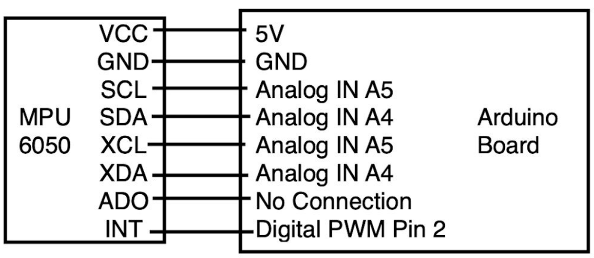

Figure 4. Diagram of MPU-6050 connection to Arduino

C. Data Collection

Ideally, a micro SD card would be used in conjunction with the Arduino board in order to save data locally before downloading onto the computer. However, due to a lack of additional budget for a micro SD card, it was determined that the Arduino's built-in storage would be sufficient for recording accelerometer data and the heart rate data could be uploaded in real time to the serial monitor. Therefore, the USB cable included in the "Super Starter Kit UNO R3 Project" [19] was used in lieu of the 9V battery due to it's multi-functionality as a power source and data connection.

The possible sampling locations of the XD-58C are on the earlobe and tip of a finger, as recommended by the provider. After testing both locations, the tip of the pointer finger was chosen due to its convenience and consistent data readings when attached by tape or velcro (Figure 5). The sensor requires a medium amount of pressure such that the subject can feel their heart beat at the sampling point.

Figure 5. The experimental setup of the heart rate sensor, fastened to the index finger with velcro

Plotting the data from the serial monitor yields Figure 6.

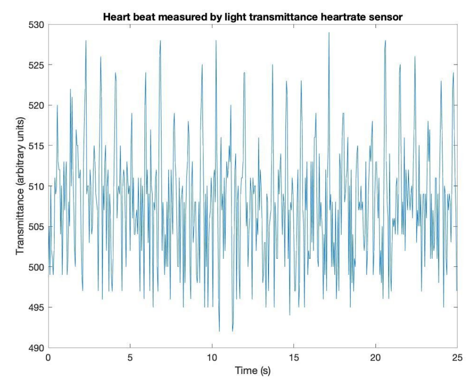

Figure 6. Plot of heart rate sensor raw data. Repeating peaks are clearly distinguishable in the raw data, which indicate each heart beat.

Possible placement locations of the MPU-6050 include the center of the waist, chest, lower back, and side hip. Though the waist, chest, and lower back are good locations for consistent results [17], higher variability between different people is actually desired in order to distinguish walking patterns between individuals. Therefore, it was hypothesized that the side hip would yield more variability between individuals, yet offer consistent enough results for a given individual in order to identify the characteristics of a person's gait. The accelerometer was fastened onto the right hip with tape or a tight waistband while ensuring that the x, y, and z axis directions were standardized across all individuals (Figure 7).

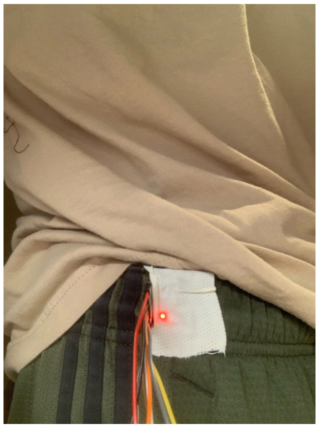

Figure 7. Experimental setup of the accelerometer, fastened to the right hip with tape.

In order to eliminate as many outside variables as possible while collecting data, a set of guidelines for conducting trials and saving data was established. All arduino code was standardized, including sampling frequency and duration. The heart rate sensor used a 50 ms sampling period for a duration of 25 seconds, resulting in 500 data points per trial. The accelerometer had a 50ms sampling period, but only could record 100 data points per axis for a total of 5 seconds per trial. This was due to the limitations of local storage on the Arduino board.

These walking trials were performed by walking in a straight line at a consistent pace for the entire duration of each trial. 15 total data sets were taken for each sensor. 10 of these trials were to be used to train the neural network and identify common parameters, with the remaining 5 trials to be used as testing data.

The stored data was uploaded onto the computer via an application called “CoolTerm”, which writes the Arduino serial monitor directly into a txt file. The txt file contains 3 columns: one for each axis reading of acceleration. Figure 8 shows a plot of the raw data, with the x and z axes centered around 0 and the y axis centered around 1. This is expected, as the unit “1” corresponds to 1g, or the force due to gravity 9.8m/sec^2.

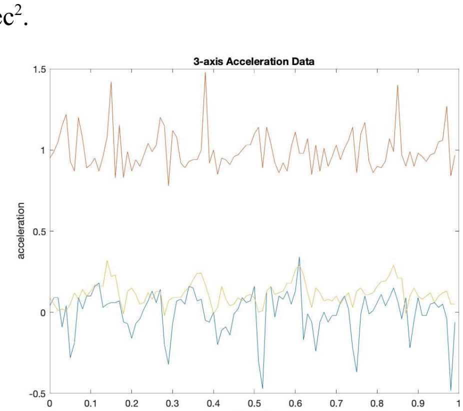

Figure 8. An acceleration plot of all 3 axes, with x, z centered around 0 (blue, yellow respectively) and y centered around 1 (red)

D. Data Processing

A total of 15 heart rate and acceleration data sets were collected for four people. Ten of the data sets were labeled training data and five were labeled testing data. The data was processed using the three methods listed below.

1. Feature comparison
2. Neural net with raw data
3. Net net with features

The accuracies of each method were determined by for three different sets of input data: acceleration data, heart rate data, and both acceleration and heart rate data.

E. Data Processing: Feature Comparison Algorithm

The first data processing method calculated features from the data and used those features to predict the identity of the person corresponding to testing data. A feature vector was calculated for all fifteen data sets for all five test subjects. Then the first six feature vectors for each student were averaged element wise to create a characteristic feature vector for each test subject. The characteristic feature vectors for each subject were placed in the columns of a characteristic feature matrix. The characteristic feature matrix is what all test data was compared against to determine the identity of a set of test data.

In order to test the accuracy of the parameter method, a parameter vector was calculated from test data. The parameter vector was subtracted from each of the columns of the characteristic feature matrix and the absolute differences between the parameter vector and characteristic parameters totalled. The lowest absolute sum of differences score corresponded to the person whose characteristic parameters most closely matched the test data parameters.

the features extracted from the acceleration data were: average magnitude, average distance from the mean, spectral centroid, the first three frequency quartiles, and maximum frequency from the FFT. The features extracted from the heart rate data were the spectral centroid, the first three frequency quartiles, and maximum frequency from the FFT.

F. Data Processing: Neural Net with Raw Data

Three neural networks were trained in Matlab with raw heart rate data, vectorized raw acceleration data, and a combination of the latter two. The purpose of training the neural nets with raw data is to see if it could identify a person despite the noise in the electrocardiography and gait patterns. The acceleration data was vectorized (by taking the magnitude) so that the signs of the values would not have to be taken into account. The 10 acceleration training data sets from each person were vectorized and averaged, such that only one value was used to represent each training data set (and later testing data set). For each neural network, either the raw heart rate values, vectorized acceleration data, or both were grouped into one large array into the train() function in Matlab. See Section III for specific details on how the neural net was trained.

G. Data Processing: Neural Net with Features

This method of data analysis combines the prior two methods. First a neural net was created in Matlab using the "patternnet" function. The number of hidden layers and nodes was different depending on which data set we were using. Then the 15 data sets from both the heart rate sensor and accelerometer were turned into 15 parameter profiles for each member. Ten of these profiles were used as training data for the neural net, while the remaining five were used to test the net after training. The output of the neural net was one of four outputs, 1000, 0100, 0010, or 0001, corresponding to each of the four group members. To test accuracy the network was considered to have predicted correctly if the output node that was the highest corresponded to the person whose data was being run through the network.

V. Results

A. Overall Results

The precision matrix from Table 1. displays the precision of the each person identification algorithm when using different sets of data. Out of the three methods of identification, the feature comparison algorithm and the neural net with raw data each using data from both sensors were the most accurate methods, with precisions of 0.95.

PRECISION MATRIX			
	Acceleration	Heart Rate	Acceleration and Heart Rate
Feature Comparison	0.9	0.4	0.95
Neural Net with Raw Data	0.55	0.5	0.95
Neural Net with Features	0.25	0.75	0.2

Table. 1. Precision matrix for predictive power of each data analysis method.

The next most precise method was the feature comparison algorithm with acceleration features, with a precision of 0.9. In the next three subsections the specific methods and their accuracies will be explained in more depth.

A. Feature Comparison Algorithm

From looking at the top row of the precision matrix in Table 1. the precisions of 0.9 and 0.95 make it clear that the feature comparison algorithm was the most accurate when using acceleration data and both acceleration and heart rate data. A confusion matrix of the most accurate method— the feature comparison with heart rate and acceleration features— is shown in Figure 9. The 1's along the diagonal of the confusion matrix indicate that the algorithm was one hundred percent accurate when given features from person B, C, and D, and only eighty percent accurate when given features from person A.

Actual Person	**Predicted Person**			
	Person A	Person B	Person C	Person D
Person A	0.8	0.2	0	0
Person B	0	1	0	0
Person C	0	0	1	0
Person D	0	0	0	1

Figure 9. Confusion matrix for the feature method using only acceleration data.

Although this method was the most accurate, the confusion matrix shows that twenty percent of the time the algorithms mistook Person A for Person B.

Notice that the precision of the feature comparison method increased from 0.9 to 0.95 when the heart rate parameters were used in conjunction with the acceleration parameters. This makes sense because adding parameters gives the algorithm more ways to distinguish between different people.

The precision was only 0.4 when using only heart rate features. This low precision can be traced to the high variation in the heart rate parameters for a single person and the use of only five features (rather than the seven used with the acceleration data). Fewer features were used with the heart rate data because some features, such as the magnitude did not make sense to use. The heart rate sensor itself recorded intensity of light reflected of a person's finger, and this light intensity changed based on how tightly the sensor was strapped to the person's finger. for this reason, features relating to the magnitude of the data could not be used. The high variability and lower number of features meant that there was a higher chance that one person's features would be close enough to another person's features that the algorithm would make an incorrect prediction.

B. Neural Net with Raw Data

Based on the three different neural network models trained with raw data, there is clear numerical evidence that the neural net confuses data between the 4 different students when either the heart rate or acceleration data is used alone. For example, in the heart rate data, there were 3 students that were approximately indistinguishable, with the accuracy probability ranging between 20% to 60%. This was because the raw data for the 3 students were within the same range: 489 to 541. The 4th student's heart rate was 100% accurate, and his raw heart rate values were higher in magnitude (ranging from 513 to 576), corresponding to an elevated heart rate. This pattern that the neural network adapted was the contributing factor to his higher probability of being identified correctly. This is significant because it raises the issue of magnitude being a determining factor, when in reality, the heart rate electrocardiography and pattern should be more weighted. Thus, this calls for a normalized means of data that is seen later on in this section of parameterized data trained neural networks.

The same issue was encountered within the raw acceleration data and neural network. While two students were generally indistinguishable, with 60%-80% false positives, the other two students had a 100% accuracy. No distinguishing features that may have characterized the explanation behind half of the students' low accuracy was passed into the neural net.

In contrast to the prior two, the neural net with raw heart rate and raw acceleration data is accurate most of the time there are more meaningful results. In Figure 10. there are three 1's and 0.8 along the diagonal of the confusion matrix.

Actual Person	**Predicted Person**			
	Person A	Person B	Person C	Person D
Person A	1	0	0	0
Person B	0	1	0	0
Person C	0.2	0	0.8	0
Person D	0	0	0	1

Figure 10. Neural Net, Raw Heart Rate, and Raw Acceleration Data Confusion Matrix.

This indicates that a combination of the raw data has a high probability of correctly identifying a person. The 0's outside of the diagonal can be interpreted as less incorrect identification or mixes between students.

This underscores how the neural network can make more accurate connections between training datasets and testing datasets with more information-combined heart rate and acceleration data. It also showed that the neural network relies on numerical patterns, such as magnitude, with the training data in attempts to interpret testing data, rather than patterns between samples of one dataset. Moreover, it showed that the Matlab neural network didn't filter noise within the data. Thus, inputting important features of the raw data, which brings out determining characteristics of a person, into the neural network would be considered.

C. Neural Net with Feature Vector Input

Actual Person	**Predicted Person**			
	Person A	Person B	Person C	Person D
Person A	1	0	0	0
Person B	0	0.6	0.4	0
Person C	0.6	0	0.4	0
Person D	0	0	0	1

Figure 11. This table represents a confusion matrix for the heart rate parameters neural net. Each row corresponds to a different person's practice data.

There are a few main takeaways from Figure 11. One thing that is clear is that when the neural net predicts that the data running through it belongs to Person D it is correct. It also demonstrates that although when Person A's data was run through the net it predicted her data 100% of the time, it also predicted Person A, resulting in false positives, on 60% of Person C's data. Similar to Person D's data, since Person B's data showed up when we were running her data through the neural net, from this it is clear that whenever the net predicts Person B it is accurate. This is despite the fact that sometimes Person B's data incorrectly resulted in Person C being predicted by the neural net. Person C was predicted by the neural net when we passed in both his and Person B's data. This means that when the net predicts Person C 50% of the time it is correct in its prediction and 50% it is giving a false reading of Person B's data. From this one important takeaway from this table is that when the neural net predicts Person D or Person B it is correct, but when it predicts Person A or Person C it can either be correct or it can be incorrectly Person C or Person B's data.

An unexpected result that came up from the neural net when running through parameters was that when both the accelerometer parameters and heart rate parameters were run through the neural net the precision was worse than when just using the heart rate parameters. This is unusual because when looking at just the parameters alone, there seems to be more variation in the accelerometer's parameters. This would lead one to believe that adding the accelerometer's parameters to the neural net would give it more information to improve its accuracy, however, in reality this is not what happened. It is also unexpected given that for both the other methods, the parameter method and the neural net with raw data, the precision greatly increased, however, this is not what happened with the neural net.

VI. Conclusion

Working with a variety of methods to compare heart rate and accelerometer data taught several lessons. Due to the inability to collect data on one set of sensors, all data had to be standardized to reduce the effect of variance between sensors. When searching for patterns in a person's data, the neural network is able to identify many more distinguishing parameters than a simple mathematical analysis can do on its own. However, neural nets can be extremely complicated without the help of Matlab's built-in tools, which had a slight learning curve itself.

By utilizing both the heart rate data and accelerometer data, the precision of the neural net increased when raw data was passed in versus either one alone. Therefore, it can be concluded that using both identification methods in conjunction will improve the security of such an authentication method. Implementing a similar authentication method in the real world would be widely accessible, as many smartwatches and fitness wearables today already contain both a heart rate sensor and accelerometer. This would also be more cost friendly for manufacturers compared to using other sensors like fingerprint readers or 3D face scanners since it does not require additional hardware.

VII. Acknowledgements

We would like to acknowledge Fionna and Joan, for volunteering to be a part of the sample data, despite not being used in the final analysis of the data.

VIII. References

[1] "www.ncbi.nlm.nih.gov." [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7060195. [Accessed: 01-May-2020].Hoang, Choi, and Deokjai, "Secure and Privacy Enhanced Gait Authentication on Smart Phone," *The Scientific World Journal*, 14-May-2014. [Online]. Available: https://www.hindawi.com/journals/tswj/2014/438254/. [Accessed: 01-May-2020].

[2] H. Kelly, "5 biometric alternatives to the password," CNN, 04-Apr-2014. [Online]. Available: https://www.cnn.com/2014/04/04/tech/innovation/5-biometrics-future/index.html. [Accessed: 01-May-2020].

[3] N. Sreedhar, "Are you the password?," Livemint, 07-Jul-2017. [Online]. Available: https://www.livemint.com/Leisure/Nx6rvtFwXAiGontsiqWqkL/Are-you-the-password.html. [Accessed: 01-May-2020].

[4] Hoang, Choi, and Deokjai, "Secure and Privacy Enhanced Gait Authentication on Smart Phone," *The Scientific World Journal*, 14-May-2014. [Online]. Available: https://www.hindawi.com/journals/tswj/2014/438254/. [Accessed: 01-May-2020].

[5] Nahid and M. H. Habaebi1, "IOPscience," IOPscience. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/53/1/012069. [Accessed: 01-May-2020].

[6] V. Bushaev, "How do we 'train' neural networks ?," Medium, 22-Oct-2018. [Online]. Available: https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73. [Accessed: 01-May-2020].

[7] Y. Upadhyay, "Feed forward Neural Networks," Medium, 08-Mar-2019. [Online]. Available: https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620. [Accessed: 01-May-2020].

[8] "Select a Web Site," Classify Patterns with a Shallow Neural Network - MATLAB & Simulink. [Online]. Available: https://www.mathworks.com/help/deeplearning/gs/classify-patterns-with-a-neural-network.html;jsessionid=c232f2f3d9746107a8a39fe6d989. [Accessed: 01-May-2020].

[9] A. Ries, C. Chang, S. Glim, C. Meng, C. Sorg, and A. Wohlschläger, "Grading of Frequency Spectral Centroid Across Resting-State Networks," Frontiers in Human Neuroscience, vol. 12, no. 436, Oct. 2018.

[10] A. Millecamps, K. A. Lowry, J. S. Brach, S. Perera, M. S. Redfern, and E. Sejdić, "Understanding the effects of pre-processing on extracted signal features from gait accelerometry signals," Computers in Biology and Medicine, vol. 62, pp. 164–174, Apr. 2015.

[11] G. Soleimani, M. E. Andani, H. Marateb and F. Bahrami, "Speed influences on complicated kinetic, kinematic and electromyographic gait data of lower extremities among healthy subjects," 2016 23rd Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME), Tehran, 2016, pp. 321-326.

[12] G. Quer, P. Gouda, M. Galarnyk, E. J. Topol, and S. R. Steinhubl, "Inter- and intraindividual variability in daily resting heart rate and its associations with age, sex, sleep, BMI, and time of year: Retrospective, longitudinal cohort study of 92,457 adults," Plos One, vol. 15, no. 2, Feb. 2020.

[13] Singha, T.B., Nath, R.K., & Narsimhadhan, A.V. "Person Recognition using Smartphones' Accelerometer Data". ArXiv, abs/1711.04689. 2017. Apr 22, 2020

[14] T. Gupta, "Deep Learning: Feedforward Neural Network," Medium, 16-Dec-2018. [Online]. Available: https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7. [Accessed: 01-May-2020].

[15] G. Ognjanovski, "Everything you need to know about Neural Networks and Backpropagation - Machine Learning Made Easy...," Medium, 07-Feb-2019. [Online]. Available: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a. [Accessed: 01-May-2020].

[16] "DIY Heart Rate Sensor," Arduino Project Hub. [Online]. Available: https://create.arduino.cc/projecthub/Ingeimaks/diy-heart-rate-sensor-a96e89. [Accessed: 01-May-2020].

[17] "MPU-6050: TDK," InvenSense. [Online]. Available: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. [Accessed: 01-May-2020].

[18] T. Valencell, "Optical Heart Rate Monitoring: What You Need to Know," Valencell, 06-Apr-2018. [Online]. Available: https://valencell.com/blog/2015/10/optical-heart-rate-monitoring-what-you-need-to-know/. [Accessed: 01-May-2020].

[19] "Human?," *Are you a human?* [Online]. Available: https://www.newegg.com/p/37E-001G-00015. [Accessed: 01-May-2020].