Implementation

# Implementation

# 

The implementation outlines the practical steps and processes taken to achieve the project's goals and objectives. This section describes how the project team utilized various tools and technologies to process data, explore different machine and deep learning algorithms, and evaluate model performance to achieve optimal results. It also highlights the Scrum framework employed to ensure adaptability, flexibility, teamwork, and stakeholder engagement throughout the project's duration.

### Deep Learning

Deep learning is a subfield of artificial intelligence, which is simply a neural network with three or more layers. These neural networks mimics how the human brain functions, however still far short of being able to match its capabilities, enabling it to "learn" from vast quantity of data. A neural network with only one layer can still make approximation predictions but an additional hidden layers can help to optimise and refine the accuracy result. Deep learning can automatically identify the collection of features that separate several categories of data from one another after ingesting unstructured material in its raw form (such as text or photos). This reduces the need for some human interaction and makes it possible to handle bigger data sets. [1]

### Machine learning

Machine learning is a subfield of artificial intelligence (AI) and computer science which focuses on using data and algorithms to simulate how humans learn, gradually increasing the accuracy of the system. It is usually more dependent on human intervention to learn. Human experts determine the set of features to understand the differences between data inputs, usually requiring more structured data to learn. [2]

# Algorithms Used – Deep Learning

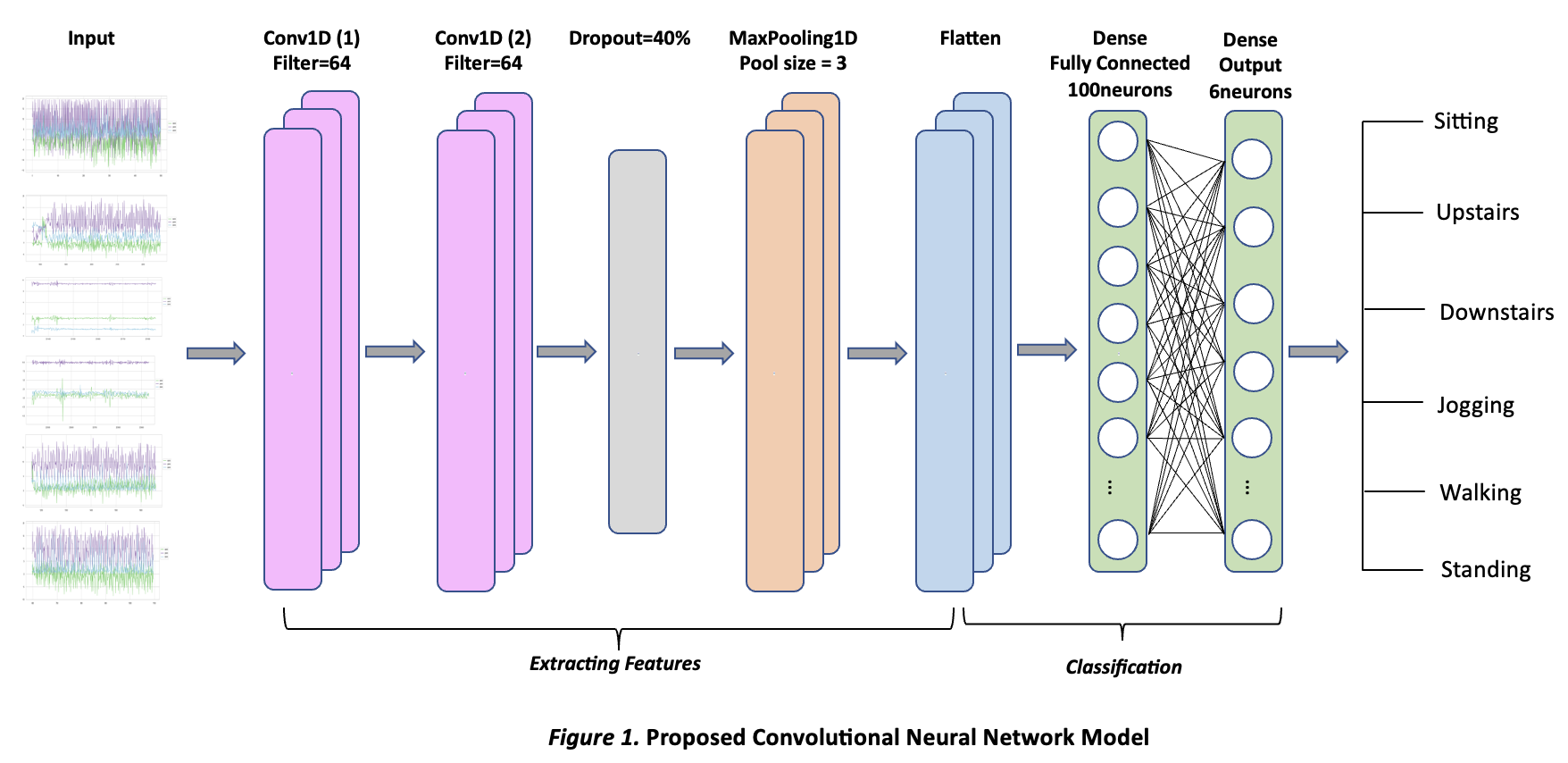
### CNN

**Definition**: Convolutional Neural Network (CNN) is a type of deep learning model that starting to gain attention nowadays due to its high capability of processing data to automatically and adaptively learn spatial hierarchies of features, from low-level to high-level patterns. [3]

### Methodology

In this research work, a customized 1D CNN deep learning model has been developed to detect Human Activity Recognition (HAR). The proposed overall flow topology is depicted in Figure 1. Six (6) categories of human movement were examined: walking upstairs and downstairs, jogging, sitting, standing, and walking. For the six-class dataset, a deep neural network-based bespoke CNN model architecture is examined to provide an improved detection outcome. Python 3.9.12 was used to create the feature extraction and detection algorithms. The GitHub was used as a repository to compile and document the model’s implementation. A Python package was used in conjunction with TensorFlow to implement the Keras model. The other Python libraries used for the model development, data training, and testing are Matplotlib, Scikit-Learn, Pandas, seaborn, and NumPy.

The architecture of the proposed model includes two (2) 1D convolution layers, 1D max pooling layer and two (2) dense layers. The accelerometer data are fed to the first two (2) 1D convolutional layers with sixty-four (64) convolutional filters and a kernel size of three (3), which is used to assist in the learning of more abstract and hierarchical characteristics of the data. The ReLU activation function is applied to its output to achieve its optimal features. The features are dropped first at forty percent (40%) to avoid overfitting before feeding to max pooling layer. Then, the obtained features are fed to one (1) max pooling layers, with a max pooling window size of three (3), which helps to downscale the features that is not in used and replaced it with the most important features and extracted for the next level of layer. The obtained important features are flattened before proceeding to a fully connected layer. Finally, two (2) dense layers with ReLU activation layers and softmax functions are used for the multi-class classification. A fully connected layer with hundred (100) neurons receives the output from the flattened layer and its output is then processed to the ReLU function. The softmax layer receives the output of the fully connected layer and computes a probability distribution over the six activity classes. Below diagram shows the proposed architecture of the model.



# Algorithms Used – Machine Learning

### Random Forest CLASSIFIER

**Definition**: Random Forest is a commonly used machine learning algorithm which combines the output of multiple decision trees to reach a single result. As it addresses both classification and regression problems, its wide adoption was due to its simplicity and adaptability functions. [4]

### Gradient Boosting Classifier

**Definition:** Gradient Booster Classifier is a machine learning technique commonly used for regression and classification tasks. It is an algorithm that builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n\_classes\_ regression trees are fit on the negative gradient of the loss function, e.g. binary or multiclass log loss. [5]

### Methodology

Python 3.9.1 is the programming language used to develop the machine learning model that detects the human activities as accurately as possible. Moreover, Python's support various modules and packages which promotes the modularity and reuse of codes in the program which easily make the coding smoothly executed.

### Data Preparation

The process starts by importing the various libraries, including NumPy, Pandas, Matplotlib, Seaborn, and regular expressions. These libraries are used for data processing, visualization, and machine learning tasks.

The text dataset "WISDM\_ar\_v1.1\_raw.txt" are read and imported using the Pandas library. The accelerometer data is measured along the x, y, and z axes, and the dataset includes a timestamp and a user ID for each data point.

The imported dataset is cleaned to ensure that the Machine Learning model will perform better. The data cleaning process involved dropping rows with incomplete data, removing symbols like semicolons, and transforming the data type from string to float.

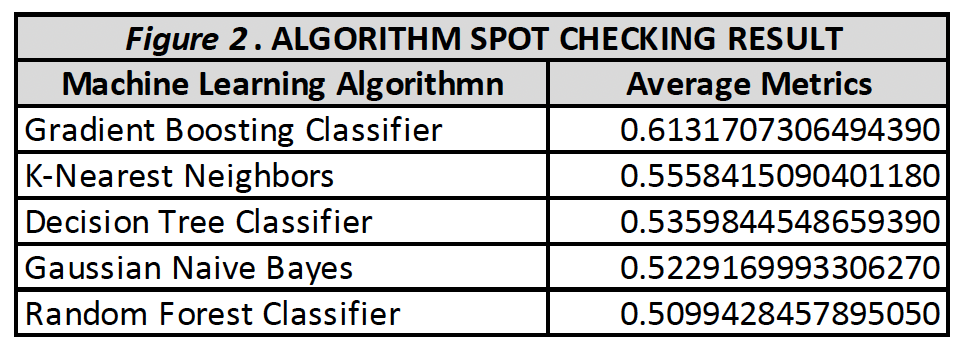
### Model Building and Training

The model was built by choosing the correct machine learning classifiers to solve the problems and features that go into the models. The first step of the model building is deciding what might be the appropriate machine learning classifiers to solve the problem based on the type and characteristics of the dataset. The WISDM dataset is a structured and classification type in which the applicable machine learning classifiers are the following- K Nearest Neighbor, Decision Tree, Naïve Bayes, Random Forest, and Gradient Boosting Classifier. The algorithms are imported from scikit-learn library to be used and fitted into the dataset.

*Algorithm Spot Checking*

A Spot-checking technique is applied in the applicable machine learning to provide a first set of results quickly and objectively based on the predictive modelling problem. [6] This will provide an idea if a problem is indeed predictable or not. The cross\_val\_score from sklearn’s model\_selection was used to calculate the average accuracy results. It gives an overview which Machine Learning algorithms will yield a high accuracy result.

Below figure 2 shows the average accuracy results of the spot checking done:



Based on the result, gradient booster classifier yields the highest mean accuracy rate, which interprets that the algorithm can predict 61% of the classification problem.

After conducting the step for Spot-checking of the algorithms, the train\_test\_split function from the Scikit-learn library is then used to split the cleaned dataset into training and testing sets. 80% of the data are in the training set, and 20% are in the testing set. The goal of dividing the dataset into a validation set is to avoid overfitting, which occurs when a model gets exceptionally good at classifying samples in a training set but cannot generalise and generate correct predictions.

*Hyperparameter Tuning*

Finding the optimal combination of a collection of hyperparameters to maximise the performance of the model is known as hyperparameter tuning. The GridSearch cross validation from Scikit-learn library is the approach taken in this research for the hyperparameter tuning. With this technique, a grid of hyperparameters' potential values is generated. The cross validation performs an exhaustive search over a specified range of hyperparameter values and returns the combination of hyperparameters that results in the best performance on the validation set (in this case, the training set). The results of the selected algorithm's best hyperparameters are as follows:

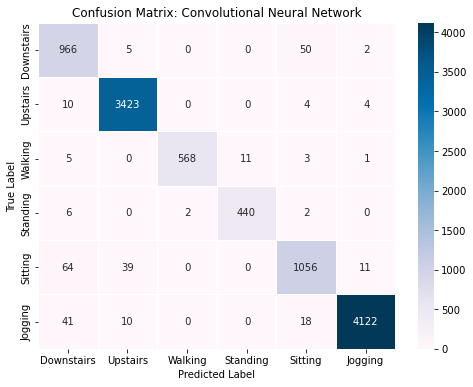
For Random Forest Classifier, the best hyperparameter for criterion is entropy. From 'criterion’: ['gini', 'entropy']

For Gradient Booster Classifier, the best hyperparameters for criterion and loss are friedman\_mse and deviance, respectively. From {'criterion': ['friedman\_mse', 'squared\_error'], 'loss': ['log\_loss', 'deviance', 'exponential']}

# Outcome Result, Model Evaluation and Visualisation

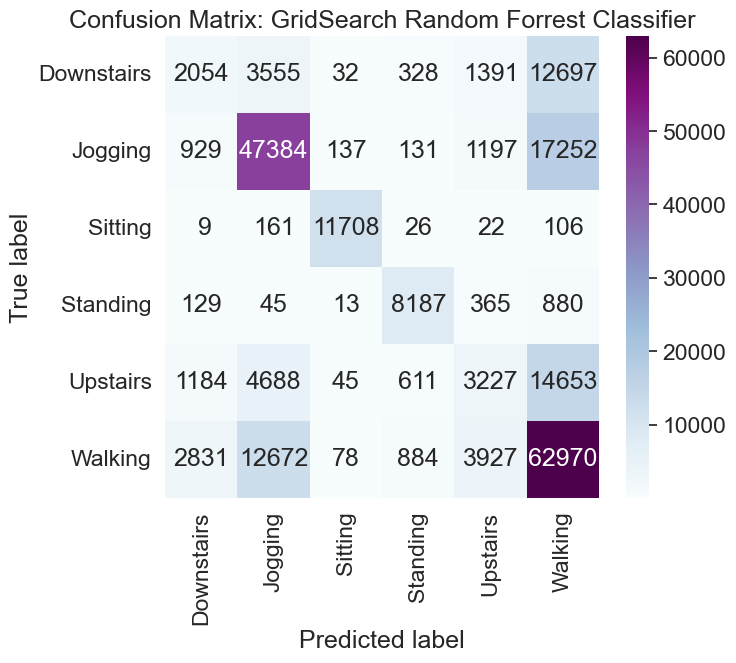
The machine learning models are evaluated by calculating the metrics such as accuracy, precision, recall and F1 score. These metrics provides statistical information on how successfully the algorithm can classify the activities based on accelerometer data. The results are further visually evaluated by classification report and confusion matrix. The classification report shows in a tabular form, the metrics result (accuracy, precision, recall and F1 score) for every variable. The metrics result is the measurement of the quality of predictions from the classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True Negatives and False Negatives are used to predict the metrics of the classification report. The confusion matrix on the other hand plots in a table all the predicted and actual values of a classifier.

### CNN



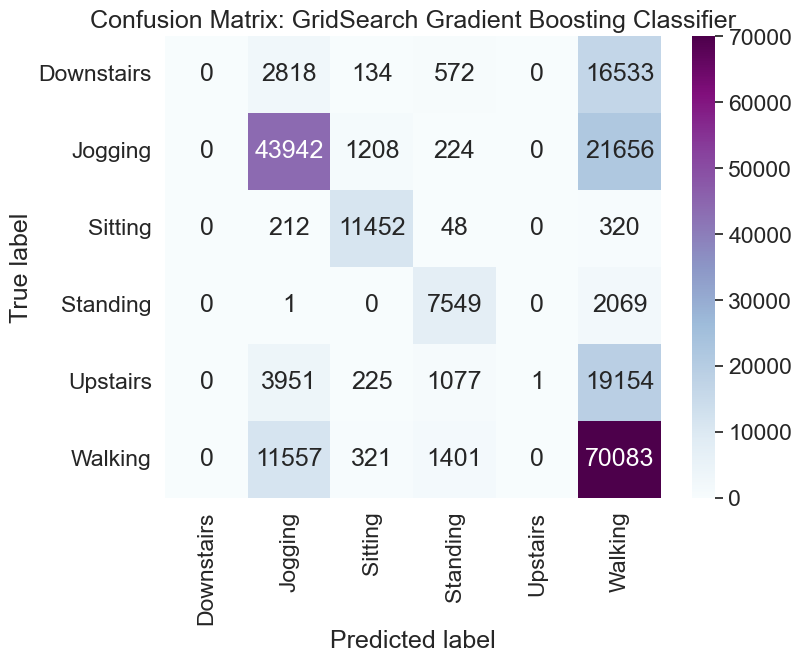
The performance metrics of the Convolutional Neural Network (CNN) algorithm for human activity recognition were evaluated based on the confusion matrix. The CNN algorithm achieved an accuracy rate of 97.45%, indicating that 97.45% of the human activities were predicted correctly. The precision rate of the algorithm was 97.50%, indicating that 97.50% of the positive predictions were accurate. The recall or sensitivity rate was 97.45%, indicating that 97.45% of the total positive class was predicted as positive. The F1 score of the algorithm was 97.45%, which is the harmonic mean of precision and recall/sensitivity. These metrics suggest that the CNN algorithm performed well in accurately predicting human activities.

### Random Forest CLASSIFIER



The matrix is used to evaluate the performance of a classification algorithm by comparing the actual and predicted values of the target variable. The Random Forest Classifier's confusion matrix indicates that out of the total number of human activities predicted, only 62.60% were predicted correctly, resulting in an accuracy score of 62.60%. The precision score indicates that only 59.00% of the positive predictions were accurate. The recall or sensitivity score indicates that out of the total number of positive classes, 62.60% were predicted as positive. The F1 score is 59.51%, which is the harmonic mean of precision and recall/sensitivity, and provides an overall measure of the classifier's performance. These metrics suggest that the Random Forest Classifier did not perform well in accurately predicting human activities.

### Gradient Boosting Classifier



Based on the confusion matrix of Gradient Boosting Classifier, the following metrics can be determined: The accuracy rate of the model is 61.44%, which means that 61.44% of the human activities are predicted correctly. The precision of the model is 70.95%, indicating that 70.95% of the positive predictions made by the model are accurate. The recall or sensitivity rate of the model is 61.44%, which means that 61.44% of the total positive class are predicted as positive. Lastly, the F1 score is 54.62%, which is the harmonic mean of Precision and Recall/Sensitivity.

REFERENCES:

[1] “What is deep learning?”. IBM.com.

<https://www.ibm.com/topics/deep-learning>.html (accessed 24 04 2023)

[2] “What is machine learning?”. IBM.com

<https://www.ibm.com/topics/machine-learning.html> (accessed 24 04 2023)

[3] R. Yamashita, M. Nishio, R. Kinh Gian Do and K. Togashi. Springeropen.com. <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9#:~:text=CNN%20is%20a%20type%20of,%2D%20to%20high%2Dlevel%20patterns.html> (accessed 24 04 2023)

[4] “What is random forest?”. IBM.com

<https://www.ibm.com/au-en/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.html> (accessed 24 04 2023)

[5] “sklearn.ensemble.GradientBoostingClassifier”. Scikit-learn.org.

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html> (accessed 24 04 2023)

[6] J. Brownlee. How to Develop a Framework to Spot-Check Machine Learning Algorithms in Python. MachineLearningMystery.com

<https://machinelearningmastery.com/spot-check-machine-learning-algorithms-in-python/> (accessed 25 04 2023)