Goldsmiths
UNIVERSITY
OF LONDON

IS71059C/B: Big Data Analysis (2023-24)

Human Activity Recognition Trondheim for Machine Learning using Hadoop Big Data Platform

Sandor Kanda

Table of Contents

Table of Figures

List of Tables

Introduction

Human activities are important because they reveal information about their identity, personality, physical characteristics, and mental state. Recognising activities is natural and simple for the average person, but the process requires complex sensing, learning, and inference functions for a computer. Modern computers face significant challenges in detecting their surroundings, learning from previous experiences, and applying knowledge for activity inference. [1] The professionally annotated Human Activity Recognition Trondheim (HARTH) data contains the sensor recordings of participants wearing three-axial accelerometers in a free-living setting for approximately two hours.

This study seeks to tackle these obstacles by examining the HARTH dataset and constructing a machine-learning workflow to identify human activities accurately. The aim of this study encompasses pre-processing the unprocessed sensor data, extracting pertinent characteristics, implementing and comparing diverse machine learning algorithms such as Decision Trees, Random Forests, Logistic Regression, and Deep Learning models, and evaluating their performance using suitable metrics. In addition, the computational efficiency of the models will be evaluated, and methods for expanding the solution using distributed technologies such as Hadoop's HDFS and Spark/Spark MLlib will be suggested. This research aims to enhance the development of reliable and adaptable human activity recognition systems by utilising the HARTH dataset, which provides realistic and challenging data. These systems have the potential to be applied in various fields, such as healthcare, fitness tracking, and smart homes.

Topic Proposal

Data with summary statistics

The dataset for the HARTH analysis is sourced from the sensors that 22 participants wore. The sensors are two 3-axial Axivity AX3 accelerometers attached to the right thigh and the lower back. The provided sampling rate is 50Hz. Video recordings of a chest-mounted camera were used to annotate the performed activities frame-by-frame. The professional readings and annotations are provided in comma-separated values (CSV) format. Each participant's data is in a separate file.

The full dataset contains 6.461328 million rows of data and eight features. The variables are represented in Table 1 below.

Table 1. Variables Description

Column Name	Description
timestamp	date and time of recorded sample
back_x	acceleration of back sensor in x-direction (down) in the unit g
back_y	acceleration of back sensor in y-direction (left) in the unit g
back_z	acceleration of back sensor in z-direction (forward) in the unit g
thigh_x	acceleration of thigh sensor in x-direction (down) in the unit g
thigh_y	acceleration of thigh sensor in y-direction (right) in the unit g
thigh_z	acceleration of thigh sensor in z-direction (backward) in the unit g
label	annotated activity code

In Table 1 above, the timestamp of the back and the thigh values are features, and the label is the target variable to predict.

The target labels are the annotated activities that can be walking, running, shuffling, stairs (ascending), stairs (descending), standing, sitting, lying, cycling (sit), cycling (stand), cycling (sit, inactive), cycling (stand, inactive).

The first participant’s annotated data shows the time spent in each activity and the percentage of the time in Table 2 below.

Table 2. Activity Duration for a single participant

Label	Duration (hh:mm:ss)	Percentage
Walking	01:56:42	22.96
Running	00:01:46	0.35
Shuffling	02:04:11	24.43
Stairs (ascending)	00:04:57	0.98
Stairs (descending)	00:01:27	0.29
Standing	02:04:10	24.43
Sitting	02:08:39	25.31
Lying	00:06:25	1.26

Table 2 above shows that the participant spent most of the time sitting (2 hours 8 minutes/ 25.31%) and the least time spent running (1 minute 46 seconds/ 0.35%) along other listed activities.

Hypotheses

This research investigates various hypotheses regarding human activity recognition using the HARTH dataset. The initial hypothesis posits that algorithms can effectively classify human activities by analysing accelerometer data. This hypothesis is crucial to identifying and differentiating different human activities using sensor data. The purpose of testing this hypothesis is to showcase the efficacy of machine learning algorithms in resolving the activity recognition problem. The hypothesis has the potential to lead to the creation of precise activity recognition systems that can be applied in a range of areas.

The second hypothesis emphasises the significance of feature selection in recognising activities. It is postulated that specific characteristics derived from the accelerometer data are more informative for identifying activities than others. This hypothesis aims to determine the most pertinent and distinguishing attributes of activity recognition. By investigating this hypothesis, it is possible to ascertain the key features that have the greatest impact on the classification performance. This could decrease the computational complexity by prioritising the most informative features. The hypothesis significantly impacts the advancement of activity recognition systems, as it allows for utilising the most pertinent features, resulting in improved efficiency and effectiveness.

Finally, there is a hypothesis that the performance of various machine learning algorithms differs based on the particular activity being identified. This hypothesis investigates the difficulty of choosing the most appropriate machine learning algorithm for accurately identifying multiple types of activities. To test this hypothesis, one must compare the performance of different algorithms on various activity classes and identify any discernible patterns or trends. This hypothesis significantly affects identifying the most effective algorithms for specific activities. It allows for the creation of customised activity recognition models that maximise performance for each type of activity.

Planned Analysis

The data is sourced from the archive.ics.uci.edu domain and was manually downloaded to a local repository. The dataset is licensed under a Creative Commons Attribution 4.0 International license, allowing sharing and adaptation for this research. The experiment will utilise Hadoop's HDFS distributed cluster environment. The implementation will be written using Pyspark. The data pre-processing steps, using the MLlib library, aim to handle missing values if present, remove outliers, and normalise the data, making it suitable for the predictive models. The data analysis will utilise the statistical and data visualisation techniques to gain valuable insight into the data distribution. The feature variables are abstract and do not provide identifiable information that may be linked to individuals.

Logacjov et al.'s .experiment employed seven machine learning models to recognise human activities: k-nearest neighbours (k-NN), support vector machine (SVM), random forest (RF), extreme gradient boost (XGB), convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and CNN with multi-resolution modules. The Support Vector Machine (SVM) obtained the highest performance with an average F1-score of 0.81, recall of 0.85, and precision of 0.79 using a leave-one-subject-out cross-validation technique. The XGB model exhibited the second-highest level of performance, while the k-NN model came after it. The deep learning models, including CNN, BiLSTM, and multi-resolution CNN, exhibited inferior performance compared to the traditional machine learning models. [2]

This research aims to compare four different models. Decision trees are simple, interpretable, and well-suited for numerical features. [3] Random forests can capture non-linear relationships between features and activities, aligning with the hypothesis that different activities can be accurately classified using machine learning algorithms. [4] Logistic regression serves as a baseline model to compare the performance of more complex algorithms. [5] Multi-Layer Perceptron's (MLP) ability to automatically learn hierarchical representations of the input data can be beneficial for capturing intricate patterns in the accelerometer signals. [6]

The above models will be compared, and a set of evaluation metrics will assess the performance of the trained models. The classification metrics are accuracy, precision, recall and F1-score.
In conclusion, by leveraging the power of Hadoop's HDFS for distributed storage and Spark's distributed computing capabilities, along with its MLlib library, the research can effectively handle the large-scale HARTH dataset, pre-process the data, train and evaluate multiple machine learning models, and assess their performance using various evaluation metrics. This distributed approach enables efficient dataset analysis, overcoming the challenges posed by its massive size and complexity.

Implementation

Data Acquisition

The acquired data from the local repository was uploaded onto the Data Server Manager (DSM) platform using the terminal command line from the. DSM location, the folder containing the CSV files was moved to an HDFS location.

The notebook file is on *https://dsm-jupyter.doc.gold.ac.uk/user/skand001/notebooks/CW2/bdacw2_dsm.ipynb*.

The same file on the submission folder is on the path *'./cw2_pyspark/ bdacw2_dsm.ipynb*' location. The data is located on HDFS on '*/user/skand001/harth_2*', also on the web address 'https://dsm-jupyter.doc.gold.ac.uk/user/skand001/tree/CW2/harth_2'.

The link to the data is shared on a Google Drive link.

DSM Location

The necessary PysSpark libraries were imported into the notebook. Four CSV files containing four participants' annotated sensor outputs were combined into one data frame using spark.read.csv. The sensor reading data type is float64, the target label is int64 format, and no number (NaN) values are found. The data pre-processing steps started with searching for missing values do not present in the dataset.

The server performance was insufficient to execute the experiment, as the notebook hung excessively long, making it difficult to progress. The module leader instructed that the research must be completed in local mode if any issues were encountered on the server.

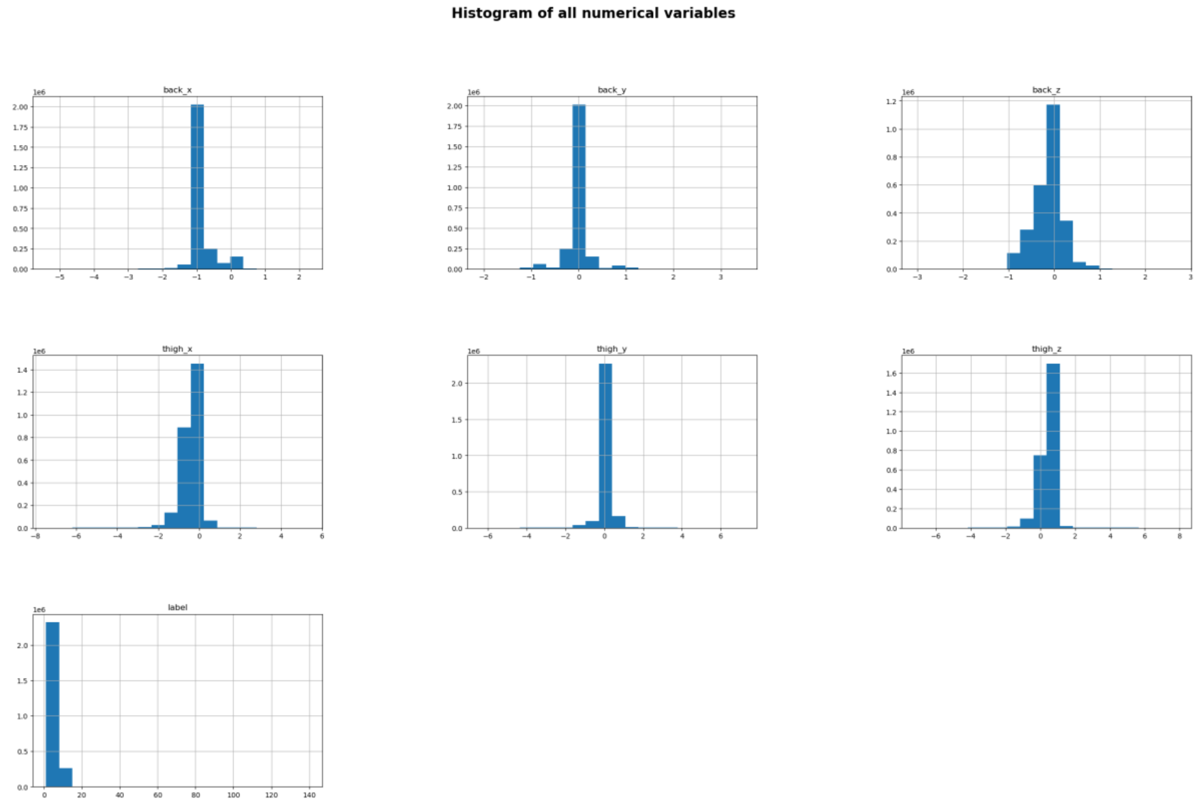
Therefore, to progress, the author continued the experiment on the HARTH data on a GPU server. However, the PySpark/ Big Data environment was not replicable, but the data analysis and the machine learning pipeline provided valuable results.

The submission folder contains a '*cw2_python'* folder, a data folder, and the 'bda.ipynb' notebook file, the script for the analysis's end-to-end pipeline, excluding HDFS and PySpark. A 'requirements.txt' file is also created, ensuring reproducibility and consistency across environments.

Exploratory Data Analysis

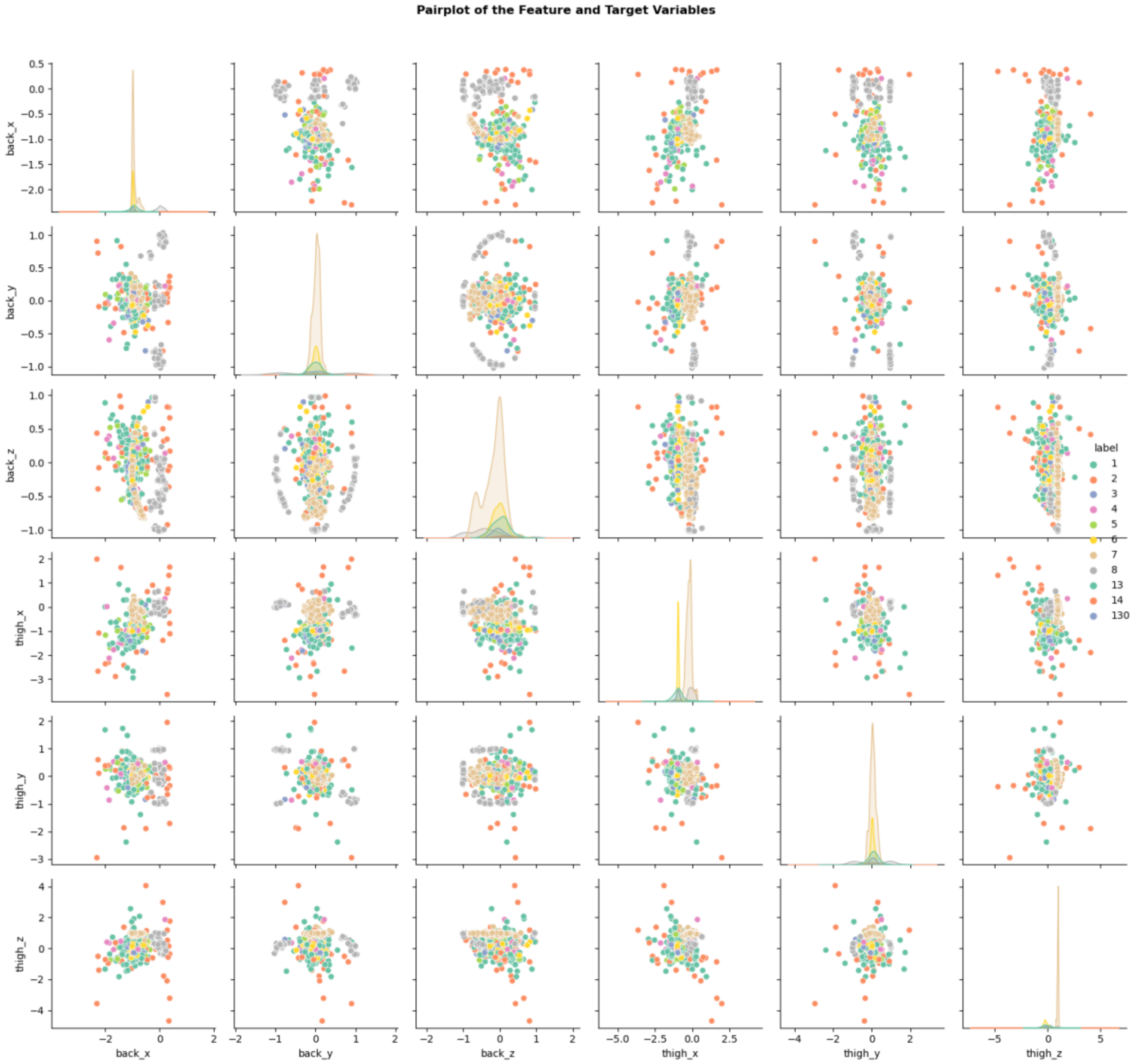
The data distribution is visualised in the histogram in Figure 1.

Figure 1. Histogram of the feature and target variables



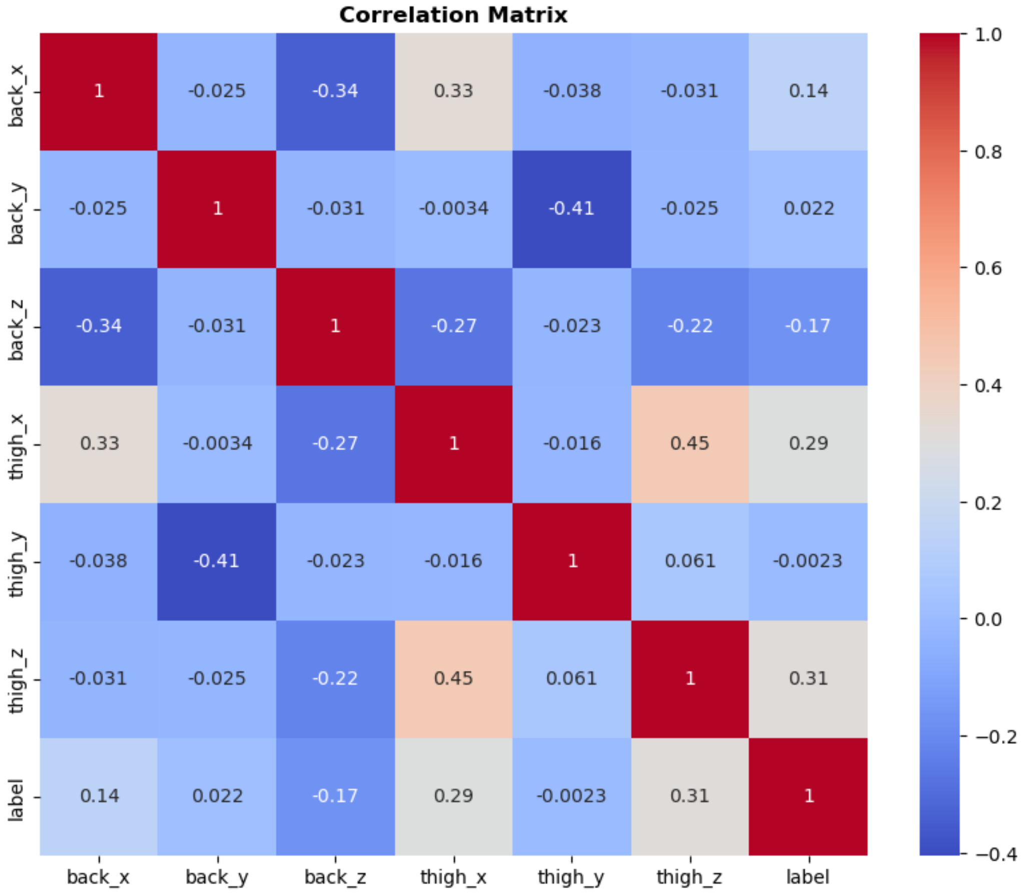
In Figure 1 above, the back and thigh variables have a normal distribution due to the sensor's calibration. However, the labels tend to be slightly left-skewed due to the multiple classes.

Figure 2. Pair plot of the predictors and the target variable



In Figure 2 above, the pair plot shows normalised distributions for variables such as "back_x," "back_y," "back_z," "thigh_x," "thigh_y," and "thigh_z," all centred on zero. Notably, the scatter plots coloured by "label" show clear clustering, indicating that these features effectively distinguish between categories. Correlations between features suggest the possibility of redundancy, whereas outliers may impact model performance. The variable "label" exhibits a discrete distribution, indicating its categorical nature. To reduce the computational requirements and to prevent an overcluttered plot, 2000 randomly selected rows of data were used for this visualisation.

Figure 3. Correlation Matrix of the sample of the dataset



In Figure 3 above, the correlation matrix displays a combination of positive and negative correlations between the variables. The diagonal depicts a perfect correlation between each variable and itself (1.0). The highest positive correlations are found between high_z and label (0.31) and high_x and thigh_x (0.45). The strongest negative correlations are found between back_y and thigh_y (-0.41) and thigh_y and high_y (-0.38). Overall, the correlations range from weakly negative to moderately positive, indicating some interrelationships among the variables but no extremely strong associations.

Table 3. Descriptive Statistics Table

Measure	back_x	back_y	back_z	thigh_x	thigh_y	thigh_z	label
count	2580668	2580668	2580668	2580668	2580668	2580668	2580668
mean	-0.8741502	0.00182915	-0.1328038	-0.4712772	0.05489203	0.5656109	6.214782
std	0.3199065	0.2419559	0.3225808	0.4765173	0.3292622	0.5753035	3.927469
min	-5.406958	-2.071573	-3.055944	-7.483251	-6.38598	-7.20822	1
25%	-0.9998179	-0.0647157	-0.3045925	-0.9176257	-0.0706086	0.06640893	6
50%	-0.9841057	0.01575966	-0.0719985	-0.3429117	0.03958397	0.9048722	7
75%	-0.8915544	0.08074149	0.05998398	-0.1495664	0.1557272	0.9663542	7
Max	2.291708	3.482324	2.723054	5.372528	7.182237	7.914523	140

Table 3 above shows the descriptive statistics of the dataset used on a GPU server, using the data of 7 participants. The table includes the predictors and the target variable, the count, the average values, the standard deviation, the minimum and maximum values, and the quartiles.

Table 4. Class Distribution of the dataset

Activity	Counts
sitting	1517231
standing	322101
walking	268666
lying	216104
shuffling	117451
cycling (sit)	36784
running	32854
stairs (ascending)	32384
stairs (descending)	28738
cycling (stand)	6635
cycling (sit, inactive)	1560
cycling (stand, inactive)	160

Table 4 above indicates that the dataset is imbalanced. The sitting activity is the most dominant class, and cycling (stand, inactive) is the least common class.

Machine Learning Algorithms

After the data (n = 2.580668e+06) was pre-processed and ready for the machine learning pipeline, it was split into 70% training, 15% validation, and 15% testing.

Random Forest

The random forest algorithm is an extension of the bagging method. It uses bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, or feature bagging, generates a random feature subset, ensuring low correlation among decision trees. [7]

In this experiment, the random forest algorithm was initialised with 100 estimators. The training data was evaluated on the validation.

Table 5. Random Forest Classifier Performance on Validation Data

Random Forest - Validation Metrics
Accuracy: 0.9183311805734953
Precision: 0.9152517181163798
Recall: 0.9183311805734953
F1 Score: 0.9140720728196632

Table 5 shows the results of the Random Forest classification algorithm on the validation data, with all four metrics achieving a value above 0.91.

Decision Trees

Decision Trees are a non-parametric supervised learning method for a classification problem. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. [8]

Table 6. Decision Tree Performance on Validation Data

Decision Tree - Validation Metrics
Accuracy: 0.8862955308705761
Precision: 0.8863775476816477
Recall: 0.8862955308705761
F1 Score: 0.8863303990545803

Table 6 shows the Decision Tree algorithm's performance on the validation data. It is slightly lower than the Random Forest, just over 0.88 for all four classification metrics.

Logistic Regression

Logistic regression is a classification algorithm for datasets with numerical input variables and a categorical target variable with two values or classes. The multinomial logistic regression suits the hypotheses. The logistic regression model will use the multinomial loss function. It is suitable when there are more than two classes, and the goal is to predict the probability of each class. 'lbfgs' Limited-memory Broyden-Fletcher-Goldfarb-Shanno) is an optimisation algorithm that belongs to the family of quasi-Newton methods. It is particularly useful when the number of features is relatively large compared to the number of training instances. [9]

Table 7. Multinominal Logistic Regression Performance on Validation Data

Logistic Regression - Validation Metrics
Accuracy: 0.8111418238181348
Precision: 0.767975216084394
Recall: 0.8111418238181348
F1 Score: 0.7735513722009466

Table 7 above represents the evaluation metric values of the multinominal Logistic Regression. The results are lower than those of the previous algorithms.

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a neural network where the mapping between inputs and output is non-linear and one or more hidden layers with many neurons stacked together. Each layer feeds the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer. [10] The MLP in this experiment consist of 100 hidden layers and uses the ReLU (Rectified Linear Unit) activation and Adam solver.

Table 8. Multi-Layer Perceptron Neural Network Performance on Validation Data

MLP - Validation Metrics
Accuracy: 0.8967656936192199
Precision: 0.8865465717996525
Recall: 0.8967656936192199
F1 Score: 0.8870825297524052

Table 8 above shows the Multi-Layer perceptron evaluation metrics, which tend to be higher than the Decision Tree values.

Figure 4.A Scatter plot of the Model Performance Comparison

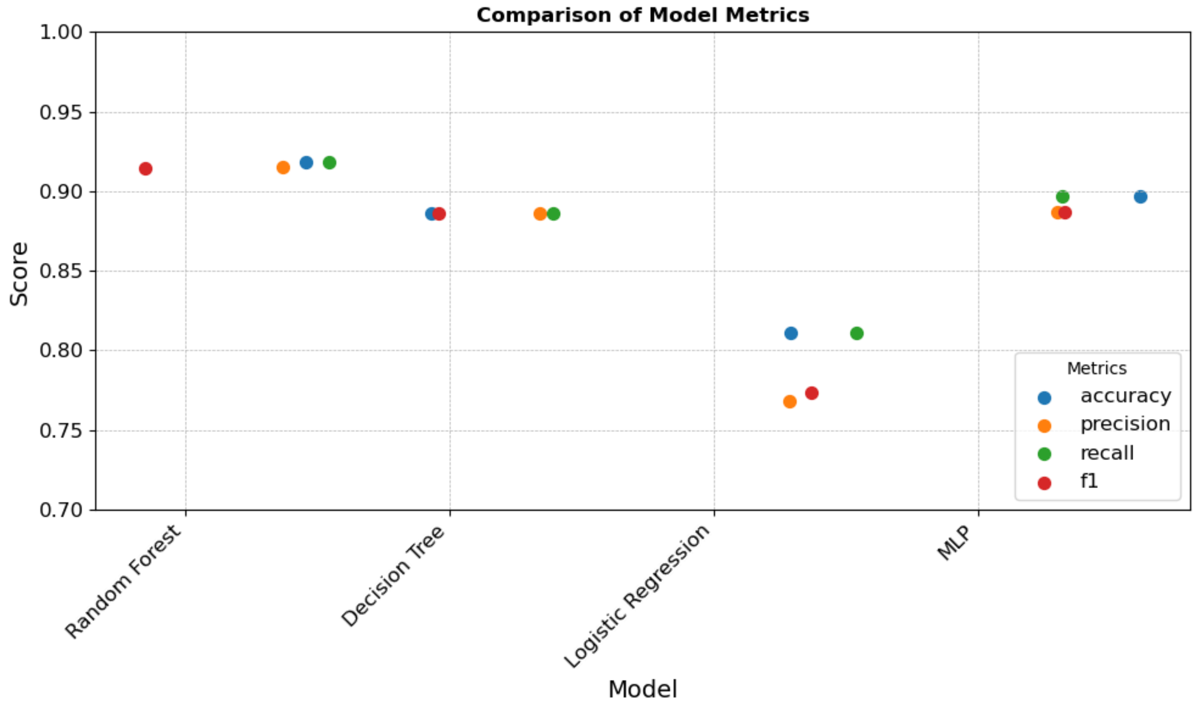


Figure 4 above shows the scatter plot of the four machine learning models' performance on the validation data and the metrics. The plot shows that the random forest algorithm performance achieved the highest scores.

Model Evaluation

Due to the dataset imbalance problem, the F1 Score evaluation metric was prioritised, as it is more suitable and provides a more precise evaluation than accuracy. It is the combination of the Precision and Recall metrics. Finally, the random forest algorithm was run on the unseen test data.

Table 9. Random Forest Evaluation Metrics on the Test Data

Random Forest Performance on the Test Set
Accuracy: 0.91882
Precision: 0.91559
Recall: 0.91882
F1 Score: 0.91466

Table 9 shows the random forest algorithm performance on the unseen test data. The results are very close to those from the validation set.

Figure 5. Scatter Plot of the Random Forest Metrics on the test data

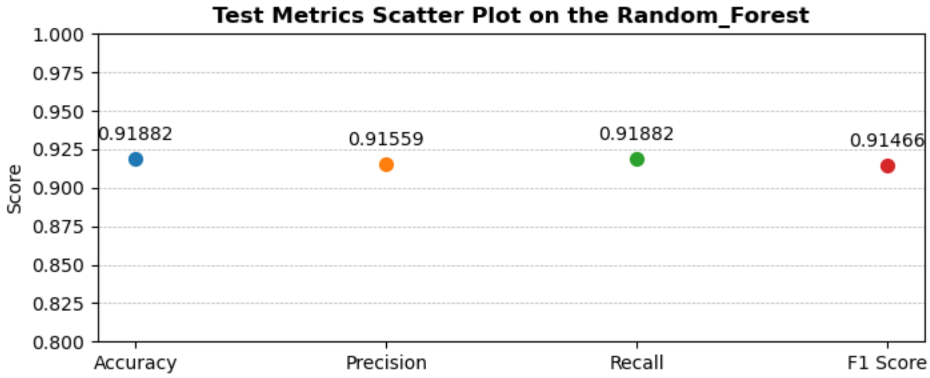


Figure 5 above is a visual representation of the values from Table 9 for the best-performing model on the test data.

Conclusion

This study specifically addressed the issue of identifying and categorising human activities by utilising the HARTH dataset. The objective was to create a machine-learning workflow that effectively categorises human activities by analysing accelerometer data. The study postulated that machine learning algorithms can accurately differentiate between human activities. It also suggested that certain features provide more valuable information for activity recognition. Furthermore, the study found that the performance of the algorithms differs depending on the specific activity being classified.

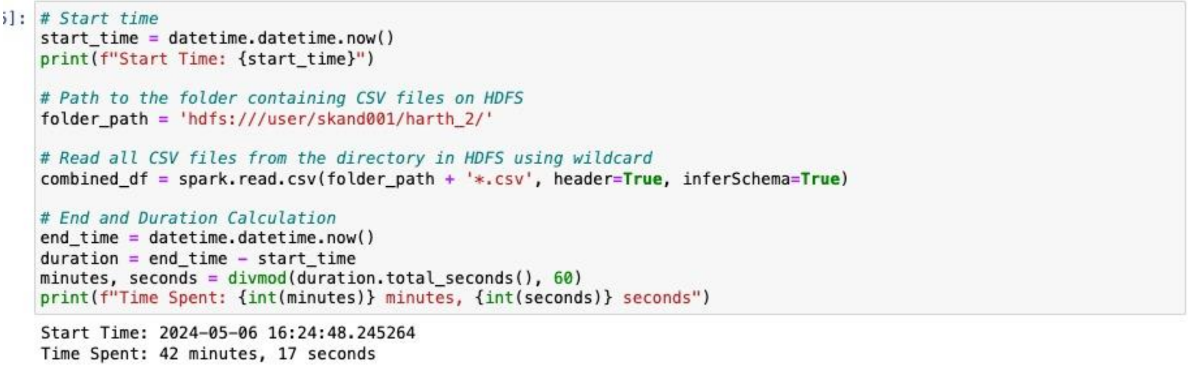
A distributed system for analysing large amounts of data was created to test these hypotheses. This system utilised Hadoop's HDFS to store the data and Apache Spark's MLlib to investigate it. The unprocessed sensor data underwent pre-processing, where pertinent characteristics were extracted. Subsequently, various machine learning algorithms, such as Decision Trees, Random Forests, Logistic Regression, and Multi-Layer Perceptron (MLP), were employed.

The results demonstrated that the Random Forest algorithm exhibited superior performance on the validation and test datasets, attaining an F1 score of 0.91466 on the test set, which outperformed Logacjov et al.'s average F1-score of 0.81. This discovery corroborates the original hypothesis that machine learning algorithms can proficiently categorise human activities by utilising accelerometer data. Selecting features emphasised the significance of particular characteristics in differentiating between activities, thereby confirming the second hypothesis. Moreover, the different performance of the algorithms in different activity categories supports the third hypothesis.

The proposed solution's implementation is distributed, aiming to utilise Hadoop's HDFS and Spark/Spark MLlib. This ensures that the solution can handle significantly larger datasets by scaling accordingly, which is essential for practical implementations of human activity recognition systems.

The notebook hung excessively long, making it challenging to analyse. Despite multiple attempts to optimise the code and troubleshoot the issue, the server's performance remained insufficient to handle the experiment's computational requirements.

Figure 6. The DSM server hung while loading the data from HDFS.



```
# Start time
start_time = datetime.datetime.now()
print(f"Start Time: {start_time}")

# Path to the folder containing CSV files on HDFS
folder_path = 'hdfs:///user/skand001/harth_2/'

# Read all CSV files from the directory in HDFS using wildcard
combined_df = spark.read.csv(folder_path + '*.csv', header=True, inferSchema=True)

# End and Duration Calculation
end_time = datetime.datetime.now()
duration = end_time - start_time
minutes, seconds = divmod(duration.total_seconds(), 60)
print(f"Time Spent: {int(minutes)} minutes, {int(seconds)} seconds")
```

```
Start Time: 2024-05-06 16:24:48.245264
Time Spent: 42 minutes, 17 seconds
```

Figure 6 above shows a snippet of a simple task loading 190Mb of data in the notebook on https://dsm-jupyter.doc.gold.ac.uk/user/skand001/notebooks/CW2/bdacw2_dsm.

The inability to utilise the university's big data environment is a substantial loss for this project. Working with PySpark and Hadoop's HDFS would have provided valuable hands-on experience managing and processing big data using industry-standard tools. Moreover, it would have demonstrated distributed computing frameworks' scalability and performance benefits in tackling complex machine-learning tasks.

Despite this setback, the data analysis and machine learning pipeline implemented on the GPU server yielded valuable results. The insights gained from this experiment highlight the importance of robust infrastructure and optimised computing resources in handling big data projects. Future work should explore ways to enhance the university's big data environment, ensuring that students can fully leverage the power of distributed computing frameworks like PySpark and Hadoop's HDFS in their research endeavours. Future research could incorporate supplementary sensor modalities, examine the influence of various feature selection techniques, and assess the effectiveness of more sophisticated deep learning architectures. Ultimately, the study showcases the efficacy of machine learning algorithms in accurately identifying human activities by utilising the HARTH dataset. The suggested distributed big data analysis pipeline offers a scalable solution for activity recognition tasks with great potential to impact healthcare and the wearable device industry.

References

[1] ‘https://archive.ics.uci.edu/dataset/779/harth’.

[2] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, ‘HARTH: A Human Activity Recognition Dataset for Machine Learning’, *Sensors*, vol. 21, no. 23, p. 7853, Nov. 2021, doi: 10.3390/s21237853.

[3] L. Bao and S. S. Intille, ‘Activity Recognition from User-Annotated Acceleration Data’, in *Pervasive Computing*, vol. 3001, A. Ferscha and F. Mattern, Eds., in Lecture Notes in Computer Science, vol. 3001. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–17. doi: 10.1007/978-3-540-24646-6_1.

[4] K. Ellis, J. Kerr, S. Godbole, G. Lanckriet, D. Wing, and S. Marshall, ‘A random forest classifier for the prediction of energy expenditure and type of physical activity from wrist and hip accelerometers’, *Physiol. Meas.*, vol. 35, no. 11, pp. 2191–2203, Dec. 2014, doi: 10.1088/0967-3334/35/11/2191.

[5] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, ‘Activity recognition using cell phone accelerometers’, *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[6] F. Ordóñez and D. Roggen, ‘Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition’, *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/s16010115.

[7] ‘https://www.ibm.com/topics/random-forest’.

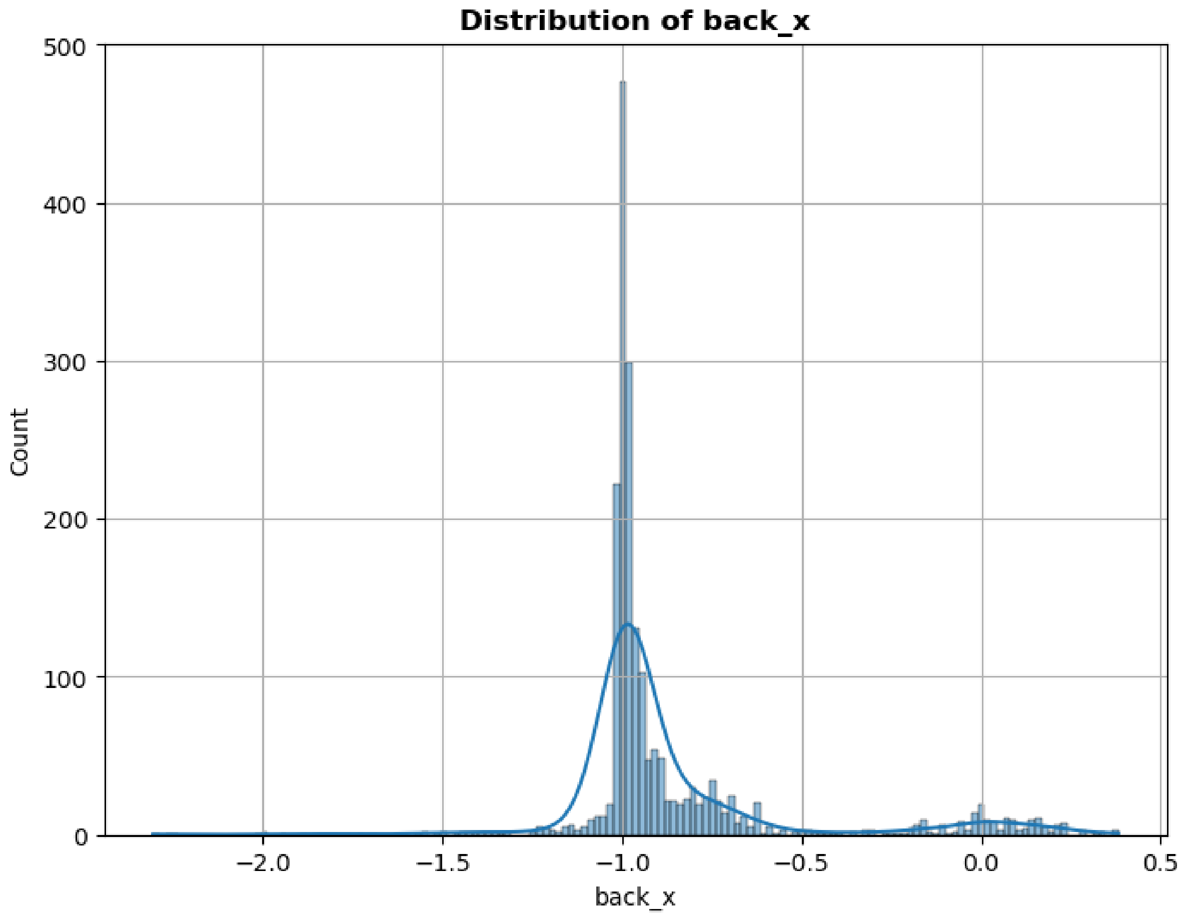
[8] ‘https://scikit-learn.org/stable/modules/tree.html’.

[9] ‘https://machinelearningmastery.com/multinomial-logistic-regression-with-python/’.

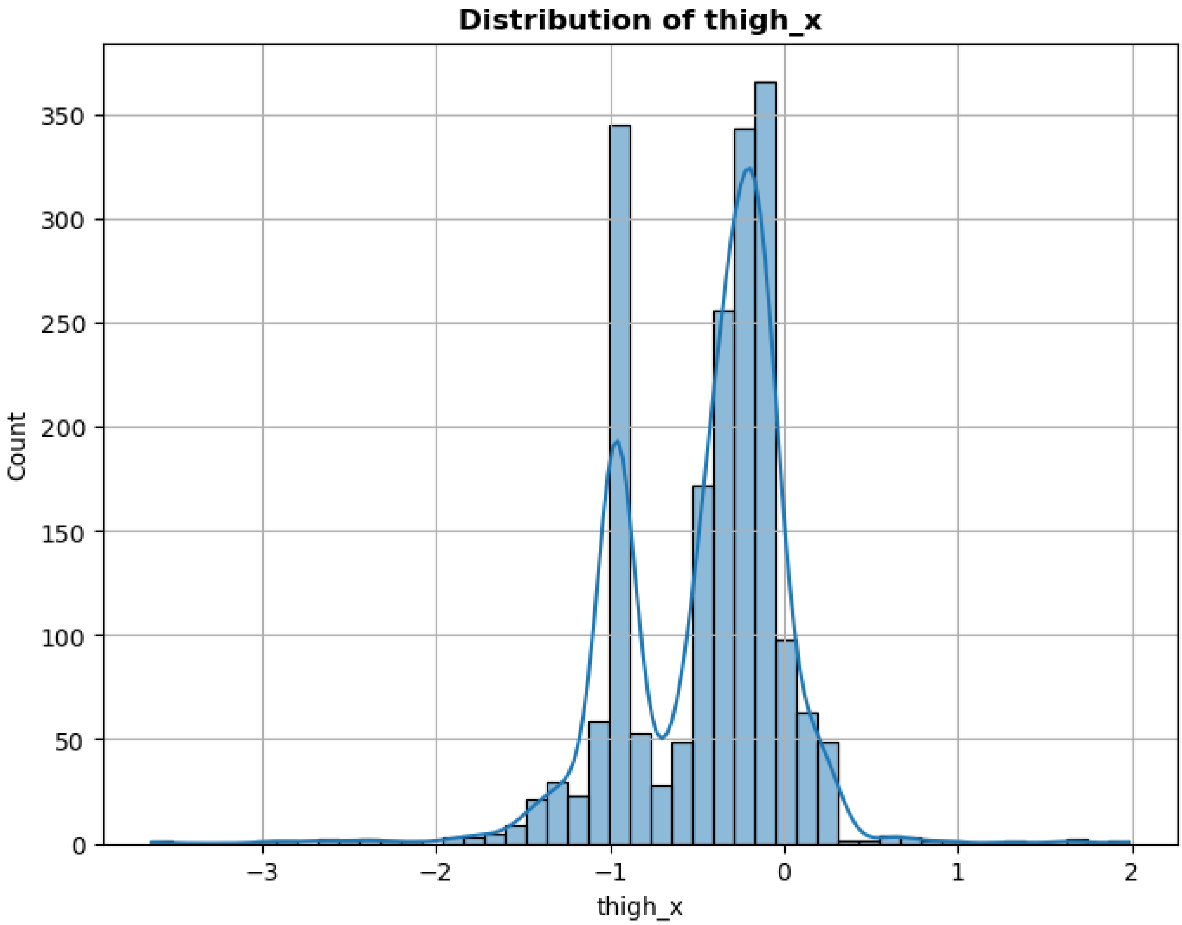
[10] ‘https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141’.

Appendix

Appendix 1. Distribution of Back X sensor data



Appendix 2. Distribution for the Thigh X sensor data


Distribution of thigh_x
Count
thigh_x