A Comparison of Deep Learning Models in Human Activity Recognition and Behavioural Prediction on the MHEALTH Dataset

Jamie O'Halloran[1], and Edward Curry[2]
j.ohalloran1994@gmail.com, and edward.curry@insight-centre.org
Data Science Institute,
National University of Ireland Galway, Galway, Ireland

Abstract. The problem of classifying body gesture and motion along with aiming to predict states of action or behaviour during physical activity is refered to as Human Activity Recongition (HAR). Inertial Measurement Units (IMUs) prevail as the key technique to measure range of motion, speed, velocity and magnetic field orientation during these physical activities. On-body inertial sensors can be used to generate body motion and vital signs recording signals that can successfully learn models and accurately classify physical activities. In this paper, we compare the approaches of Extreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), CNN + LSTM Hybrid (ConvLSTM) and Autoencoder by Random Forest (AE w/ RF) to classify human activities on the MHEALTH dataset. All six of our classification models use raw, unstructured data obtained from 4 inertial on-body sensors. We examine multiple physical activities and on-body inertial sensors, showing how body motion and vital signs recordings can be modified to be fed into machine learning models using diverse network architectures. We also compare the performance of the machine learning models to analyse which model best suits multisensory fusion analysis. The experimental results of this paper on the MHEALTH dataset consisting of 12 physical activities collected from 10 subjects with the use of four differnet inertial sensors, are highly encouraging and consistently outperform exisiting baseline models. MLP and XGBoost attain the highest performance measures with accuracy (90.55%, 89.97%), precision (91.66%, 90.09%), recall (90.55%, 89.97%) and F1 score (90.7%, 89.78%) respectively.

Keywords: human activity recognition, deep learning, classification, extreme gradient boosting, neural networks

1 Introduction

Human Activity Recognition (HAR) using wearable sensors entails recognising a subjects physical movements by analysing data generated from on-body wearable sensors. These inertial sensors are accelerometers, gyroscopes and magnetometers while the activities identify as Activities of Daily Living (ADL). As mentioned in [1], ADL's involves one's self and body, with specific emphasis on mobility. Sensor-based HAR is dominating current research due to the applicability of sensor fusion which entails the integration of sensor data from multiple sensors which drives analytical results in terms of reliability, accuracy and completeness.

On this view, deep learning methods are continuing to consistently advance and improve the field of HAR. XGBoost is leading the forefront with its in-depth knowledge and computational ability to take data-oriented classification tasks and successfully select and process invaluable features from the data. In this paper, we apply 6 machine learning models to the HAR problem. We build and train several models using on-body sensor signal data generated from 4 different sensors, and we analyse the results in order to identify which model best suits the data in terms of accuracy, precision, recall, F-score and total amount of misclassified instances. This paper shows that XGBoost is the highest performing model due to its ability to peform parallel optimisation and tree pruning while limiting overfitting and consistently learning sparse features.

The rest of this paper is structured as follows: Section 1.1 gives a brief overview of the problem description detailing how HAR usage can aid the healthcare domain. Section 2 presents an overview of the related work for human activity recognition. Section 3 provides an overview of the MHEALTH dataset, the architecture of each model and the approach taken throughout the research. Experiment results are discussed in section 4. Section 5 presents a discussion section while the final section, section 6, discusses future work as well as an overall conclusion.

1.1 Motivation

Human activity recognition has shown to be effective in benefiting clinicians in the treatment and remote monitoring of patients. This field is not only vital for diagnosis and treatment, but also an assessment of how likely a medical patient will fall ill or die from certain diseases or health problems. To show the great importance of activity recognition in the health sector, analytically driving an improvement in accuracy in classifying patients' activities improves the relationship of patients and clinicians as well as reducing the possibility of a fatality.

This paper revolves around the topic of using deep learning to benefit the healthcare industry. One aspect that deep learning could benefit is Remote Patient Monitoring (RPM). The sufficient monitoring of remote persons' activities in real-time can yield great benefits in medical environments. Doctors, nurses and clinicians can build strong relationships with, and improve the experience of their patients, by analysing data send to them via RPM technologies. The data sent to them via RPM can develop a personalised care plan and engage in joint decision-making to foster better outcomes. Wearable sensors, generating this data, can feed data to a clinician in real-time, leading to a significant reduction in continuous patient monitoring and aid diagnostic analysis.

This system could be beneficial to the elderly, those suffering from chronic illness and those who are prone to heart attacks (or serious medical conditions). According to [2], chronic heart failure (CHF) is the most common cause of readmission for patients in the USA. It is estimated that up to 84% of readmissions within a 7-day period were considered preventable, while 76% of 30-day readmission were also considered preventable [2]. The best way to provide protection to patients that are prone to chronic heart failure, chronic illness, disease spreading as well as aiding remote patient monitoring and providing a quick response to fall detection is a Human Activity Recognition Health Model.

In regards to preventing disease spreading, the primary task of the model should be the early detection and prevention of the disease as oppose to recommending preventative measures to cure the diagnosis. The model could provide accurate and timely measures ensuring the disease does not come to surface. It could provide benefits to remote patient monitoring as a part of the intervention could be the early detection and prevention of an elderly person falling, signalling them to control certain movements and be more aware of surroundings. This leads to monitoring patients who are suffering from chronic illness. Similar to preventing the spreading of disease, the model could monitor and control the illness to ensure it does not take hold of the patient and suggest preventative measures if the patient is in critical condition.

2 Related Work

Sensor-based activity recognition is a a continuously evolving field of AI, with a wide-range of research being produced annually. Nguyen, Fernandez, Nguyen and Bagheri [3] give an extensive introduction to HAR, with the integration of multiple sensors. Nguyen et al. [3] built an XGBoost machine learning method using wrist-worn accelerometer data, RGB-D camera data and environmental sensor data to classify activities. This unique approach achieved an increased improvement of 38% accuracy in comparison to previous studies. An average recognition accuracy of 90% and a brier score of 0.1346 was also achieved. Mo, Li, Zhu, and Huang [4] compares convolutional neural networks and multilayer perceptron performance on the classification of activities based on the CAD-60 Dataset. The CAD-60 Dataset [5] contains RGB-D video sequences of subjects performing physical activities. The Microsoft Kinect sensor recorded the sensor signals. Mo et al. [4] focuses on data pre-processing along with feature extraction to generate highly accurate performance results. The model presented combines CNN and MLP by using CNN for feature extraction and using MLP for the classification of the activity. Their model achieved 81.8% accuracy across twelve different types of activities, outperforming existing state-of-the-art.

One aspect that is missing from the previously discussed related work about machine and deep learning models is a comparison of the classification performance of XGBoost, MLP, CNN, ConvLSTM, AE w/ RF and LSTM. Our aim of this paper is to conduct an investigation and compare these six different machine and deep learning algorithms with each other to evaluate which network best suits the MHEALTH dataset.

3 Experiments

The purpose of this paper is to analyse the performance comparison of deep learning algorithms on the MHEALTH dataset. We aim to identify the best deep learning algorithm suited to the MHEALTH dataset, using on-body inertial sensor data, with respect to the activity classification task.

3.1 MHEALTH Dataset

We analyse a dataset collected by Oresti Banos, Rafael Garcia and Alejandro Saez that is freely available from The UCI Machine Learning Repository [6]. The MHEALTH dataset consists of body motion and vital signs recordings from ten subjects with each of different characteristics [6][7]. The subjects' task is to perform 12 different types of activities. The accelerometer, gyroscope and magnetometer placed on the subjects' body measure acceleration, rate of turn and magnetic field orientation. These sensors measures the range of motion experienced by each subjects' body parts. The collected dataset comprises body motion and vital recordings of the ten subjects' performing the physical activities as stated above. Shimmer2 [BUR10] wearable sensors were used for the recordings. Elastic straps complement the sensors on the subjects' chest, right wrist and left ankle.

Figure 1 The following figure outlines three subjects' performing three different activities: 'lying down', 'cycling' and 'waist bends forward'. The Shimmer2 [BUR10] wearable sensors which are attached by elastic straps are clearly visible on the subjects' chest, right wrist and left ankle.

3.2 Approach

In regards to the input adaptation, the streaming signals were fed into the neural networks using a model-driven approach. The methodology process consisted of 7 steps; data preparation, feature extraction, one-hot encoding, training/testing split, hyperparamter setting, model compilation and model evaluation. The MHEALTH dataset consists of static data, it does not change after being recorded and is essentially a fixed dataset. Step 1, data preparation, involves feature extraction, encoding labels to one-hot form, converting the raw data into the right shape for input into the model, normalising the data and finally splitting the data into training and testing. The next step is feature extraction. The MHEALTH dataset consist of 10 log files, with each log file corresponding to each of the ten subjects. In order to extract the features (signal attributes) and labels (activities) of each log file, a feature extraction method is used to successfully extract all the features and labels of each subjects log file. The third step involves encoding the labels to one-hot form. Step 4 involves splitting the data into training and testing in the ratio of 80:20. In step 5, Hyperparameters such as batch size, number of epochs, learning rate, number of hidden layers, type of hidden layers, shape of input, shape of output and the number of parameters are set. Step 6 involves compiling the model, ensuring it is ready to be fitted. It is necessary to structure each model into organised layers. Once the hyperparameters are tuned accordingly, as outlined in step 5, compiling the model can begin. The compiled model is then fitted to the training data in order to classify the volunteers' activities. The final step is model training and evaluation. When the model is compiled and fitted on the training data, it is evaluated against both the training data and the testing data. The models predicted output is compared with the true output.

We built each model revolving around these five aspects: identifying network architecture, identifying network layers, choosing an optimiser, choosing the loss function and hyperparameter setting. Each network model utilises the data values given for each of the 23 signals recorded from the four sensors in order to classify our class variable, which is the movement that each subject performs. Fine-tuning the hyperparameters allow for beneficial development of the training process outcome.

Table 1 MLP Architecture: The MLP model contains 706,317 data instances. The first hidden layer contains 128 units, the second hidden layer contains 256 units, the third hidden layer contains 512 units while the fourth hidden layer contains 1024 units.

MLP	Input layer	Adam: Learning rate set to 0.0001	Categorical Crossentropy	Batch size: 32 Number of epochs: 20
	2 Dropout layers			
	4 Hidden layers			
	Output layer			

Table 2 CNN Architecture: The CNN model contains 245,584 data instnaces. The first hidden layer has 128 neurons, the second hidden layer has 256 neurons while the third hidden layer has 512 neurons.

CNN	Input layer	Adam: Learning rate set to 0.0005	Categorical Crossentropy	Batch size: 32 Number of epochs: 20
	2 ID convolution layers			
	2 MaxPooling1D layers			
	2 Dropout layers			
	3 Hidden layers			
	Output layer			

Table 3 ConvLSTM Architecture: The ConvLSTM model contains 191,376 data instances. The first hidden layer contains 128 units, the second hidden layer contains 256 units while the third hidden layer contains 512 units.

ConvLSTM	Input layer	Adam: Learning rate set to 0.001	Categorical Crossentropy	Batch size: 32 Number of epochs: 20
	2 ID convolution layers			
	2 MaxPooling1D layers			
	1 LSTM layer			
	2 Dropout layers			
	3 Hidden layers			
	Output layer			

Table 4 AE w/ RF Architecture: The AE w/ RF model contains 23,711 data instances. The first hidden layer contains 128 units, the second hidden layer contains 64 units while the third encoding hidden layer contains 512 units, the fourth and fifth hidden layers contain 64 and 128 units respectively.

AE w. RF	Input layer	Adam: Learning rate set to 0.0005	Categorical Crossentropy	Batch size: 32 Number of epochs: 20
	Encoding layer			
	4 Hidden layers			
	Output layer			

Table 5 LSTM Architecture: The LSTM model contains 175,373 data instances. The first hidden layer contains 128 units, the second hidden layer contains 256 units while the third hidden layer contains 512 units.

LSTM	Input layer	Adam: Learning rate set to 0.0001	Categorical Crossentropy	Batch size: 32 Number of epochs: 20
	2 LSTM layers			
	2 Dropout layers			
	3 Hidden layers			
	Output layer			

The full architectural structure of each model is presented in tables 1-6. Setting up each model involved dropout regularisation, normalising inputs, limiting vanishing and exploding gradients and weight initialisation. Dropout regularisation allowed 0.4 (40%) of diverse sets of hidden layers to be 'dropped' as each epoch is initialised, leading each model to learn minute details about the data while updating weights during gradient descent. Normalising inputs enhanced performances by reducing the amount of time the model takes to learn the data while accelerating the training phase. The use of the ReLu activation function led to a reduction in vanishing and exploding gradients and significantly enhanced speed, accuracy and precision. Adam optimisation was set as the learning rate as the hyper-parameters require little or no tuning. The learning rate is fine-tuned to '0.0005' to enhance the speed of the learning process for each neuron.

We set the following hyperparameters for each model: learning rate, number of hidden layers, number of hidden units for different layers, batch size and the number of epochs. The number of hidden layers and the number of hidden units for different layers varied across each model. They ensure results are conclusive, relevant and maximised. The batch size for each model was set to 32 while the number of epochs was set to 20. Batch normalisation ensured successful updates in data values across more than one layer in each model. Batch normalisation allowed each model to reparameterise after each subsequent layer, allowing for successful updates. Batch normalisation provides a key role in constant coordination and updates to ensure results provided accurate predictions in activity.

Table 6 The following table outlines the hyperparameter settings applied before implementation of the XGBoost Architecture.

XGBoost Model	
Max_Depth	10
Number of parallel threads	4
Number of classes	13
Evaluation metric	merror
Objective	multi:softmax
Trainable parameters	161,959
Number of rounds	10

Table 6 outlines the parameters that are set for the implementation of the XGBoost model: The maximum depth of the tree used in the model is set to 10. It’s vital that the model doesn’t become too complex and lead to overfitting. The number of parallel threads used to run XGBoost in this instance is 4. The number of classes is set to 13. The evaluation metric is set to ‘merror’, which is multiclass classification error rate. The softmax objective is set for the XGBoost model, as it is a multiclass classification task.

In conclusion of the hyperparameter evaluation, we showed that: 1. regularisation is excellent in minimising overfitting for the MHEALTH dataset. 2. Adam is the best optimisation algorithm that suits this data. 3. Fine-tuning the hyperparameters to suit the subject data yields excellent, insightful results while speeding up training the model. 4.

3.3 The NULL Class

Human activity recognition systems contain a vast amount of streaming data. Only a certain percentage of this streaming data is significant in the performance of a HAR system. There is a slight imbalance between the portion of significant data and insignificant data. This leads to some of the activities to be easily confused with activities that have similar range of motion patterns and are irrelevant in predicting the activity in question. For example, jogging is often mistaken for running and cycling is often mistaken for running upstairs. These easily confused activities are the so-called NULL class. Detecting, monitoring and modelling the NULL class is a tough task. The NULL class often represents a massive portion of the dataset. As seen in [8], the NULL class represents 72.28% of the whole dataset. It is good practice to remove the NULL class if there is a skewed pattern in the dataset. If the datasets attribute information and labels differ substantially from the correctly classified activities, then the NULL class problem may be identified and appropriate action or precautions taken. The NULL class is not a huge problem. At most, it leads to minor confusion when classifying activities in HAR systems. As seen in [9], applying self-learning can reap benefits of the NULL class. The studies in [9] present a performance comparison of self-learning activity spotters to show the benefits of this proposed approach. Results yielded an increase of 15% in performance, which outlines that the NULL class if managed accordingly can generate great model performance.

4 Experimental results

A confusion matrix visualises the percentage of misclassified instances of each approach. The matrix describe what activities are misclassified, as well as the depth in which each misclassification occurred.

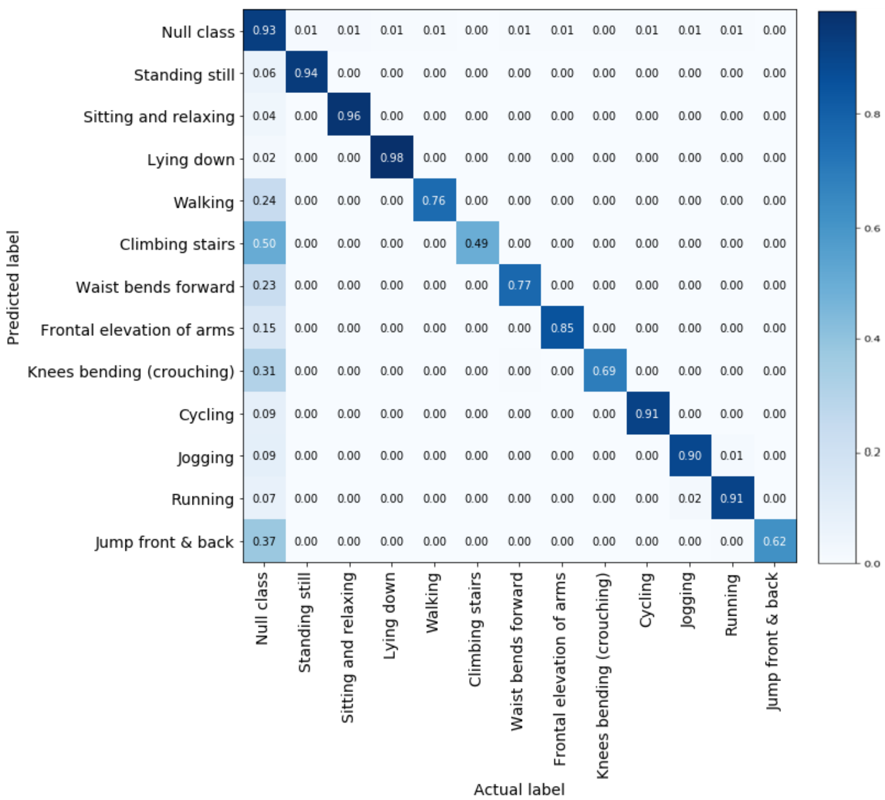

Figure 2 The XGBoost confusion matrix outlines the accuracy for correctly classifying each activity. The XGBoost approach achieved an accuracy of 89.97%.

As explained in section 3.3, there is a significant class imbalance due to the presence of the NULL class. As seen in figure 2, the NULL class has a significant contribution on the amount of false positives and false negatives detected. It accounts for a large portion of misclassified activities. Including the NULL class in the analysis results leads to a high percentage of data processed as ‘not an activity of significant interest’ or ‘not a classifiable activity’ in terms of the labels (activities). In order to conduct appropriate analysis of each confusion matrix and greater understand the data; the NULL class is ignored as it accounts for 71-72% of the dataset, depending on each individual models trainable parameters. As presented in table 7, the MLP model slightly struggles in distinguishing between the jogging activity and running activity with 7% (425 times) of the activity misclassified, while XGBoost misclassified 3% (186 times) of the activity. Similarly, for the activities ‘jumping front and back’ and ‘jogging’, XGBoost and MLP made little errors with 19 (<1%) and 46 (<1%) respectively. MLP misclassified 471 instances while XGBoost misclassified only 281. CNN, ConvLSTM and LSTM misclassified 1341, 2533 and 2742 instances respectively. The enhanced performance of XGBoost in classifying these activities and producing fewer errors is due to its extreme gradient boosting framework that can implement effective tree pruning, regularisation and parallel processing subsequently.

Table 7 The total number of misclassified instances for each approach is presented in the following table.

Machine Learning Method	***Total Misclassified Instances***
XGBoost	**281**
MLP	471
CNN	1341
ConvLSTM	2533
AE w/ RF	2689
LSTM	2742

As presented in table 8, we were able to produce a very competitive accuracy, precision, recall and F-score by implementing sensor data from gyroscopes, accelerometers, magnetometers and an electrocardiogram. The deep learning approaches presented in this paper were able to learn complex relationships from inputs to outputs while supporting the 23 features and 12 classes in question. Our approaches prevailed as superior with their ability to learn linear and non-linear relationships and learn multivariate inputs.

Table 8 The following table presents our approaches performance comparison. Each architecture is compared in terms of accuracy, precision, recall and F1 score. Upon comparison of each architecure in terms of each evaluation metric and total misclassified instances, XGBoost is the top performing model due to its performance, speed and scalability.

Architecture	***Accuracy***	***Precision***	***Recall***	***F1 Score***
MLP	90.55%	91.66%	90.55%	90.7%
XGBoost	89.97%	90.09%	89.97%	89.78%
CNN	83.91%	83.47%	83.91%	82.98%
ConvLSTM	83.89%	83.69%	83.89%	83.2%
AE w/ RF	83.27%	82.59%	83.25%	81.54%
LSTM	78.09%	74.86%	78.09%	75.6%

Table 3 compares the accuracy, precision, recall and F1 Score of the proposed machine learning approaches. MLP attains the highest percentage of the four performance comparison measures, achieving 90% or greater. XGBoost falls slightly short of the top spot but still achieving excellent results, achieving 89% or greater. ConvLSTM, CNN and AE w/ RF achieve satisfactory results, with LSTM being the poorest performing model.

5 Discussion

The main conclusions from the comparison of MLP, XGBoost, CNN, LSTM, ConvLSTM (CNN+LSTM) and AE w/ RF on the MHEALTH dataset is that: MLP and XGBoost reaches a higher accuracy (90.55%, 89.97%), precision (91.66%, 90.09%), recall (90.55%, 89.97%) and F1-score (90.7%, 89.78%) respectively. MLP and XGBoost are significantly superior in their ability to distinguish between similar activities (e.g., 'jogging/running' and 'climbing stairs/knees bending (crouching)'). To the authors' knowledge, XGBoost has not been implemented on the MHEALTH dataset in order to classify each subjects' activity. Findings suggest that XGBoost can be successfully applied to the MHEALTH dataset and be comparable to existing state of the art baselines.

These conclusions reinforce the hypothesis that an XGBoost model created and implemented on the MHEALTH dataset to predict human activities generates significant power to learn temporary feature activation dynamics and make decisive predictions in classifying the subjects' predicted activity. The XGBoost architecture offers much better analysis characteristics than the other five classification models. These characteristics include regularisation, tree pruning, tree depth and sparse features. XGBoost identifies the vital signs and range of motion of the activities in question more accurately. All of these findings mentioned in this discussion section reiterate the hypothesis that XGBoost is the best performing model and is highly suited to analysing MHEALTH data.

Although MLP outperformed XGBoost in terms of accuracy, precision, recall and F1-score, MLP misclassified 471 instances while XGBoost misclassified only 281. CNN, ConvLSTM and LSTM misclassified 1341, 2533 and 2742 instances respectively. In terms of overall accuracy, precision, recall, F1-score and number of correctly classified instances, XGBoost is the top performing model. This details the known domain of appropriateness for the XGBoost framework, which has never been reported on the MHEALTH dataset before.

Many deep learning architectures implement convolutional, pooling and dropout layers successively, to increase model performance and reduce the degree of data complexity. However, implementing these layers are not strictly vital. XGBoost does not include convolutional, pooling or dropout layers as it processes data under the gradient boosting framework. XGBoost is excellent for increasing performance and speed due to its ability to implement a variety of gradient boosted decision trees to analyse data and generate meaningful, decisive conclusions. XGBoosts results (accuracy 89.97%, precision 90.09%, recall 89.97%, F1-score 89.78%) proves it can generate excellent performance with a high degree of data complexity presented by the MHEALTH dataset. Convolutional, pooling and dropout layers also present many benefits. They are becoming significantly useful in analysing data spread across a more profound period.

Machine learning architectures, which are fully connected, contain values in the dense layer that must be linked with every parameter value of the last feature map (previous layer). This leads to the formation of a weight matrix that is significantly large, it is vital in ensuring the parameters of the connection doesn't get out of proportion. The Gradient boosting framework built into XGBoost ensures that the amount of parameter values needed is minimised. XGBoost is said to me a more complex network, but it is formed of a reduced number of parameters, and is directly linked to the outstanding benefits it produce in respect to GPU memory and hard drive computational processing power.

6 Conclusion and Future Work

In this paper, we presented a comparative study of deep learning algoritgms for the HAR problem. We focused our research on the MHEALTH dataset, which contains a diverse set of activities as well as sensor data extracted from four different wearable, electronic sensors. Our aim was to examine the classification proficiency of each individual deep learning model. Our experimental results show that Extreme Gradient Boosting (XGBoost) achieved the highest classification capability, upon analysing its accuracy (89.97%), precision (90.09%), recall (89.97%), F1-score (89.78%), confusion matrix and total amount of misclassified instances (281). XGBoost can undoubtedly address the problem of human activity recognition in the context of MHEALTH data.

Future work on the application of XGBoost to real-world data, particularly around HAR in the healthcare domain, is recommended. To be precise, conducting analysis on 100+ subjects' could be interesting, in order to justify the classification capabilities to a broader range of subjects', which could lead to more insightful conclusions showing why and how the model behaved on certain subjects like it did. Long-term monitoring is another possibility of future work. Another possibility for future work is to compare the neural networks models performance metrics when using data from individual sensors or subsets of the MHEALTH dataset. This would increase practicality in producing a real world HAR solution.

An important addition to this project would be to focus more on the XGBoost implementation due to the successful performance it achieved. XGBoost is excellent for model interpretability, which is a huge aspect in machine and deep learning nowadays. Due to time constraints, analysing XGBoost shapley values wasn't feasible. Shapley values allows the XGBoost model to analyse a feature set and identify each feature's marginal contribution to the overall classification prediction. Shapley values provide a very detailed account as which features greatly influenced the model. They offer transparency as well as global approximations. LIME (Local Interpretable Model-agnostic Explanations) is another technique, which would have benefited this research greatly. It also offers detailed account of model interpretability, detailing the highly important influential features. Lime offers local approximations while shapley offers global approximations. Upon extending this research, a comparison of both measures to improve model interpretability would greatly benefit the whole research.

References

1. Wiener, J.M., Hanley, R.J., Clark, R. and Van Nostrand, J.F., 1990. Measuring the activities of daily living: Comparisons across national surveys. *Journal of gerontology*, *45*(6), pp.S229-S237.
2. Chamberlain, R.S., Sond, J., Mahendraraj, K., Lau, C.S. and Siracuse, B.L., 2018. Determining 30-day readmission risk for heart failure patients: the Readmission After Heart Failure scale. *International journal of general medicine*, *11*, p.127.
3. Nguyen, T.T., Fernandez, D., Nguyen, Q.T. and Bagheri, E., 2017, November. Location-aware human activity recognition. In *International Conference on Advanced Data Mining and Applications* (pp. 821-835). Springer, Cham.
4. Mo, L., Li, F., Zhu, Y. and Huang, A., 2016, May. Human physical activity recognition based on computer vision with deep learning model. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings* (pp. 1-6). IEEE.
5. re3data. 2019. Cornell Activity Datasets: CAD-60 & CAD-120. [online] Available at: https://www.re3data.org/repository/r3d100012216.
6. Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado-Terriza, J.A., Lee, S., Pomares, H. and Rojas, I., 2015. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online*, *14*(2), p.S6.
7. Banos, O., Garcia, R., Holgado-Terriza, J.A., Damas, M., Pomares, H., Rojas, I., Saez, A. and Villalonga, C., 2014, December. mHealthDroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living* (pp. 91-98). Springer, Cham.
8. Li, F., Shirahama, K., Nisar, M., Köping, L. and Grzegorzek, M., 2018. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, *18*(2), p.679.
9. Amft, O., 2011, June. Self-taught learning for activity spotting in on-body motion sensor data. In *2011 15th Annual International Symposium on Wearable Computers* (pp. 83-86).