IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), Bali, November 2021

Smart Monitoring System for Stroke Rehabilitation

Mais Al Atallah, Zainab Jamil, Hania Khafagy, Mehnaz Ummar, Fadi Aloul, Assim Sagahyroon
Department of Computer Science and Engineering
American University of Sharjah, UAE
{*g00075015, g00073453, g00074231, g00075564, faloul, asagahyroon*}@aus.edu

***Abstract*—Stroke survivors are vulnerable to post-stroke upper limb disabilities and physiotherapy is typically recommended to improve their movement. This paper introduces a patient-centered smart system where patients performing rehabilitation exercises can receive automated visuals of their improvements in the comfort of their homes. Moreover, the patient's health is also monitored based on their heart rate, recommendations regarding their improvement based on the exercises performed are provided and the patient's likelihood of stroke recurrence is predicted by the system. The system uses accelerometer and heart rate readings from a smartwatch along with readings from a stretch sensor attached to an exercise band. These readings are stored in the cloud and real-time databases, which are retrieved in the mobile application, where data is processed using algorithms to assess the improvement as well as generate recommendation and prediction models.**

***Keywords*—stroke, rehabilitation, pitch, machine learning**

I. Introduction

Stroke is a leading cause of death and disability in the world and occurs mostly due to clotting of an artery leading to the brain. Post-stroke patients tend to suffer from paralysis, weakness, awareness, learning, and memory [1]. A stroke can further cause loss of strength and feeling on one side of the body which can affect a stroke survivor's function.

One of the commonly prescribed stroke rehabilitation methods for patients with upper limb difficulties is using stretch bands to perform specific exercises regularly to improve and strengthen the affected muscles. However, evaluation of the patient's improvement when using these exercises is usually done by physiotherapists during a physical visit to a medical institution and using visual assessment. They then fill out different test scores such as the Functional Ability Scale (FAS) score to get a numerical score of the patient's improvement [2]. This traditional way is not only inefficient, time-consuming, and tedious for the patient but can also result in human error and hence, an automated approach to conduct rehabilitation is preferred.

To ease the rehabilitation process for stroke patients from the above-mentioned difficulties, this paper proposes an automated rehabilitation assistance in the form of a mobile application with the incorporation of smartwatch sensors and a stretch sensor system attached to their exercise band. The mobile application is designed to evaluate the patient's improvements, provide recommendations and give a prediction of stroke recurrence. This system will hence help the patient undergo rehabilitation at home with minimal assistance from a professional and allows health providers to remotely monitor the patient's health and hence their safety during the course of an exercise.

II. Literature Review

Stretch sensors have been utilized to evaluate patient movement as shown by Eschmann and Héroux [1]. They performed an observational study of the stroke patient's movements by placing two stretch sensors on both fingers. The data is sent from the stretch sensor to a mobile application using Bluetooth, and then the data is stored and analyzed. This paper utilizes stretch sensor values to visualize the patient's finger movements. With the increasing digitization of traditional disciplines, the use of wearable technology for remote patient monitoring is currently emerging. Hence, Lee et al. [3] and Patel et al. [4] papers positively reviews the use of wearable technology for rehabilitation. Lee et al. [3] propose using wearable sensors, one placed on the stroke-affected upper limb and the second on the other upper limb to recognize goal-directed movements in Activities of Daily Living (ADL). The wearable smartwatch has a built-in accelerometer and gyroscope which can be used to give feedback by measuring various metrics such as acceleration, velocity, displacement, and angular velocity. Whereas Patel et al. [4] address the different techniques used to incorporate sensors into worn textiles for rehabilitation purposes; the incorporation of inertial sensors such as accelerometers and gyroscopes for motion detection with textiles.

Accelerometer data is essential for monitoring stroke patient's movements. This data is acquired using an accelerometer sensor which is either attached externally or inbuilt within a smartwatch. Patel et al. [5] and Park et al. [6] both utilize an accelerometer for monitoring purposes. The work by Patel et al. [5] is a novel approach for assessing the quality of upper limb movements by providing an estimate of clinical scores obtained by using the FAS. The accuracy of the FAS scores obtained by analysis of the accelerometer data was compared to the actual FAS scores provided by a clinician. Uniaxial and biaxial accelerometers are placed on different aspects of the arm as shown in Fig. 1. Data segmentation and feature extraction is done on the accelerometer data and then Random Forests algorithm is used to train this data. This is done to estimate the FAS scores obtained for each motor task performed.

The work presented by Park et al. [6] comprises a wireless wristband device which is created to monitor the movement of the forearms. This device relies on an accelerometer that monitors the changes in acceleration created on the device while exercising, and consequently provides graphical analog feedback that is read and digitized by a custom Printed Circuit Board (PCB). This work also emphasizes the importance of accelerometers for measuring movement.

This paper seeks to contribute to the vast majority of existing literature by incorporating two separate aspects (stretch sensor system with the exercise band and smartwatch sensors) into one. The proposed rehabilitation system uses sensors to automate the process of measuring the patient's improvement; the sensors used are stretch sensors, accelerometer and heart rate sensor. The stretch sensor is placed on the elastic band, whereas the accelerometer is a built-in sensor in the smartwatch. The measurements from the sensors are stored in a database and eventually made use of by a mobile application, which stores and analyzes the readings to evaluate the progress and provide feedback and recommendations to the patient. The system also utilizes machine learning to predict the recurrence of another stroke.

III. Proposed System

The proposed system is visually depicted in the block diagram in Fig. 1. The stroke patient will be wearing a smartwatch and using the mobile application to perform exercises. The stretch sensor system, including the rubber resistive cord, and the NodeMCU ESP8266 chip are embedded together and will be attached on the TheraBand used to perform the rehabilitation exercises. Once the patient initiates the exercise service on the mobile application, the accelerometer and heart rate readings from the smartwatch are sent to the mobile phone via Bluetooth and then to the cloud database via Wi-Fi. Simultaneously, stretch sensor data (resistance of the stretch sensor) is sent to the real-time database, from the NodeMCU chip via Wi-Fi. The mobile device and the database undergo two-way communication in order to exchange raw sensory data, extracted feature sets and processed data.

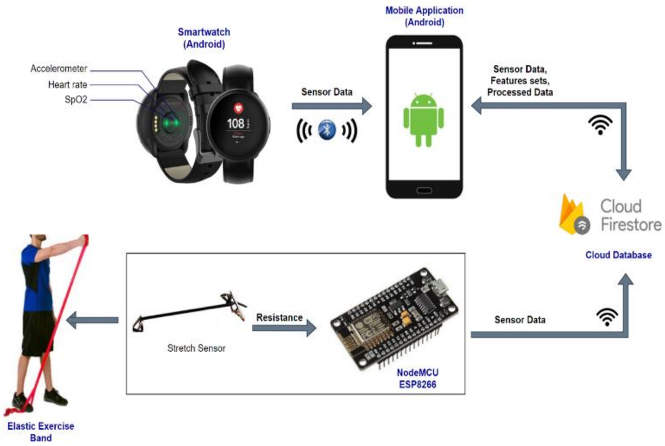

Fig. 1. Architectural layout of the system

The system allows three specific exercises: pull up, pull down and arm to side [7]. These exercises are prescribed to the patient with a certain number of repetitions, that is adjusted based on their improvement level during rehabilitation.

IV. System Implementation

A. Hardware System Implementation

The hardware component of the system included a 10 kΩ resistor for voltage division, wires, and the NodeMCU ESP8266 [8] chip connected to a power bank as shown in Fig. 2. The system uses a rubber cord stretch sensor to measure the stretch extent of the elastic exercise band; the extent of stretch is measured by calculating the resistance, i.e., the rubber cord's tendency to resist the flow of charge (current). A pair of crocodile wires are used to connect the cord to the NodeMCU chip; one wire connects a cord's end to the analog input port and the other to the ground. Additionally, the hardware system also comprises a Huawei Watch 2 smartwatch, from which sensor readings such as accelerometer data are extracted.

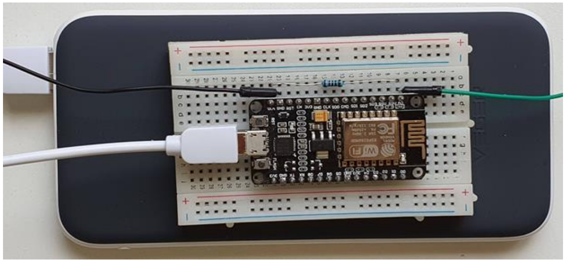

Fig. 2. Board connections for stretch sensor system

B. Data Collection

The NodeMCU chip of the stretch sensor system is programmed using Arduino IDE to read the resistance of the cord being stretched based on the below equations (1) and (2) where V_{in} and V_{out} are the input and output voltages of the chip, and R_{ref} is the reference resistor of 10 kΩ added as a voltage divider in the circuit. Hence, the program uses the voltage division equation (1) below to find the value of the conductive stretch cord's resistance (R_{cord}) as shown in (2).

$$V_{out} = \frac{V_{in} \times R_{cord}}{R_{ref} + R_{cord}} \quad (1)$$

$$R_{cord} = \frac{R_{ref}}{V_{in}/V_{out} - 1} \quad (2)$$

After finding the resistance value read from the analog input, it is pushed to the Firebase real-time database. The system records the value of resistance in kΩ every 3 seconds and an array of resistance values for a particular exercise are stored under the start timestamp of the exercise.

An Android smartwatch application is used to read and send the accelerometer and heart rate readings to Firestore using background processes. The readings from the sensor were read every 3 seconds and stored under a document of the start timestamp, to match the interval and labelling used for the NodeMCU readings.

C. Software Entities

1) Improvement Model

The improvement model is designed such that the patient can see graphical views of their improvement for a specific exercise throughout their rehabilitation. The stretch improvement is plotted based on the stretch sensor data retrieved from the Firebase real-time database, whereas the distance/speed improvement is plotted based on the accelerometer data retrieved from the Firestore database. The patient can choose whether to view the improvement overtime or the current improvement based on the exercise performed on that current day.

Improvement overtime: The improvement overtime graphs show the maximum stretch and maximum distance/speed (depending on the exercise chosen) that the patient has been able to achieve every day for that chosen exercise. If the exercise chosen is pull up, for example, the maximum stretch and maximum distance the patient has achieved in every pull up exercise performed will be displayed in a bar graph format, as shown in Fig. 3.

Current improvement: The patient can choose to view their current improvement of their last performed exercise. The current improvement is found by comparing the exercise performed on the specified day to the initial exercise performed on the first day and plotting both sets as line charts to show comparison as displayed in Fig. 3.

Fig. 3. Improvement model graphs

Before achieving the current improvement graphs shown in Fig. 3, both the accelerometer data and stretch sensor data must be processed.

Stretch sensor data: Fig. 4 shows the preprocessed stretch sensor data. This raw data is processed by finding the local maximum and minimum points in the graph. A local maximum is defined to be a point that is higher than the value before and the value after; a local minimum is a point that is lower than its adjacent points. After finding the local maximum and minimum points, the difference of each high peak and its following low trough represents the amount the cord has been stretched. The processed clean sets of data are shown in Fig. 5.

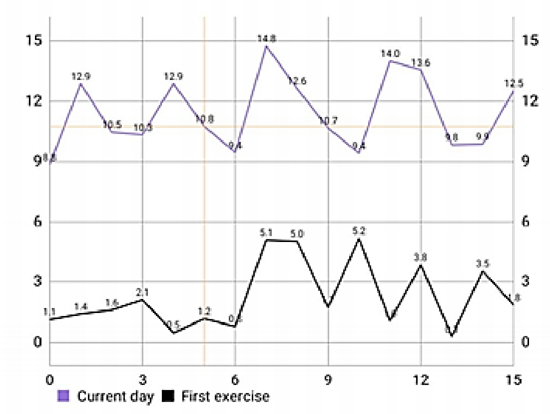

Fig. 4. Pre-processed stretch improvement data

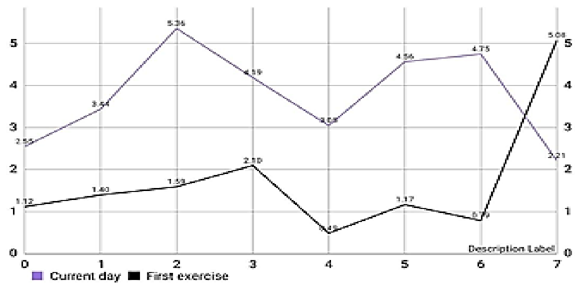

Fig. 5. Processed stretch improvement data

Accelerometer sensor data: Accelerometer readings are recorded and stored into the Cloud Firestore via the smartwatch that the patient wears. There are different values that can be extracted using these accelerometer values, such as velocity, position, angle rotation and so on. The raw accelerometer values are plotted in the graph shown in Fig. 68. These values mirror the patient's arm acceleration which includes speed and direction during their exercises. Another variable extracted from the accelerometer is the pitch angle which quantifies the arm's angle with the horizontal as shown in Fig. 7. This angle ranges from -90 degrees to +90 degrees. The -90 degrees indicates the arm to be completely downwards, whilst the +90 degrees indicates the arm to be completely upwards when performing the exercises. Hence, to monitor improvement using accelerometer values, the pitch angle was calculated from the accelerometer readings using the following equation:

$$pitch\ angle = \tan^{-1}\left(\frac{acc_x}{\sqrt{{acc_y}^2+{acc_z}^2}}\right) \tag{3}$$

The current day raw values of the pitch angle are plotted against the baseline pitch angle values for the three specific exercises. Next, like the stretch measurements, these raw pitch angle values are processed and the difference between the max and min angles are calculated to see the amount of angle changed by the patient. Both the current and baseline graphs are plotted against each other to visualize the improvement of the patient as shown in Fig. 7.

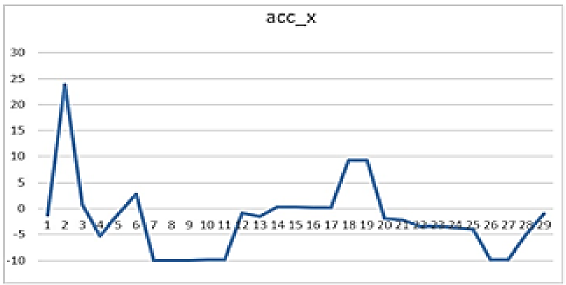

Fig. 6. Raw accelerometer values

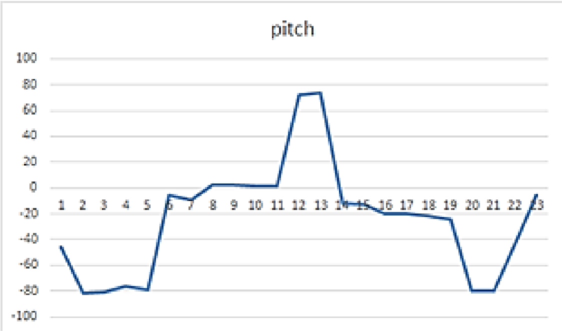

Fig. 7. Processed pitch angle values

2) Recommendation Model

A rule-based expert system is developed and used for the recommendation model. To monitor the patient's normal resting heart rate (nhr) rules were created based on the information and data provided by Ms. Margeaux Blignaut [8], the Stroke Rehabilitation Team Lead, and her team at Amana Healthcare [9]. As per the team's instructions, the patient is notified to use the system to record their heart rate at the beginning of the day before any food or medication to acquire their nhr. This is then checked against rules based on their initial nhr from the database, recorded during the pre-evaluation stage and also mapped to a provided scoring chart to ensure regular monitoring of the patient's heart health.

Based on the stroke rehabilitation team's approval, the rule-based engine to provide recommendations based on their performance in their respective exercises was built in such a way that if the patient is going beyond their capabilities and putting themselves at risk, they are advised to decrease the angle at which the band is pulled (if the patient exceeds their maximum possible pitch value set by the doctor) or decrease the amount of force applied when stretching the band (if the patient exceeds their maximum possible stretch value set by the doctor). In addition to this, if the patient is not stretching the band to their full potential based on their improvement level and their best stretch and pitch value achieved throughout their rehabilitation, they are advised accordingly. Finally, based on the mode of the stretch sensor resistances, if the mode or is not as high or close to their maximum obtained stretch, then the patient is informed that they are not maintaining their energy throughout the exercise and are advised to do so in order to maximize their improvement. Note that the physicians are provided with their own access to all patient records and can provide their own recommendations, when required.

3) Prediction Model

The stroke prediction dataset was obtained from 'kaggle' (an online platform for datasets) [10] . The original dataset contains 5110 records of different patients with 12 features; 'id', 'gender' (male or female), 'age' (in years), 'hypertension' (binary - if the patient has hypertension or not), 'heart_disease' (binary - if the patient has any heart disease(s) or not), ever_married (binary - if the patient was ever married or not), 'work_type', 'residence_type' (rural or urban), average glucose level (mg/dL), BMI (Body Mass Index in kg/m2), 'smoking_status' (if the patient does not smoke, has formerly smoked or is currently smoking) and finally the target feature, stroke (if the patient has suffered from a stroke or not). All the aforementioned features are numerical, except for gender, work type, ever married, residence type and smoking status, which require encoding as a method of data preprocessing. However, based on the heat map plot and further research and consultation with the Stroke Rehabilitation Lead at Amana Healthcare [8][9], the features with very low correlation with the target feature ('ever_married', 'work_type', 'residence_type' and 'gender') were dropped.

a) Class imbalance countermeasures

The original dataset contains 5110 records, 4861 belonging to patients who did not have a stroke while 249 belonging to stroke survivors. The clear and significant difference between both classes leads to an imbalance. Consequently, a measure to handle this imbalance is through the use of resampling techniques, such as over-sampling or under-sampling the majority class. A study conducted by V. Abedi et al. [11] to predict the recurrence of stroke using machine learning models, also used an imbalanced dataset with a majority class of non-recurrence of stroke and it was observed that the models performed better when undersampling the majority class. Additionally, when comparing both techniques, undersampling performed better for the dataset used for this system as well. Thus, random records from each age group of the majority class ('stroke' = 0) were dropped to form a more balanced dataset. This was the last stage of preprocessing the data resulting in a dataset of 3006 records with 7 features. To further enhance the performance of the model, Synthetic Minority Oversampling Technique (SMOTE) from the imblearn library, which generates new and synthetic minority training samples using a nearest neighbors algorithm, is used to further reduce the imbalance of the training dataset.

b) Selection of performance metrics

In order to carry out a fair comparison on the less imbalanced dataset produced, some evaluation measures need to be calculated, such as, precision, accuracy, recall (sensitivity), f1-score and AUROC curve [11]. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions, and it is important to ensure the false positives are minimal as they cause the highest cost for a model. Thus, solely depending on the accuracy as a measure to evaluate the algorithms is impractical.

The f1-score gives a better indication on the performance of the implemented algorithm, as it takes into account both recall and precision. Moreover, even with the resampling techniques applied, the dataset is still not completely balanced and when the number of samples in each class are unequal, the accuracy can be misleading in evaluating a machine learning algorithm, but better performance metrics are confusion matrix, precision, recall and f1-score and so our focus would be on these metrics [11]. Throughout the implemented algorithms a test size of 20% was maintained, which is the ideal proportion of testing to training percentages [11].

c) Training and testing

In order to select the most suitable training algorithm, five machine learning (ML) algorithms were implemented. The selected ML algorithms were k-Nearest Neigbors (k-NN), Support Vector Classifier (SVC), Logistic Regression (LR), Random Forest (RF) and Decision Tree (DT) using the scikit-learn library [12]. For the decision tree, the main parameters tuned were the 'max_depth', 'min_samples_split' and 'max_leaf_nodes'. The max_depth parameter decides how deep the tree can get, the deeper the tree the greater the number of splits the higher the chance of overfitting. While the min_samples_split parameter indicates the minimum number of samples in a node required to split the node. Lastly, the max_leaf_nodes parameter, which indicates the maximum number of leaf nodes, also known as terminal nodes and these nodes represent class labels.

Regardless of the hyperparameters, the effect of importing Synthetic Minority Oversampling Technique (SMOTE), on the model was experimented. As aforementioned, since the dataset labels are imbalanced some measures need to be taken in order to provide enough representation for 'stroke' and 'no stroke' records. SMOTE, is an oversampling technique that works by creating synthetic samples for the minority class, in our case it is the stroke patients class ('stroke' = 1). Another solution experimented was changing the class weights, which works by increasing the weight of the minority class and decreasing the weight of the majority class, while maintaining a specified threshold in order to prevent bias towards the minority class. However, SMOTE was observed to have performed better compared to modifying the class weights.

d) Deployment

Consequently, to avoid re-training the model every time a run occurs, the model is saved and restored using the pickle module in sklearn. The model was then easily loaded back and then was deployed onto the web using the Flask framework and ngrok. The Flask framework helps us to web-enable the python model via an HTTPS local URL and ngrok basically creates the server to run our model on the internet and hosts it to the public internet over secure tunnels without the requirement of any public domain or IP address. Using the URL generated by ngrok, we can acquire the prediction result for a patient based on their data, as a JSON object. The mobile application uses the patient's required fields from their 'Profile_data' collection in Firebase and gets the prediction result from the deployed model as a JSON object with the help of Volley API, which is then parsed. If the patient navigates to the 'view prediction' activity, based on the prediction result, a message is first printed out to the patient stating whether they are likely or not to suffer from another stroke and their risk factors are printed along with recommendations based on them generated using a rule-based expert system.

V. RESULTS AND EVALUATION

In a 'relaxed' state, the resistance of the stretch sensor is about 350-400 Ω per inch. Therefore, before integrating the sensor in the system, the length of the cord and its resistance value in the relaxed state must be recorded and compared with the theoretical value, which should be between 350*length and 400*length. As the cord gets stretched, the resistance increases because the particles get further apart. The cord used in the system is 19 inches long. Hence, the expected resistance range of the cord in its relaxed state should be in the range of 6.65-7.60 kΩ, while the experimental value measured is 5.306428 kΩ. It is important to know the length of the cord used before integrating it in the system because it can only be stretched about 50-70% longer than the resting length; hence, if a 19" cord is stretched more than more than 28.8", there is a risk the sensor gets damaged.

The smartwatch application used background processes, leaving the smartwatch unaffected. Since the smartwatch sensor readings were sent to the cloud database (Firestore), the readings were not sent in real-time but the document as a whole was uploaded after the completion of an exercise, which took on average 3 minutes to be uploaded onto the database from the smartwatch, via Bluetooth and WiFi. This in-turn affected the response time of the services on our mobile app that use the smartwatch readings, if the patient chose any of those services right after performing the exercise.

For the prediction model, as mentioned all five ML algorithms were tested and produced the results listed in table 1. Firstly, the k-NN algorithm under-represented the positive stroke records. Further, the SVC is of high computational complexity and thus, was not ideal for this problem with a moderately large dataset. LR resulted in a significant number of false negatives and false positives. Finally, RF's performance is slightly higher than DT's but may be slowed down with increasing number of trees as the number of records increases. The chosen ML model is using decision tree classification, which is a non-parametric supervised learning algorithm. The reason we chose this algorithm is because it maximizes the information gain. Additionally, it is relatively faster than aforementioned algorithms in classifying unknown records and more efficient in training datasets with categorical attributes. Furthermore, it was also observed in the tabulated results of [11], that the highest performance metrics were for that of the DT model using under-sampling.

As an evaluation measure of the algorithm's performance the Area Under the Receiver Operating Characteristics (AUROC) curve was plotted, it basically gives an indication on the model's ability to distinguish between classes. The higher the AUC probability, the better the model. The model showed promising results with an area under the AUROC curve of 0.672 as shown in Fig 8.

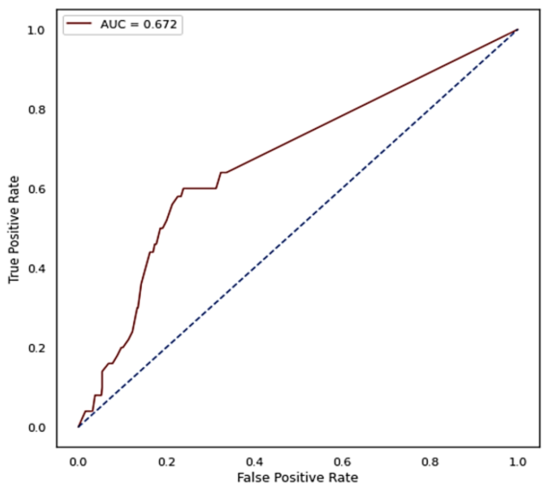

Fig. 8. AUROC curve of stroke prediction model

TABLE I. CLASSIFIER RESULTS FOR ALL ALGORITHMS

Algorithm	**Classifier Results for All Tested Algorithms**			
	Weighted precision	***Weighted recall***	***Weighted f1-score***	***Accuracy***
k-NN	84%	90%	87%	90%
SVC	83%	91%	87%	91%
LR	87%	69%	75%	69%
RF	92%	88%	90%	88%
DT	86%	84%	85%	84%

The proposed system was tested using 2 test subjects. Both the test subjects were females, age range of 20-25 years, and no pre-existing condition. The accuracy of the data collected was verified by visually mapping the subjects arm movements with the pitch angle and stretch values obtained from the system. Then, two medical profiles were created for each subject, and they began exercising with different levels to check the graphical views and recommendations provided, which produced positive results. These positive results indicate that appropriate graphs, recommendations, and prediction was displayed to the subjects on the mobile app based on their condition and the exercises that they performed.

VI. CONCLUSION

This paper attempts to help stroke patients performing rehabilitation. Stretching elastic bands is a common method used in stroke rehabilitation to strengthen the patient's muscles. The main objective highlighted is to build a mobile application to automate the process of rehabilitation by using the inbuilt sensors in the smartwatch, such as the accelerometer and heart rate sensors along with the stretch sensor attached to the elastic exercise band. All the sensor readings are used to measure the patients' improvements and provide recommendations and predictions to help them during rehabilitation without direct doctor intervention. Based on the tests executed on the system using the subjects, the overall system performance is satisfactory as it gives expected results.

A potential extension of the paper is to tackle more stretch band exercises that include the rotation of the arm. Such exercises require measuring the rate of rotation done during the exercise, and hence it requires using an additional sensor such as a gyroscope. Another extension would be to test the proposed system with additional subjects for further accuracy.

ACKNOWLEDGMENT

This research project was funded by the American University of Sharjah, Undergraduate Research Grant, URG21-013. The team would like to acknowledge the contribution and guidance received from the coadvisors of this project, Dr. Assim Sagahyroon and Dr. Fadi Aloul. The team would also like to acknowledge Dr. Raafat AbuRukba and Mr. Jayroop Ramesh for providing the necessary application to retrieve sensory data from the smartwatch and store it in the cloud database, in addition to Mr. Ahmad Nabulsi for assisting with the circuit setup. Lastly, the team acknowledges the help and guidance received from Amana Healthcare from Healthpoint Hospital in answering the queries that arose [9].

REFERENCES

[1] H, Eschmann, M. Héroux, J. Cheetham, S. Potts, and J. Diong. “Thumb and Finger Movement is Reduced After Stroke: An Observational Study”, Plos One, vol. 14, no. 6, 2019. [Online]. Available: ProQuest, https://doi.org/10.1371/journal.pone.0217969

[2] S. I. Lee, C. Adans-Dester, A. O'Brien, G. Vergara, R. M. Black-Schaffer, R. Zafonte, J. Dy, and P. Bonato, “Predicting and Monitoring Upper-Limb Rehabilitation Outcomes Using Clinical and Wearable Sensor Data in Brain Injury Survivors”, IEEE Transactions on Biomedical Engineering, 2020. [Online]. Available: IEEE Xplore, https://doi.org/10.1109/TBME.2020.3027853

[3] S. Lee, C. Adnas-Dester, M. Grimaldi, A. Dowling, P. Horak, R. Black-Schaffer, P. Bonato, J. Gwin S. “Enabling Stroke Rehabilitation in Home and Community Settings: A Wearable Sensor-Based Approach for Upper-Limb Motor Training”, IEEE Journal of Translational Engineering in Health and Medicine, vol. 6, 2018, pp. 2. [Online]. Available: IEEE Xplore, https://doi.org/10.1109/JTEHM.2018.2829208

[4] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers. “A Review of Wearable Sensors and Systems with Application in Rehabilitation”, Journal of NeuroEngineering Rehabilitation, vol. 9, no. 21, 2012. [Online]. Available: Journal of NeuroEngineering and Rehabilitation, https://doi.org/10.1186/1743-0003-9-21

[5] S. Patel, R. Hughes, T. Hester, J. Stein, M. Akay, J. Dy, P. Bonato, “A Novel Approach to Monitor Rehabilitation Outcomes in Stroke Survivors Using Wearable Technology”, Proceedings of the IEEE, vol. 98, no. 3, pp. 450-461, 2010. [Online]. Available: IEEE Xplore, https://doi.org/10.1109/JPROC.2009.2038727

[6] J. Park, S. Robinovitch, and W. S. Kim. “A Wireless Wristband Accelerometer for Monitoring of Rubber Band Exercises”, IEEE Sensors Journal, vol. 16, no. 5, pp. 1143-1150, 2016. [Online]. Available: IEEE Xplore, https://doi.org/10.1109/JSEN.2015.2499158

[7] C. Stinear, and S. Ackerley. “PREP Upper Limb Rehab Exercise Booklet”. The University of Auckland, 2015. [Online]. Available: ResearchGate, https://doi.org/10.13140/RG.2.1.3780.8489

[8] M. Blignaut, Linkedin, 2021. [Online]. Available: https://www.linkedin.com/in/margeaux-blignaut-3a5983a9/?originalSubdomain=ae.

[9] Amana Healthcare, 2021. [Online]. Available: https://www.amanahealthcare.com/en/

[10] "Stroke Prediction Dataset", Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset/version/1

[11] V. Abedi, V. Avula, D. Chaudhary, S. Shahjouei, A. Khan, C. J. Griessenauer, J. Li, and R. Zand, “Prediction of Long-Term Stroke Recurrence Using Machine Learning Models,” Journal of Clinical Medicine, vol. 10, no. 6, p. 1286, Mar. 2021.

[12] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", Jmlr.csail.mit.edu, 2021. [Online]. Available: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.