Activity Recognition Using IoT and Machine Learning

Johanna Olnén and Julia Sommarlund

***Abstract*—Internet of Things devices, such as smartphones and smartwatches, are currently becoming widely accessible and progressively advanced. As the use of these devices steadily increases, so does the access to large amounts of sensory data. In this project, we developed a system that recognizes certain activities by applying a linear classifier machine learning model to a data set consisting of examples extracted from accelerometer sensor data. We obtained the data set by collecting data from a mobile device while performing commonplace everyday activities. These activities include walking, standing, driving, and riding the subway. The raw accelerometer data was then aggregated into data points, consisting of several informative features. The complete data set was subsequently split into 80% training data and 20% test data. A machine learning algorithm, in this case, a support vector machine, was presented with the training data set and finally classified all test data with a precision higher than 90%. Hence, meeting our set objective to build a service with a correct classification score of over 90%.**

Human activity recognition has a large area of application, including improved health-related recommendations and a more efficiently engineered system for public transportation.

***Sammanfattning*—Internet of Things-enheter, så som smarta telefoner och klockor, blir numera allt mer tillgängliga och tekniskt avancerade. Eftersom användningen av dessa smarta enheter stadigt ökar, ökar också tillgången till stora mängder data från sensorer i dessa enheter. I detta projekt utvecklade vi ett system som känner igen vissa aktiviteter genom att tillämpa en linjär klassificerande maskininlärningsmodell på en uppsättning data som extraherats från en accelerometer, en sensor i en smart telefon. Datauppsättningen skapades genom att samla in data från en smart telefon medan vi utförde vardagliga aktiviteter, så som promenader, stå stilla, köra bil och åka tunnelbana. Rå accelerometerdata samlades in och gjordes om till datavektorer innehållandes statistiska mått. Den totala datauppsättningen delades sedan upp i 80% träningsdata och 20% testdata. En maskininlärningsalgoritm, i detta fall en supportvektormaskin, introducerades med träningsdatan och klassificerade slutligen testdatan med en precision på över 90%. Därmed uppfylldes vårt uppsatta mål med att bygga en tjänst med en korrekt klassificering på över 90%. Igenkänning av mänsklig aktivitet har ett stort användningsområde, och kan bidra till förbättrade hälsorekommendationer och en mer effektiv kollektivtrafik.**

***Index Terms*—Activity recognition, sensors, smartphone, data processing, accelerometer**

Supervisors: Carlo Fischione, Hossein Shokri Ghadikolaei

TRITA number: TRITA-EECS-EX-2020:153

I. Introduction

Human activity recognition can be defined as the study of movements and activities of a singular person, or a larger population, using machine learning analysis on data collected from sensors. The data may either be collected from videos, images, or directly from sensors embedded in smart devices [1].

Activity recognition requires large amounts of data, which has created obstacles in the past. The collection of data has previously been difficult and expensive, as it initially required custom hardware with specific sensors. As the use of Internet of Things devices, such as smartphones and smartwatches, steadily increases, so does the access to large amounts of sensory data. The devices incorporate various sensors, including accelerometers (acceleration sensor), gyroscopes (Angular Velocity Sensor), and GPS sensors (location sensor). This has increased the interest in using smartphone-based sensors to build valuable applications, in this case being human activity recognition. [1]

Activity data can be analyzed using machine learning. A machine learning algorithm creates a mathematical model based on data which can make predictions or classifications depending on parameters of the initial data set. The chosen algorithm for this project is a support vector machine, as described more in detail in Section V. [2]

Human activity recognition has a large area of application. By tracking one individual's, or an entire population's movements, an extensive understanding of the public's regular physical activity can be gained, and lead to improved health-related recommendations. A better understanding of a population's movements can also be used to improve municipal travel and lead to more efficient public transportation [2].

A. Project description

The project aims to develop a service that can recognize certain human activities, which includes developing a data set used to train and test the activity recognition service. The goal is to build a service with a correct classification score of over 90% for the following activities: standing, walking, driving, and riding the subway.

II. Data collection

A. Method

The data set was obtained using an application on our smartphones to collect 3-axial linear acceleration, using the embedded accelerometer sampled at the frequency of 50 Hz. A location sensor was also used to determine the position when gathering data, sampled at the best possible accuracy.

We assessed that it was enough with accelerometer data to recognize our chosen activities, and when collecting the data we did not consider the orientation of the device.

The data was collected one activity at a time and by one person. To ensure that the label of the conducted activity was as accurate as possible, we paused the data collection when the subway or car reached a stop. With this approach, each created file only included accurate and clean data of the corresponding activity.

B. Limitations

As a result of the COVID-19 situation, the regulations and general guidelines from the Public Health Agency prevented the collection of data for public transportation. After discussions with our supervisor, we decided to continue with the existing data set and exclude bus transportation since we had not collected that particular data at the time. This caused the number of data points for subway transportation to be the limiting factor in the data set.

C. The Activities

The performed activities in this experiment included walking, standing, driving and riding the subway. The choice of activities was based on the fact that they are common among peoples daily routines, and most often performed for longer time periods, thus simplifying the collection of data. The raw accelerometer data associated with all performed activities, with the x, y, and z values are plotted in Figure 1.

As one observes the changed values in acceleration for the activities in Figure 1, it shows that the associated values of acceleration exhibit behaviors that are coherent with the presumed motions.

Stationary activities, standing in this case, are expected to have relatively constant acceleration values and no periodic behavior. One can recognize this property in the graph for standing. The graph for walking shows that the activity involves periodic motion, which would be expected. The x, y, and z axis all vary periodically with high peaks.

The data for driving and subway transportation are comparable. The subway plot, however, shows larger magnitudes and higher peaks of the acceleration value. Regular stops occur while traveling by subway, leading to frequent both acceleration and deceleration, which explain the periodic peaks.

We drove on a highway while collecting the data for car transportation, resulting in a near-constant speed. Plots over the acceleration data for driving consequently shows a minimum variation in the acceleration values.

III. Data Processing

The support vector machine, described in Section V, uses examples of data to make classification decisions. The raw accelerometer data was accordingly processed and aggregated into data points consisting of informative features.

A. Signal Pre-processing

The accelerometer signal consists of two main components: gravitational acceleration and body acceleration. To separate the raw accelerometer data into these components we first

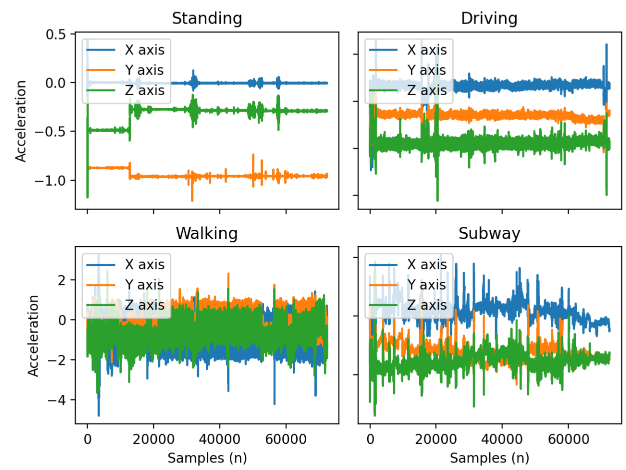

Fig. 1. Plots of the raw 3-axial linear acceleration collected at 50 Hz, for the conducted activities. Every sample included in the final data set is visualized.

had to assume that the gravitational force only had low-frequency components. A low-pass Butterworth filter with a 0.3 Hz cutoff frequency was subsequently applied to the accelerometer signal to be able to distinguish between body and gravity acceleration. The divided accelerometer signal was then split into fixed-width frames with a length of 2.56 s, and with a 50% overlap.

The values for the signal processing were selected based on the data processing of a similar public data set [3].

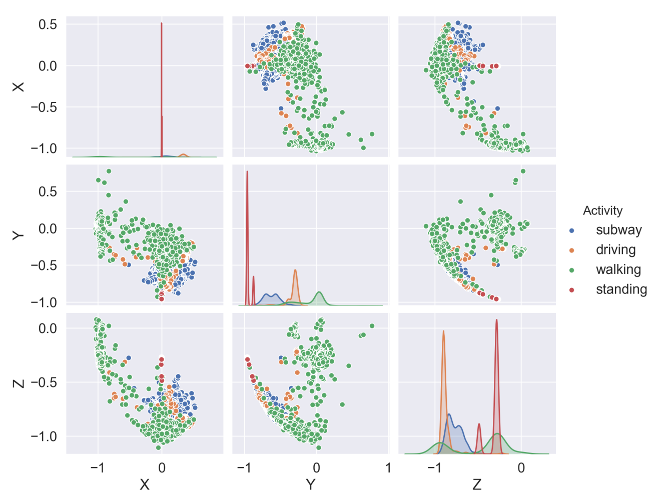

Fig. 2. Pair plot of the total 3-axial linear acceleration for the conducted activities.

Figure 2 displays a pair plot of the average total acceleration, for each of the four activities. A pair plot visualizes the distribution of individual features, as well as the correlation between features. This way one can observe which activities are the most separable.

B. Feature vectors

When the two acceleration signals were divided into 2.56-second segments, we next extracted features based on these 128 samples. The features described below were generated for each axis, using MATLAB, with the code attached in Appendix A and available on Github [4]. The variables are defined as:

μ = the mean value of the observations
n = the number of observations
x_i = each observed value.

- Average: mean value of the acceleration for each frame.
- Standard deviation (std): a measurement of the distribution in every frame.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2} \tag{1}$$

  Equation (1) is the definition of standard deviation.

- Skewness: measure of the asymmetry of each frame.

$$Sk = \frac{n\sqrt{n-1}}{n-2}\frac{\sum_{i=1}^{n}(x_i-\mu)^3}{(\sum_{i=1}^{n}(x_i-\mu)^2)^{3/2}} \tag{2}$$

  Equation (2) is the definition of skewness.

- Kurtosis: measure of the weight of the tails relative to a normal distribution.

$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4}{[\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2]^2} \tag{3}$$

  Equation (3) is the definition of Kurtosis.

IV. Data set structure

The data set, presented in Table I, consists of the resulting examples from the process described in Section III. The number of data points associated with each activity was set to the same value, determined by the number of data points collected from subway transportation, thereby producing a data set with 4 520 data points.

TABLE I
Data set format

Id	body_acc_mean_X	...	activity	activity_name
0	0.0071	...	5.0	subway
1	-0.0292	...	5.0	subway
2	-0.1653	...	5.0	subway
...	...	...	...	...
4519	-0.0045	...	1.0	standing

Every example is generated from a window of readings, and contains the 24 features listed below:

1) body_acc_mean_X
2) body_acc_std_X
3) body_acc_skewness_X
4) body_acc_kurtosis_X
5) body_acc_mean_Y
6) body_acc_std_Y
7) body_acc_skewness_Y
8) body_acc_kurtosis_Y
9) body_acc_mean_Z
10) body_acc_std_Z
11) body_acc_skewness_Z
12) body_acc_kurtosis_Z
13) total_acc_mean_X
14) total_acc_std_X
15) total_acc_skewness_X
16) total_acc_kurtosis_X
17) total_acc_mean_Y
18) total_acc_std_Y
19) total_acc_skewness_Y
20) total_acc_kurtosis_Y
21) total_acc_mean_Z
22) total_acc_std_Z
23) total_acc_skewness_Z
24) total_acc_kurtosis_Z

V. Machine learning algorithm

A machine learning algorithm builds a mathematical model and framework with the ability to make decisions or predictions based on data. Different machine learning models use different types of learning, supervised learning, unsupervised learning, and reinforcement learning, for example [2].

A supervised learning model is presented with the correct labels or categories, while unsupervised learning does not learn any desired output. The unsupervised learning model is assumed to find the structure of the mathematical model autonomously. Reinforcement learning lets the model interact with a dynamic environment, with the intent of completing a task. The model is then presented with reflective feedback, based on the approach, in order to interpret and learn from it.

The learning model that was best-suited for our project resulted in being supervised learning, as the process of labeling our data set was simple. Next, there are several different algorithms in the category of supervised learning. The choice of algorithm depends on the size and format of the data set. Our complete data set was small, consisting of only 4 520 data points, suggesting that a short calculation time is secondary. However, considering that the data set contained 24 features, it was advantageous with an algorithm that remains accurate in high dimensional spaces. With this in consideration, the algorithm of choice was a support vector machine.

A. Theory

The support vector machine (SVM) we chose is a linear classifier machine learning model that uses the principle of supervised learning. A linear classifier is a model that makes classification decisions based on a linear combination of different features connected to a data point. These features are presented as a feature vector. When a labeled data set is presented to a support vector machine, the machine builds a model that separates data points into different categories to a hyperplane, aiming to separate the different categories to the largest extent possible. The dimension of the hyperplane is one less than the number of features in each feature vector.

The separation between categories is measured in the margin, and the SVM aims to have the largest possible margin. The SVM uses a method called soft margin to obtain the best classification result. The concept of soft margin is that the mathematical model will allow for some discrepancies and wrong classifications in the training data set, to be able to keep the largest margin possible between the categories. The error in the vectors is measured in the error norm. The error norm is described with the variable ξ in Equation (4).

$$y_i(+b) \geq 1 - \xi_i \tag{4}$$

The classifier then maximizes the margin by minimizing Equation (5), where y_i is the i-th category and $w \cdot x_i - b$ is the current output. The absolute value of w is the absolute value of the normal vector to the hyperplane separating the categories, and C is a constant [5].

$$min\left[\frac{1}{2}||\ \mathbf{w}\ ||^2 + C\sum_{i=1}^{n}\xi_i\right] \tag{5}$$

There are different ways to evaluate a classifier. A confusion matrix is a table that shows the performance of a classifier by presenting the number of categories that were classified correctly, and what the wrongly classified data points were classified as. Other ways to evaluate performance include precision, recall, F_1-score, and accuracy. Precision is defined as the amount of correct, or relevant classifications, compared to the total number of classifications, while recall is the total number of correct classifications that were made. The F_1-score takes both precision and recall into account and gives an overall accuracy for the classifier. The F_1-score is calculated as Equation (6) and is presented as a decimal number between 0 and 1, where 1 is the best score.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6}$$

B. Implementation

The implementation of the support vector machine was made in Python. The model was imported from Scikit-learn, a machine learning library containing multiple algorithms. The data set was split into 80% training data and 20% test data. Every activity was split evenly to ensure that the test and training data included the same amount of data points from each activity. We also wanted to preserve the correct consecutive order of samples in the test data, meaning that if one data point is from a certain part of a time-series for driving data, then the next data point should also be from the same series of consecutive samples. To accomplish this, we solely shuffled the training data. Although the accuracy in one sample may be low, we know that the following samples are from the same class and the classifier will hopefully make the right decision in the majority of cases.

The model was then trained with the labeled training data. The implementation is presented in appendix B and on Github [4].

VI. Results

The chosen algorithm was presented with the test data set and its performance was evaluated. The performance is illustrated by the confusion matrix presented in Figure 3, and the normalized confusion matrix in Figure 4. The matrix has been normalized with the class support size, which is the number of elements in each class. Figure 3 shows the total amount of both correctly and wrongly classified data points for each activity, and what the wrongly classified data points were classified as. Figure 4 presents the amount of correctly classified data points for each activity in the diagonal boxes, represented with a decimal number between 0 and 1, where 1 equals a correct classification score of 100%.

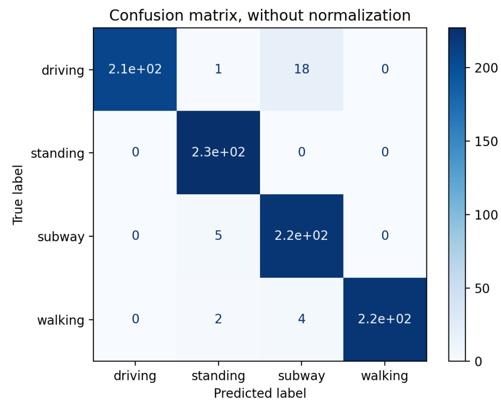

Fig. 3. Confusion matrix presenting the performance of the service.

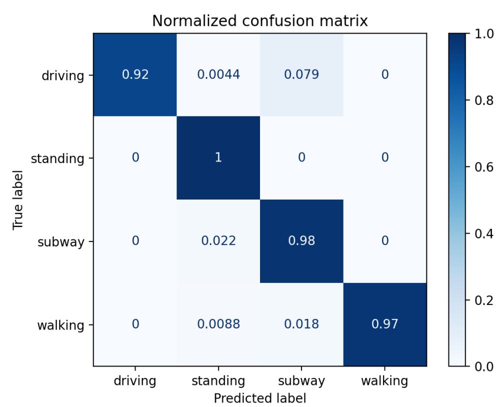

Fig. 4. Normalized confusion matrix presenting the performance of the service, normalized by the class support size

A summary of the results are also illustrated in the classification report presented in Table II. The table includes precision, recall, F_1-score and support.

VII. Related work

In the initial phase of the project, we reviewed different studies that had previously been done on human activity recognition. We examined and compared how the different projects had conducted their collection of data, their way of

TABLE II
CLASSIFICATION REPORT

	Precision	**Recall**	**F_1-score**	**Support**
Driving	1.00	0.92	0.96	226
Standing	0.97	1.0	0.98	226
Subway	0.91	0.98	0.94	226
Walking	1.00	0.97	0.99	226
Accuracy	-	-	0.97	904
Macro avg	0.97	0.97	0.97	904
Weighted avg	0.97	0.97	0.97	904

processing data, as well as what machine learning algorithm they used to make classifications.

The projects we examined, Chevalier [6], Bhaskar [7] and Kwapisz, Weiss and Moore [8], involve different activities than ours, which include sitting, standing, lying, walking, walking upstairs, walking downstairs, jogging. All projects use phone-based accelerometer sensors, and Chevalier and Bhaskar also used gyroscope sensors for the collection of data. As mentioned in Section II, we chose to solely use the accelerometer like Kwapisz et al. This decision was based on the fact that our activities differ more in acceleration data, compared to Chevalier and Bhaskar, and do not require information regarding the orientation of the device to differentiate between them. The studied projects used different frequencies, Bhaskar and Chevalier used frequency of 50 Hz while Kwapisz et al. used 20 Hz. We decided to use 50 Hz.

We also looked at the data processing of the projects. Both Chevalier and Bhaskar applied a noise-filter, sampled fixed-width sliding windows of 2.56 seconds with a 50% overlap. They also used a Butterworth-filter with a cut-off frequency of 0.3 Hz to separate the gravitational force and body acceleration. This process is the same we used on our data.

Unlike Bhaskar and Chevalier, Kwapisz et al. did not process the data with filters, instead they divided the data into 10 second segments and extracted features directly from the segments. Kwapisz et al. extracted a total of 43 different features based on statistical measurements for each axis from the segments. These included average acceleration, standard deviation, average absolute difference, average resultant acceleration, time between peaks as well as binned distribution. Like Kwapisz et al., we decided to use feature vectors as well, but fewer features and based on other measuerments. We used 24 features instead of 43 and we took measurements such as skewness and kurtosis into account.

As a classifier algorithm, Chevalier uses a recurrent neural network with long short-term memory cells, which is a neural network with feedback connections. This type of algorithm does not require the data to be in the form of feature vectors but requires knowledge and understanding of neural networks. Kwapisz et al. evaluates three different methods, decision tree, logistic regression as well as multilayer neural network. The best results was achieved with decision tree. Bhaskar tries different algorithms, including the one Chevalier uses, as well as a support vector classifier and a neural network with one-dimensional convolution. He also explores a divide and conquer algorithm for one-dimensional convolution, where activities are first divided into dynamic and static classes and before using the neural network. The best accuracy Bhaskar achieved with the divide and conquer method, followed by a support vector classifier. Due to our limited knowledge of machine learning, the classifier of choice was a support vector machine, as described more in detail in Section V.

VIII. DISCUSSION

A. *Result analysis*

As shown in the confusion matrix in Figure 3, and the normalized confusion matrix in Figure 4, the correct classified activities for each category are all above 90%. Driving had the lowest score of correct classification, with a score of 92%. Walking and subway had high scores of correct classification, 97%, and 98%. Standing had a score of classification at 100%, which means the model classified all data within the activity correctly.

Additionally, table II shows that the F_1-score is high for all activities. Subway, being the only activity that has an F_1-score below 0.95, has a score of 0.94, while driving, standing, and walking has scores of 0.96, 0.98, and 0.99 respectively. The deviations from optimal results can have different explanations. One cause could be that we did not perform any cleaning of data after it was collected. Another possible cause is that the data set only consists of data from one sensor, the accelerometer. Figure 3 shows that 18 data points that are recorded while driving a car are classified as riding the subway. This can be explained by the two activities having similar accelerations, and the model would probably need more data from other sensors to correctly classify the two activities. The overall high results can be explained by a relatively clean data set, considering the precision during the data collection, and well-chosen processing of the data.

Since our data collection came to an abrupt stop due to the outbreak of COVID-19, we did not have the chance to include bus transportation in the data set. If it would have been included, we believe more sensory data, such as gyroscope or GPS, would have been needed to separate car and bus transportation. Assuming that they would present similar acceleration data, it could become difficult to separate the two activities. As the data collection was smaller than expected, the importance of processing the data in an informative way was necessary in order to still get satisfactory results.

The service can be used for different areas of application. As we mentioned earlier, it can be used to map a population's movements, both in regards to public transportation and daily physical activities. By gaining this information, the system for public transportation has the possibility of forming an accurately planned and more efficient infrastructure. The service can also be used to map connections between movement and different diseases, such as high blood pressure, cardiovascular diseases, obesity, among others. This can lead to improved health-recommendations and better treatment of the mentioned diseases.

B. Future work

The service we created could be further developed in different ways. Activities such as riding the bus, running, or cycling can be added to the service, which would make the service present a more holistic perspective over a subject's movements and activities.

The data for the existing activities could also be collected more extensively, using more people to collect data on several occasions. This would create a more diverse data set with better accuracy to reflect a realistic vision of the activities. The data for driving, in particular, would improve by including various traffic situations, considering that the activity varies based on different environments. Driving on a highway is, for example, very different from driving in an urban environment.

Another improvement that could be implemented to the service is to utilize the GPS sensor embedded in smartphones to record the location of the subject. If location data is added to the data set, it would be possible to integrate the open-source, public transportation API to specify which bus or which subway line the subject is using.

IX. Conclusion

As the access to sensory data is increasing, so is the use of studying human activity recognition. By using the sensors built in a smartphone, we created a data set for four different activities, to be classified using a machine learning algorithm. This service can provide useful knowledge about the habits of a large amount of users, solely by accessing data from embedded sensors in a smart-phone. The result of the project meets the set goals for the project, with an overall accuracy of above 90%. The satisfactory classification results can be explained by a relatively clean and precise data set, and well-chosen processing of the data. The service can be improved by increasing the number of activities, developing and expanding the current data set, and by adding additional sensory data to be able to map movements within communal transportation.

Appendix

Appendix A - Data processing in Matlab

Appendix B - Implementation of support vector machine in Python

Acknowledgment

We would like to thank our supervisor Hossein Shokri Ghadikolaei for always being available for questions and guidance, as well as Carlo Fischione for raising important questions and encouraging discussions about the project's impact on society.

References

[1] M. Hassan, S. Huda, M. Uddin, A. Almogren, and M. Alrubaian, "Human activity recognition from body sensor data using deep learning," *Journal of Medical Systems*, vol. 42, no. 6, pp. 1–8, April. 2018.

[2] O. Reyes and L. Jorge, *Smartphone-Based Human Activity Recognition*, 1st ed., ser. Springer Theses, Recognizing Outstanding Ph.D. Research, London, 2015.

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. (2013, Apr.) A public domain dataset for human activity recognition using smartphones. UC Irvine Machine Learning Repository. Bruges, Belgium. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Human%20Activity%20Recognition%20Using%20Smartphones

[4] J. Olnén and J. Sommarlund. (2020, May.) Activity recognition. GitHub. Stockholm, Sweden. [Online]. Available: https://github.com/juliasommarlund/activityrecognition

[5] C. Campbell, *Learning with support vector machines*, ser. Synthesis digital library of engineering and computer science. San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA): Morgan Claypool, 2011.

[6] C. Guillame. (2016, Mar.) Lstms for human activity recognition. GitHub. [Online]. Available: https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition?fbclid=IwAR3KrWrSXFVo_rAtx8lVxrstVthdO-G4izXxv4CIcoz5twtDeg7G3otMORM

[7] B. Udi. (2020, Jan.) Human activity recognition using deep learning. GitHub. Seoul, Korea. [Online]. Available: https://github.com/UdiBhaskar/Human-Activity-Recognition--Using-Deep-NN

[8] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.