Human Activity Recognition using Traditional and Deep Learning Models with TensorFlow Lite Optimization

By Azhar khan

1. Introduction

Human Activity Recognition (HAR) plays a crucial role in various applications, including healthcare, sports, and human-computer interaction. This project aims to explore and compare the effectiveness of traditional machine learning models and deep learning models for HAR using the HAR dataset. Additionally, we investigate the benefits of optimizing deep learning models using TensorFlow Lite, which enables efficient deployment on resource-constrained devices.
We have been able to achieve a accuracy of 96.30% using traditional machine learning model(logistic regression) on engineered features of 9 axis IMU data and an accuracy of – on CNN model on the time series data of the IMU sensor itself.And with 91.37% size reduction and only 0.33% reduction in accuracy.

2. Dataset

Human Activity Recognition, abbreviated as HAR, involves predicting a person's actions based on their movement traces using sensors.

One of the commonly used human activity recognition datasets is the "Activity Recognition Using Smartphones Dataset," which was made publicly available in 2012.

The dataset was provided by Davide Anguita et al. from the University of Genova, Italy, and is fully described in their 2013 paper titled "A Public Domain Dataset for Human Activity Recognition Using Smartphones." In their 2012 paper titled "Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine," they applied machine learning algorithms to model the dataset.

The dataset is accessible for free from the UCI Machine Learning Repository under the name "Human Activity Recognition Using Smartphones Data Set."

Data collection involved 30 subjects, aged between 19 and 48, who performed six standard activities while wearing a waist-mounted smartphone. The smartphone recorded x, y, and z accelerometer data (linear acceleration) and gyroscopic data (angular velocity) at 50 Hz,

resulting in 50 data points per second. Each subject repeated the activities twice, once with the device on their left-hand side and once on their right-hand side.

The raw data is not directly available, but a pre-processed version of the dataset was provided. Pre-processing included applying noise filters to the accelerometer and gyroscope data, dividing the data into fixed windows of 2.56 seconds (128 data points) with 50% overlap, and separating the accelerometer data into gravitational and body motion components.

Feature engineering was applied to the window data, resulting in a dataset with engineered features. They extracted a total of 561 time and frequency features commonly used in human activity recognition.

The dataset was split into training (70%) and testing (30%) sets based on data from subjects, with 21 subjects for training and nine for testing.

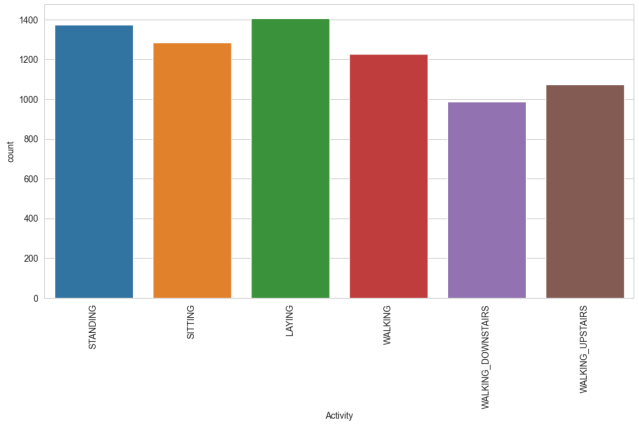
Experiments using a support vector machine intended for smartphone use (e.g., fixed-point arithmetic) achieved a predictive accuracy of 89% on the test dataset, comparable to an unmodified SVM implementation.

For researchers and developers interested in utilizing this dataset, it is freely available for download from the UCI Machine Learning repository.

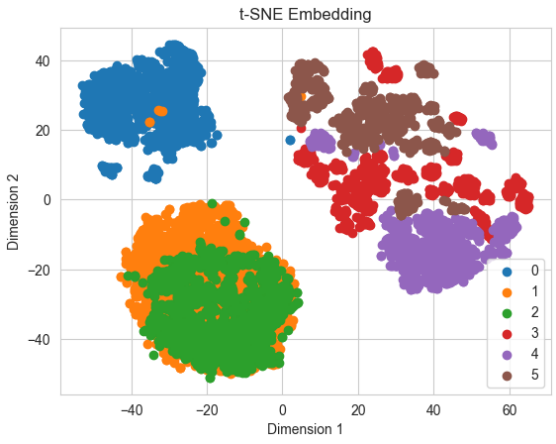
3. Preprocessing

For the traditional model we use the engineered feature dataset which has 561 features and 6 classes namely: Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing and Laying.

The Nun values were dropped and classes were categorically encoded.Then the class countplot was obtained and as we can see we have a balanced dataset.



3.1 CountPlot of classes



3.2 T-sne cluster of data

For deep learning models, there are three main signal types in the raw data: total acceleration, body acceleration, and body gyroscope. Each has 3 axes of data. This means that there are a total of nine variables for each time step.

Further, each series of data has been partitioned into overlapping windows of 2.56 seconds of data, or 128 time steps. These windows of data correspond to the windows of engineered features (rows) in the previous section.

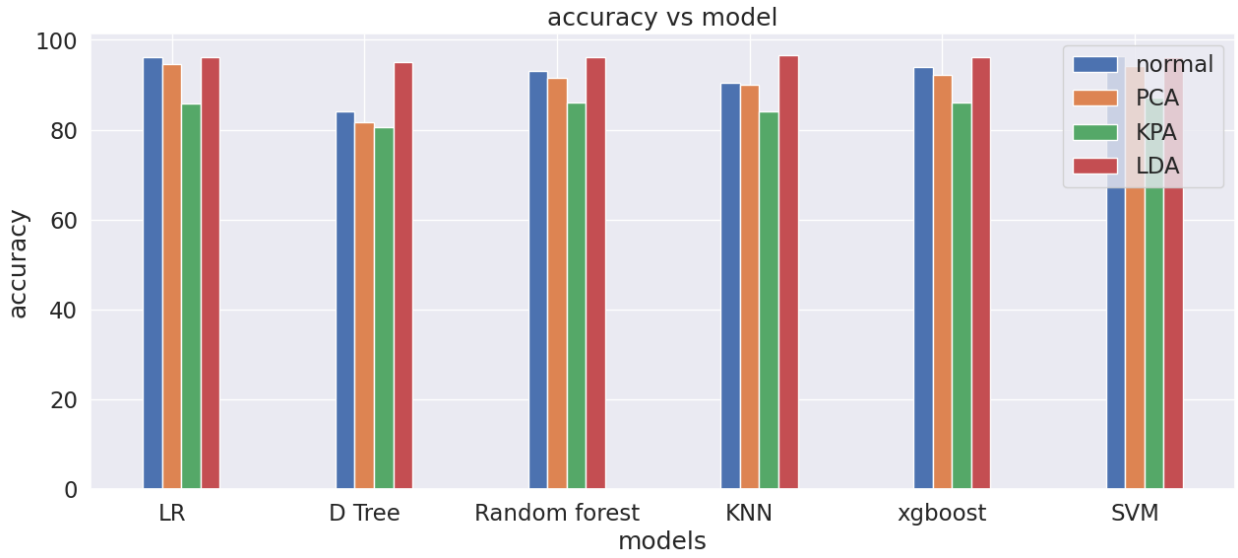
This means that one row of data has (128 * 9), or 1,152 elements. This is a little less than double the size of the 561 element vectors in the previous section and it is likely that there is some redundant data.And each row is in the form of a numpy array.

The data is visualized in a T-SNE cluster wrt to the classes in fig 3.2, and we can see that the data is fairly separable with some classes such as walking upstairs and downstairs having a major intersection.

4. Traditional model

Models used here are Logistic regression, SVM, Decision Tree, Random Forest, KNN, XGboost we have have also done dimensionality reduction using LDA PCA and KPCA and have creates three dataset similarly and have used various evaluation metrics such as accuracy score, Precision score, Recall, F1 score and have compared these metrics across all the three dataset.

The detailed metrics analysis can be found in the code.We have displayed the accuracy score plot of each model.

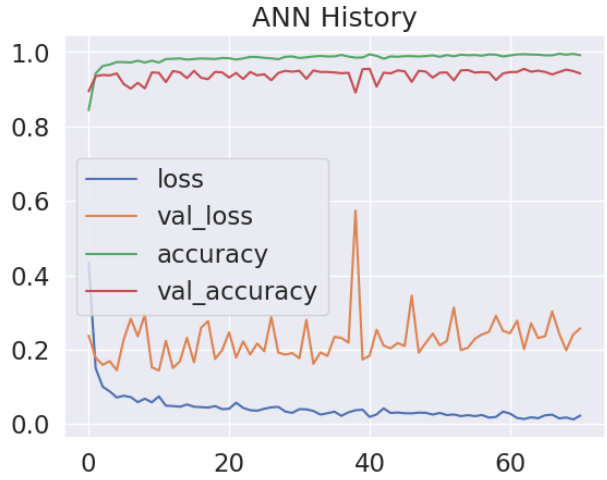


5. Deep Learning model

In deep Learning we have used ANN on engineered datasets, 1D CNN on the time series datasets and RNN.

The ann is 4 layered with architecture as Input(561),layer2(100),layer3(64),output(6).

We get an accuracy of 95.49% at 71 epochs.



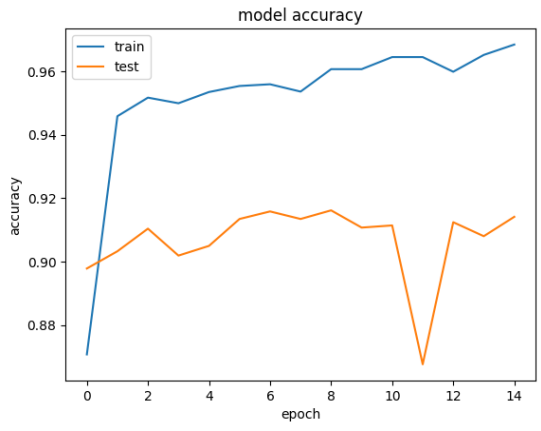
The above models are made on engineered features of the IMU sensor data and it takes lots of preprocessing to calculate the parameters hence they achieve such high accuracy scores

The further Models are made using the sliding window of the time series data from the sensor. Which is a more appropriate representation of the problem.

1D CNN architecture:

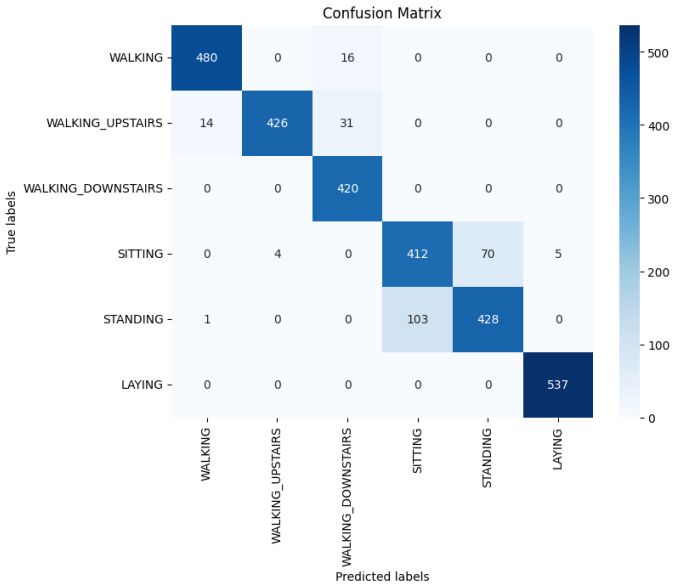
```
conv1d_4 (Conv1D),conv1d_5 (Conv1D),dropout_2 (Dropout), max_pooling1d_2
(MaxPooling 1D),flatten_2 (Flatten), dense_4 (Dense), dense_5 (Dense)
```

Model accuracy and loss vs epochs



The model performs with an accuracy of 91.72 % accuracy with a size of 4.694 MB.

CONFUSION matrix



As we can see we lose the majority of our accuracy in the Sitting and standing miss classification.

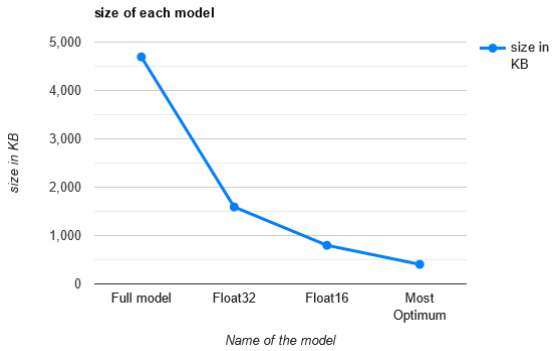
RNN model

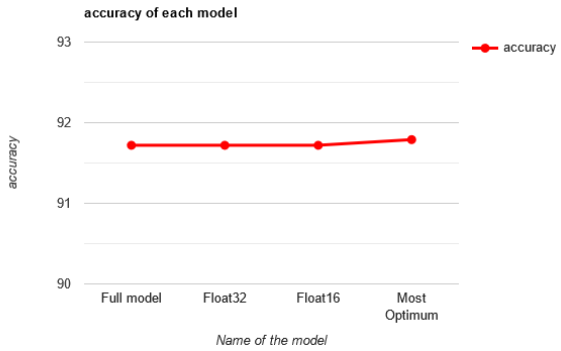
We were able to achieve 90.77% accuracy on the RNN model.

6. Model optimization using tensorflow lite

We have optimized the 1D CNN model as it performs the best on the sensor data and have compared the space occupied vs accuracy score of the model.

We have optimized the model using Quantization to float32 and float16 and then applied the best possible available optimization in the tflite library.





7. How does optimization work

There are various methods/techniques which enable us to run ML models on embedded devices. The most common/popular method is Quantization and pruning.

Quantization involves approximating a neural network model that uses floating-point numbers with a low bit-width number neural network. The goal is to reduce memory requirements, energy bandwidth, and computational costs for the neural network, making it suitable for deployment on resource-constrained devices.

However, quantization results in lower precision weights, which may lead to a loss of accuracy in the neural network. To address this accuracy loss, researchers are exploring various methods for improving quantization while maintaining acceptable performance.

8. Further path

In the project further we could try methods to train the model on the microcontroller itself rather than needing a host system to do so. In REFERENCE(3) they have tried to train a model on the board with 256KB SRAM limitations.

9. Conclusion

In this project, we delved into the fascinating realm of Human Activity Recognition (HAR) and explored the effectiveness of traditional and deep learning models for this critical task. By using a carefully curated dataset and employing various machine learning techniques, we sought to classify human activities accurately.

Our findings reveal that deep learning models, particularly Convolutional Neural Networks (CNN) , are able to infer real time sensor data and are capable of learning complex temporal and spatial patterns inherent in human activity data contributed to their superior real time performance.Where as traditional model excel in the engineered features.

However, we acknowledge that deploying deep learning models on resource-constrained devices poses challenges due to their computational complexity and memory requirements. To address this, we explored the advantages of using TensorFlow Lite, a framework optimized for efficient model deployment on mobile and edge devices. By converting and optimizing our trained deep learning models with TensorFlow Lite, we achieved substantial reductions in model size and inference speed without significant sacrifice in accuracy.

TFlite could have benefits like Enables real-time, efficient inference on edge devices, reducing model size, and conserving power for accurate and privacy and security.It has interesting

applications in Mobile Applications, Internet of Things (IoT) Devices, Autonomous Systems, Healthcare Applications, Edge Analytics and Gesture Recognition.

10. References

1. https://arxiv.org/ftp/arxiv/papers/2211/2211.04448.pdf

2. https://www.sciencedirect.com/science/article/pii/S1319157821003335

3. https://arxiv.org/pdf/2206.15472.pdf

4. https://arxiv.org/ftp/arxiv/papers/2101/2101.06709.pdf