IoT Devices - Human Activity Classification

AII-530-01-SP23 - Data Analytics and IoT

Rodrigo Burberg, Safiya Afrin, Muntasar Alsaray

Human Activity Recognition Models

In recent years, there has been a growing interest in utilizing machine learning (ML) to extract insights from Internet of Things (IoT) device data, especially in the domain of human activity recognition. Human activity tracking models have numerous real-world applications in fields such as healthcare, sports, and smart home automation. In healthcare, human activity tracking models can be used to monitor the physical activity of patients, providing insights into their health and wellness. These models can help in the early detection and prevention of diseases and conditions, such as cardiovascular diseases and diabetes, by analyzing activity patterns and detecting anomalies. In sports, activity tracking models can be used to analyze the performance of athletes, providing valuable insights into their training routines and progress. These models can also be used in sports injury prevention, by identifying the risk factors associated with certain activities and providing personalized recommendations. In smart home automation, activity tracking models can be used to provide personalized services to users based on their activity patterns, for example smart thermostats can adjust the temperature based on the user's activity. Additionally, these models can be used to detect when a user has been involved in an automobile accident and may be used to alert a corresponding emergency response unit. Overall, human activity tracking models have a wide range of applications and potential benefits to users, making them a promising area of research and development for ML and IoT technologies.

This paper presents the development of two ML models, a Recurrent Neural Network (RNN) and a Long Short-Term Memory (LSTM), trained on an IoT dataset that tracks human activity. The RNN model was used to classify the activity as stationary or dynamic activities, while the LSTM model was employed for multiclass classification, identifying different activities such as walking, sitting, and standing, running, going up and down the stairs. The dataset used in this study was collected using a wearable sensor and consists of time-series data for various sensor signals, making it well-suited for training and evaluating RNN and LSTM models. The results show the effectiveness of the proposed models in accurately classifying human activity, demonstrating their potential for real-world applications in fields such as healthcare, sports, and smart home automation. A real-world application of such models can be used to recommend a user to engage in physical activity after a prolonged period of inactivity has been detected.

Wearable IoT Devices and Sensors

Wearable IoT devices, such as smartwatches, fitness trackers, and smartphones, have become increasingly popular over the years due to their ability to collect and transmit data about their users. Wearable IoT devices have shown great potential for monitoring and tracking human activity, providing valuable insights into individuals' health and wellbeing. The Samsung Galaxy SII, used in this study as an IoT device, is a popular smartphone equipped with various sensors, including an accelerometer and a gyroscope. Pertaining accelerometers, they are sensors that measure acceleration, or changes in velocity, of an object in a specific location. An accelerometer sensor captures readings usually expressed in units of g, where 1 g is equivalent to the acceleration due to gravity (9.81 m/s). Accelerometers can be configured to measure changes in velocity in either one, two, or three dimensions. A gyroscope is another type of sensor that measures the angular velocity, or rotational rate, of an object around an axis. They usually capture readings in units of degrees per second (°/s) or radians per second (rad/s). Gyroscopes can be configured to capture recordings in one to three dimensions. In this case, the Samsung

Galaxy SII’s accelerometer and gyroscope were both configured to capture readings in the three orthogonal directions (x, y, and z).

IoT Device Computing Considerations

The benefits of using wearable IoT devices for human activity recognition include their portability, convenience, and the ability to collect continuous and real-time data, allowing for more accurate and comprehensive analysis of human behavior. This usually means that IoT devices generate and transmit huge amounts of data, which often overwhelm cloud-based systems resulting in latency and bandwidth constraints. Depending on the application of a particular IoT device, edge computing may be preferred rather than using the cloud. For example, if an IoT device that tracks human activity detects that a user was involved in accident the model we would ideally want the model to have no latency or bandwidth issues. For other applications where there is no urgent or real-time need to process data, fog or cloud computing can be used.

IoT Dataset: Smartphone-Based Recognition of Human Activities and Postural Transitions Dataset

The dataset used in this study was collected using a waist-mounted smartphone with embedded inertial sensors, which captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years, who performed a protocol of activities composed of six basic activities, including three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs), as well as postural transitions between the static postures. The dataset was pre-processed by applying noise filters and sampled in fixed-width sliding windows of 2.56 seconds with 50% overlap. From each window, a vector of 561 features was obtained by calculating variables from the time and frequency domain. The experiments were video-recorded to label the data manually, making the dataset suitable for supervised machine learning models. The obtained dataset was randomly partitioned into two sets, with 70% of the volunteers used for generating the training data and 30% for the test data. The features were normalized and are bounded within [-1,1]. Normalizing features in machine learning models is important because it ensures that all features are on the same scale, preventing certain features from dominating others and allowing for the effective comparison and combination of different features in the model. The units used for the accelerations and gyroscope were g's and rad/s, respectively. Overall, the dataset provides a comprehensive and diverse set of sensor signals and activities, making it well-suited for training and evaluating ML models for human activity recognition.

IoT System Diagram

For this project data from a Samsung Galaxy SII was used. Although the smartphone is capable of collecting all kinds of data, we are only interested in the gyroscope and accelerometer and the data transfer mechanisms/protocols.

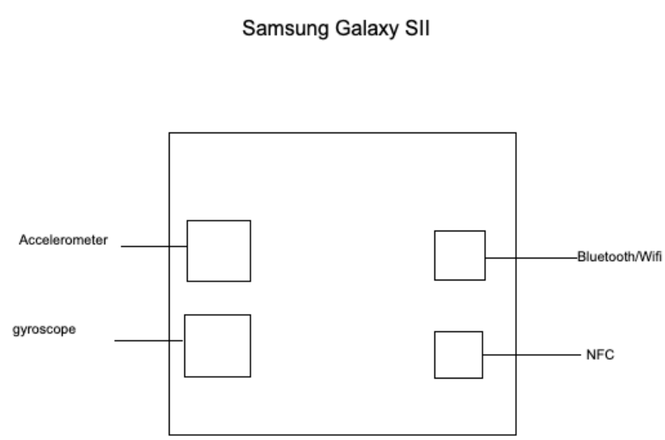

Figure 1

Tableau Visualisations

The following Tableau visualizations were produced to show the data being used for the models.

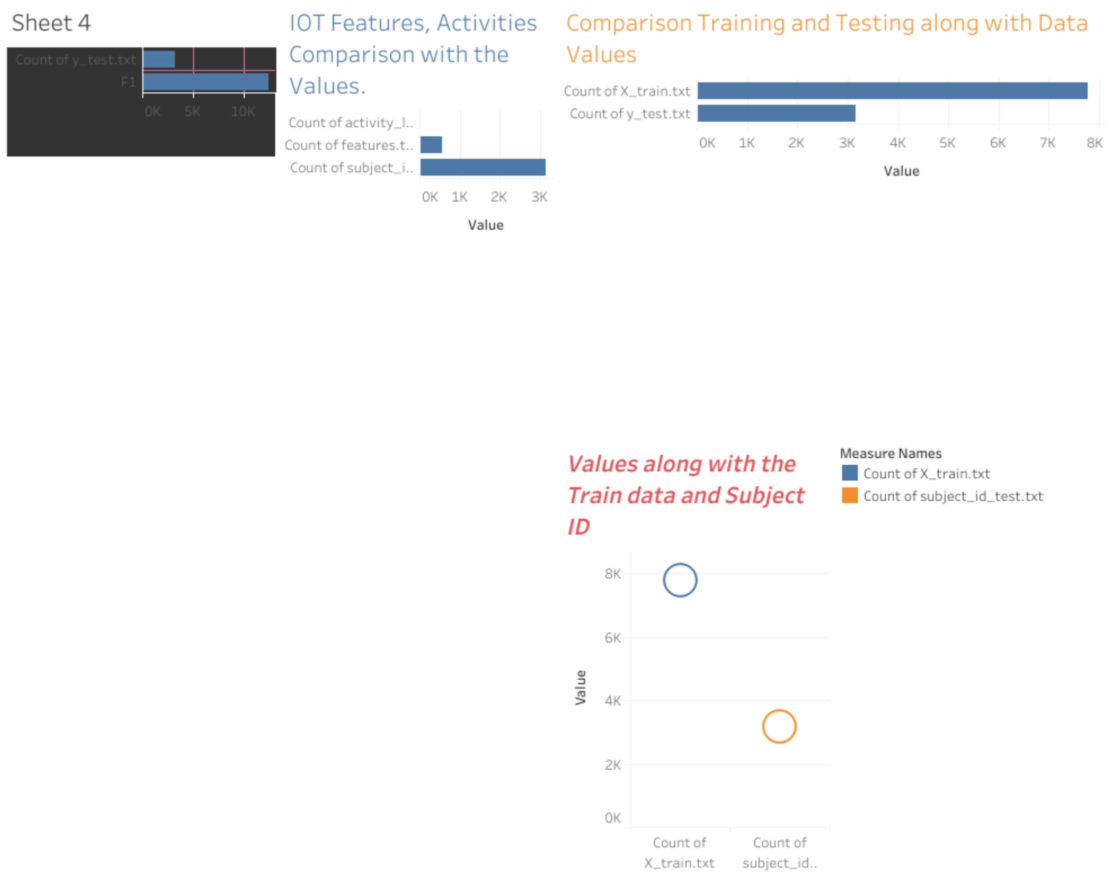

Figure 2

Time Series Models Used: RNN and LSTM

Time series data requires the consideration of temporal dependencies and patterns, making it an important type of data for ML modeling. Two popular ML models for time series analysis are the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM). In the case of the dataset used in this study, both models were used to classify human activity. The RNN model is a type of neural network designed to process sequential data, where the output at each time step is fed back as an input to the next time step. The RNN model was employed to classify the activity as stationary or non-stationary, by modeling the temporal dependencies of the sensor signals, which change over time during the various activities. However, traditional RNNs are prone to the vanishing gradient problem, where the gradients can become very small, making it difficult to train the network over long sequences. The LSTM model was developed to overcome this problem, by introducing a memory cell and three gating units that control the flow of information. The memory cell is responsible for storing information over long periods, while the gating units regulate the information flow, allowing the LSTM to selectively forget or remember information. The LSTM model was used for multiclass classification, identifying different activities such as walking, sitting, and standing. The LSTM model was able to model the temporal dependencies and patterns in the dataset, allowing it to accurately classify the different activities. While both models are effective for analyzing time series data, they differ in their ability to handle long-term dependencies. RNNs can model sequences of variable length and are more suitable for shorter sequences, while LSTMs are better suited for longer sequences and more complex data, where long-term dependencies are important. In summary, RNNs and LSTMs provide powerful tools for analyzing and modeling temporal data, and the choice of model depends on the specific problem and data characteristics.

Machine Learning models

For the first machine learning model a RNN model was used to classify the human activities as either static or dynamic. In total there are twelve human activities in the dataset, however not all activities can be easily classified as either static or dynamic since are transitional activities. Additionally, these activities present the problem of being a minority class making the dataset highly imbalanced, as we can see from Figure 3, shown below. There are essentially five methods for handling this imbalanced dataset, which are oversampling the minority class, under-sampling the majority class, deleting the minority class altogether, mapping or converting some transitional activities to another non-transitional activity such as: STAND_TO_SIT and LIE_TO_SIT as SITTING, or keeping the dataset as is. Due to the time constraints of the project, it was decided to use the dataset as is and proceed with training the model. While this approach may lead to some bias in the model towards the majority classes, it was found to be an effective approach given the scope and time constraints of the project. Although removing these transitional activities might seem like a valid approach to balance the class distribution in the dataset, this approach might lead to the loss of some valuable information. Mapping or converting some of the transitional activities to another non-transitional activity was not performed because some information may not conceptually make sense such as LIE_TO_STAND as STANDING since clearly a person must make a dynamic activity to stand up when laying down, thus this would introduce incorrect information in the STANDING feature which is a static activity.

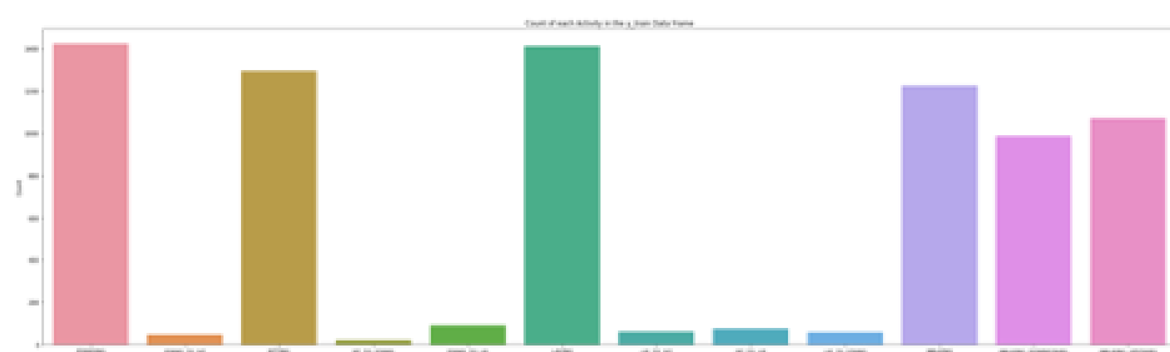

Figure 3

In order to proceed with the model some mapping of the features was performed. Additionally, since it was decided that the transitional activities would remain as is, it was also decided to group them together as another class called POS (Postural transitions). Thus, the following mappings were performed: STAND_TO_SIT, LIE_TO_SIT, SIT_TO_LIE, STAND_TO_LIE, LIE_TO_STAND, and SIT_TO_STAND to POS (for postural transitions), WALKING, WALKING_UPSTAIRS, and WALKING_DOWNSTAIRS to MOT (for dynamic), and SITTING, STANDING, and LAYING to SIT (for static). Before this mapping was performed, a dimensionality reduction technique called T-SNE (t-distributed stochastic neighbor embedding) was perfumed to cluster these activities and visualize them in two dimensions from the 561 feature vector. Figure 2 (shown below) displays the clusters of the different activities. Based on the cluster graph, it can be observed that static activities (Standing and Sitting) are clustered together, while dynamic activities (Walking_Upstairs, Walking_Downstairs, and Walking) are closely clustered. Laying (static activity) is found in a separate group and located on the positive x-axis side along with Sitting and Standing. It also validates the decision to group the transitional activities rather than deleting them or convert them to a non-transitional activity. Figure 4 (shown below) displays the mapped features and shows the mapped features are different. The data was already normalized in a range of -1 to 1 and was already split for the train and test sets, thus ready for the models.

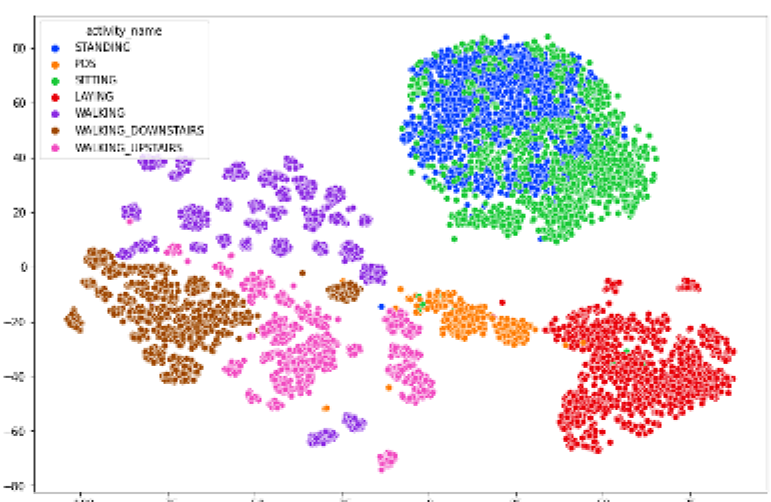

Figure 4

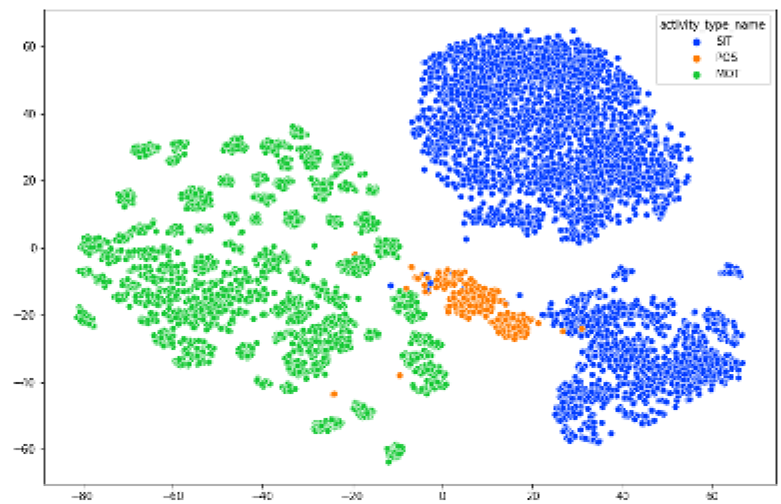

Figure 5

For the RNN model, a simple model with used consisting of an RNN layer with 64 units, a dropout layer set at .5 to prevent overfitting of the model, and the output layer having 3 units and a sigmoid activation function. The model showed very good results with a training accuracy of 100% and a validation accuracy of 99.25%. Figure 6 shows the accuracy graph of the training and validation. While it is difficult to increase the validation accuracy since it is already high, more data can be added, or a more complex model can be implemented.

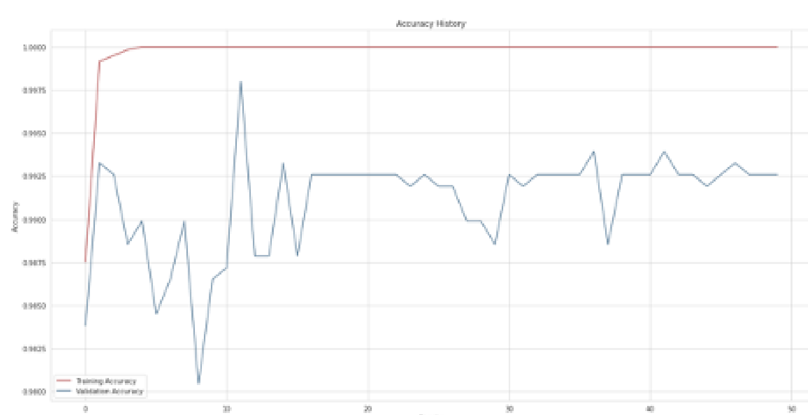

Figure 6

For the LSTM model, the same approach was used. The architecture of the model was as follows: 1 input LSTM layer consisting of 128 units, 2 hidden layers consisting of 1 dropout set at 0.5 layer to prevent overfitting a dense layer with 64 units with a ReLU activation function, and an output layer having a softmax activation function. The model was compiled using the loss function loss equal to categorical_crossentropy, the optimizer equal to adam, and the metric being accuracy. Overall, the LSTM model provided good results with a train accuracy of 86.99% and a validation accuracy of 87.27%.

References

Galaxy S II teardown – splitting 8.9 mm of the latest Samsung Technologies. – Samsung Global Newsroom. (n.d.). Retrieved February 27, 2023, from https://news.samsung.com/global/galaxy-s-ii-teardown-splitting-8-9-mm-of-the-latest-samsung-technologies

Dixon, M., & London, J. (2020, October 13). *Financial forecasting with α-RNNS: A time series modeling approach*. Frontiers. Retrieved February 27, 2023, from https://www.frontiersin.org/articles/10.3389/fams.2020.551138/full

Understanding LSTM networks. Understanding LSTM Networks -- colah's blog. (n.d.). Retrieved February 27, 2023, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/