Multimodal Sensor Data for Human Activity Recognition

Lim Teow Yong, Heng Yong Sheng, Yap Ming Hui Milton, Low Chen Ni

***Abstract*—Human activity recognition is one of the most important tasks in pervasive computing. It recognized action and goal for a person from series of observation on the person's action. This paper describes the usage of multimodal data collected from Kinect sensor and wearable inertial sensor to predict human activities. Data preprocessing techniques such as zero padding, resampling and filtering were applied to the datasets to ensure the data are in the right form. Raw Coordinates of skeleton data were transformed into Displacement Vectors and resulted in significant improvement in model accuracy. Fast Fourier Transform was applied on the inertial data to select top 5 frequencies to be used in Random Forest classifier training. Depth Motion Map was used to reduce the pixel features from Depth data prior to XGBoost training. 3 different deep learning techniques, LSTM, GRU and TCN were used to train classifiers using skeleton data. Ensemble method is used as an attempt to improve the model performance. Finally, we compared the performance of different techniques to select the best model for Human Activity Recognition classification.**

***Index Terms*— Depth Motion Map, Fast Fourier Transform, Gated Recurrent Unit , Human Activity Recognition, Long Short Term Memory, Random Forest, Temporal Convolutional Networks, XGBoost**

I. Introduction

HUMAN activity recognition (HAR) is a set of techniques aims to recognize the action and goal for a person from a series of observations on the person's action. HAR gains its popularity because it can potentially be used in proactive computing that can anticipate people's necessity in situation such as healthcare of lifecare and take appropriate action on their behalf. It can also be used in smart homes and surveillance system.

Early techniques focused on the processing sequence of images captured by cameras. In [4], the human body was represented in terms of silhouettes extracted from camera images. Author applied Fourier analysis to describe the human silhouettes and Support Vector Machine to classify them into different postures.

Microsoft Kinect was initially designed for Xbox gaming and entertainment consoles. The Microsoft Kinect launch in November 2010 has enabled motion recognition through its commercially priced sensors. With the combination of RGB-Depth camera sensor and volume sensor, Kinect allows Xbox game players to interactively control the console through body gestures and voice commands without using any other peripheral equipment. Motion recognition is the fundamental enabling technology were the semantics of a human gesture or action can be interpreted automatically. [3]

Various methods had been used by other researchers to process skeleton data prior to activity classification. Chen and Koskela [5] used normalized 3-D joint positions as features, through a common coordinate system. Another form of representation is displacement-based, where the displacement between every pair of skeleton joints is calculated, rescaled to account for variations in human body sizes and rotated to account for different orientations of the body with respect to the camera [6]. Li et al. [7] proposed the encoding of pair-wise distances of joints into joint distance maps (JDMs).

Several neural network architectures had been used in the literature to perform activity classification using skeleton data, such as the Extreme Learning Machine (ELM) [8], recurrent neural networks (RNN) [9]-[11], and convolutional neural networks (CNN) [7]. Juan et al. proposed a combination of a CNN and a Long Short-Term Memory (LSTM) recurrent network for skeleton-based human activity and hand gesture recognition [12]. The LSTM which was introduced by Hochreiter & Schmidhuber [13] is an improvement over the RNN architecture by being capable of learning long-term dependencies and preventing the vanishing gradient problem. Damla and Abdelhamid [14] explored the use of gated recurrent units (GRUs) in addition to LSTMs for activity recognition and abnormal behavior detection for elderly people with dementia. GRUs, introduced by Cho et al. [15] are similar to LSTMs but they have fewer parameters. In recent years, it has also been shown that temporal convolutional networks (TCNs) which use dilated causal convolutions are able to outperform LSTMs, GRUs and RNNs in a number of sequence modeling tasks [16].

Many researches have also used depth information in activity classification. The advantage of depth information is mainly due to its insensitively to light changes, which allow for a more distinguish activity movement in the 3D images [17]. Xia et al used Kinect depth map to extract the 3D skeleton joint location for classification through a discrete hidden markov model technique [18]. Li et al employed an action graph from sequence of depth maps for fuzzy clustering [19]. Chen et al presented a human action recognition method using depth motion maps which maps sequence of depth images into 2D projected image for classification [20].

II. Data Definition

The UTD-MHAD dataset is used for this project with activity data captured by Kinect and wearable inertial sensor. The dataset contains 27 actions performed by 8 subjects (4 females and 4 males). Each subject repeated each action 4 times.

The downloadable dataset consists of MATLAB files with conventions:

- "a{i}_s{j}_t{k}_depth.mat" for depth data
- "a{i}_s{j}_t{k}_inertial.mat" for inertial data
- "a{i}_s{j}_t{k}_skeleton.mat" for skeleton data

where (i, j, k) refers to integer indexes of (action, subject, trial). The SciPy Python library is used to import mat file to retrieve the multi-dimensional array data.

Depth data is made up of multiple frame black and white images with resolution 240x320 pixels. Inertial data were collected from inertial sensors which contains 3-axis acceleration and 3-axis rotation signals. Skeleton data features indicate the angle between human joints and Cartesian axes. Summary of the data types is listed in Table 1.

TABLE 1
DATA TYPES SUMMARY

Type	Dimension	Max Timesteps	Feature
Depth	240x320xframes	125	BW Pixels
Inertial	Samples x 6	326	Acceleration & rotation
Skeleton	20x3xframes	125	Joint orders and positions

III. METHODS AND MATERIALS

The recognition task can be classified into 3 steps: Feature Extraction, Model Learning and Performance Evaluation as illustrated in Figure A. Firstly, the data from UTD-MHAD is processed by zero padding, resampling and filtering to make sure data are in the right form for the subsequent step. Skeleton data is transformed from raw coordinates to displacement vector to improve the performance of the model. 70% of the data is randomly selected for training and the remaining 30% is used for testing. LSTM, TCN, GRU, Random Forest, XGBoost and ensemble techniques are used to train the classifier and finally the model will be tested using the test dataset.

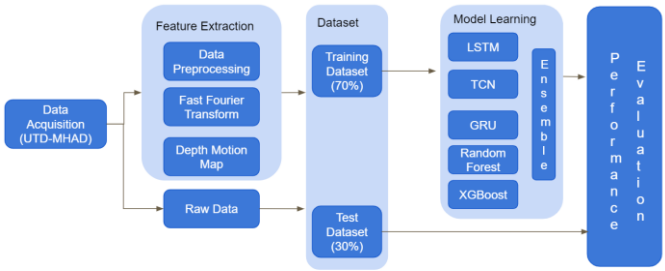

Figure A: Steps of Human Activity Recognition

Detail of each step in Human Activity Recognition is discussed in this section.

A. Feature Extraction

1) Data Preprocessing

a) Zero padding

For the skeleton dataset, the zero padding was performed in order to ensure that each subject-activity-trial combination has the same number of frames, to facilitate modeling. The maximum number of frames in a trial was found to be 125 and thus padding with leading zeroes was done to ensure each sequence was fixed at the same length as the maximum number of frames.

b) Resampling

The inertia sensor data, just like the other data modalities, possess varying number of frames due to the different time length taken by the subjects to perform an action. The timeframe was normalized by first scaling every waveform to a time period of 3 seconds. Next, the graph was resampled at a rate of 0.01 seconds, with using linear interpolation to fill in the missing values of the new sampled frequency. This creates a uniform data set of 300 points for graph plotting per sensor for all activities, subjects and trials.

c) Filtering

Noise is present in the raw sensor data that is due to miniscule vibrations and variations in muscular motion. A bandwidth filter with a bandpass frequency between 0.1 Hz and 5 Hz was implemented to filter out any transient noise from the signals.

d) Signal decomposition

Further processing can be performed on the inertia sensor data by converting it from the time domain to the frequency domain. A fast fourier transform (FFT) algorithm is used to decompose the signal into its constituent frequencies. Treating each sequence of activity as a single cycle, Fourier analysis is able to identify the major frequencies of each signal by looking at the amplitude, phase and frequency. By taking only the first five components of the fourier transform for machine learning classification, we can reduce the feature dimensionality of the dataset.

e) Transforming Raw Coordinates to Displacement Vectors

The skeleton data was stored as raw 3-D coordinates of 20 joints. We transformed these into 19 displacement vectors from a selected reference joint as performed in [6]. It makes the features location-invariant by disregarding the position of the subject and making the joint locations relative to the reference joint. The reference joint was selected to be the hip centre with coordinates (x_0, y_0, z_0) (Figure B) due to it being a central position of the body. The displacement vectors were then divided by the distance d between the shoulder centre and hip centre to make the features size-invariant, by recognizing that different humans have different builds. The simple transformation is illustrated in the formula below, where i is the joint index and $1 \leq i \leq 19$ (excluding the hip centre).

$$(x_i', y_i', z_i') = (\frac{x_i - x_0}{d}, \frac{y_i - y_0}{d}, \frac{z_i - z_0}{d})$$

This results in a total of 19*3 or 57 features for training.

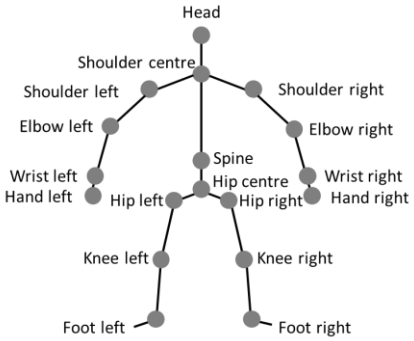

Figure B: Skeleton Joints

2) Sensor waveform classification

Sensor data for 6 degrees of freedom consisting of 3 dimensional acceleration and 3 dimensional angular velocity was sampled at a rate of 50Hz[22]. The measuring range of the accelerometer is ±8g and ±1000 degrees/second for the rotation sensor.

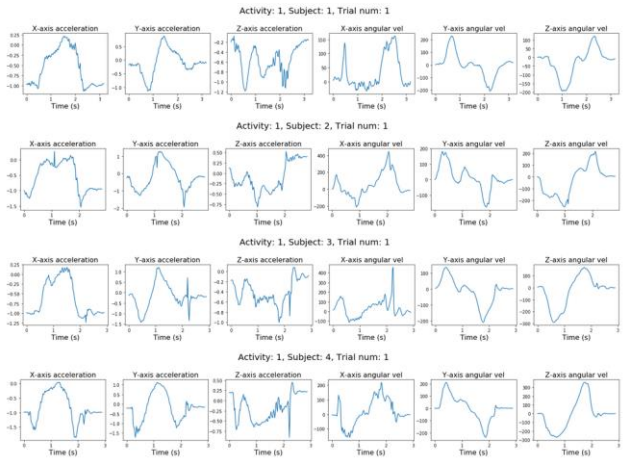

Figure C: Waveforms plotted for same activity, different subjects

Human daily activities, such as swiping arm, sitting, walking or jogging, have distinct movement patterns. In the figure above, graphs of four different subjects executing the same activity are plotted for each degree of freedom recorded by the inertial sensor. Even though the actions are performed by different persons with varying timeframes, similarities exist in the shape and waveform of the individual axis of movement, enabling classification of the activity based on signal data.

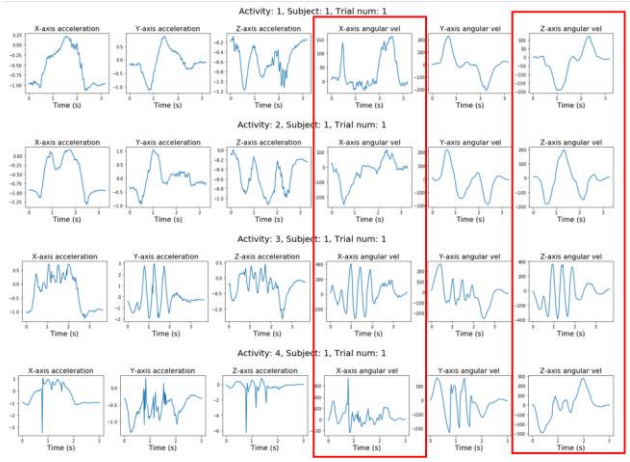

Figure D: Waveforms plotted for different activity, same subject

Conversely, different activities performed by the same subject yielded different peak frequencies as highlighted in the red bounding boxes of Figure D above. As the sequence of every waveform has been standardized to a 3 seconds cycle, the number of parameters for the LSTM and GRU is 1800, as there are 300 time-steps and 6 features.

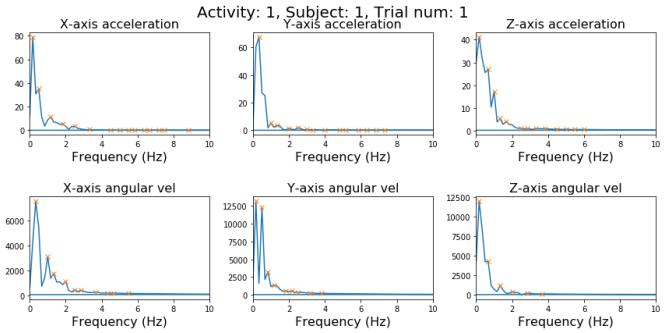

Figure E: Fourier analysis of a single activity

Next, attempts were made to improve the results from the time series recurrent neural networks by transforming the signal from the time domain to the frequency domain. The FFT analysis of an activity is shown in the Figure E above. The first five amplitudes and frequencies of the signal FFT decomposition, power spectral density and auto-correlation were extracted into a sample of 180 features each.

3) Depth Motion Maps

The depth data was captured in 240 x 320 pixels image, with varying number of N frames per activity and trial. We adopted the Depth Motion Maps (DMMs) discussed in [20] to map the difference in depth for each activity and trial into 2D orthogonal projection by the x, y and z coordinates. DMMs are used in the feature extraction process to reduce the number of pixels features required while retaining the depth information from the depth dataset. The DDMs are obtained using the formula:

$$DDM(x,y,z) = \sum|mapi\{x,y,z\} - mapi+1\{x,y,z\}|$$

where mapi represent the 240x320 pixels image in frame index i. Similarly, a bounding box was identified in each DDM where region outside the boundary box were trimmed off. The bounding box image is then resized into a 120x160 pixels image to be used as training features.

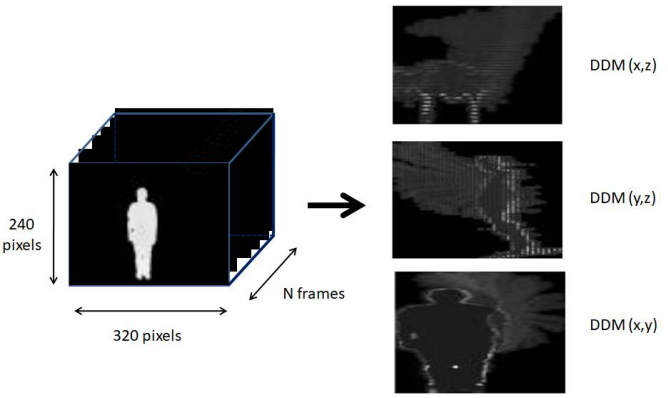

Figure F: 3 DMMs represent an action "Wave"

DDMs have effectively reduced the number of pixels feature required in the training model from a maximum of 192,000,000,000 (240x320x125frames) to 57,600 (120x160x3DDMs).

B. Model Learning - Classification Techniques

Classification is the allocating of elements to classes according to their characteristics. These characteristics are called features. In principle, elements with similar features belong to the same class [2]. The below techniques were used to classify activities into one of the 27 activities in the UTD-MHAD dataset.

1) XGBoost

XGBoost is a gradient boosting machine learning model which focus is on computational speed and model performance. In every iteration, a new model is trained to predict the error of prior model. The models were then ensembled together to make the final prediction. The gradient boosting used with decision tree improves the quality of fit for a fine balanced between bias and variance.

2) Random Forest

Random forest is an ensemble learning method for classification, regression and other tasks. A multitude of decision trees are constructed at training time which will then output the class for classification. Furthermore, random forest can mitigate for decision trees' habit of overfitting to their training set.[23]

3) Long Short-Term Memory (LSTM)

Traditionally, RNNs are used for classification of sequences and forecasting. However, due to the problem of vanishing gradients where the gradient of the loss function decays exponentially with time, they are only able to handle short-term dependencies. The LSTM is a special kind of RNN which is able to learn long-term dependencies [13] through the use of memory cells and a set of gates (input, output, forget) to control when information enters the memory, output, and forgotten.

4) Gated Recurrent Unit (GRU)

GRUs are similar to LSTMs but use a simplified structure with fewer gates (reset and update gates). The GRU controls the flow of information like the LSTM unit, but without having to use a memory unit. It simply exposes the full hidden content without any control. GRUs are also computationally more efficient than LSTMs. However, there is no concrete evidence that they perform any better or worse than LSTMs [21].

5) Temporal Convolutional Networks (TCN)

Recent research has shown that the use of a new neural network architecture, known as the TCN, is able to outperform baseline RNN-based architectures across a range of sequence modeling tasks [16]. Compared to LSTMs and GRUs, TCNs exhibit significantly longer memory and are suitable for use cases where a long history is required to be maintained in memory. They are also computationally faster and require lower memory as compared to LSTMs and GRUs [16].

C. Performance Evaluation

The dataset is divided into 70% training and 30% testing randomly. Model from model learning step is scored on the testing data to determine the model performance.

IV. Experimental Setup

Experiments were performed to evaluate the performance of the proposed data processing and modeling approaches on activity classification using the inertia, depth and skeleton data in UTD-MHAD dataset. Ensemble techniques were also evaluated on their classification performance. In each experimental setup, the dataset is randomly split into a training set and a test set in the ratio of 70:30. For performance evaluation, accuracy was used, along with other supplementary metrics such as precision, recall and F1-score.

A. Inertia

The experimental setup for activity classification using inertia data is segmented into two stages, first, LSTM, and GRU models were chosen as they are able to perform time series classification. Their performance on the normal resampled interpolated data is tested against the bandpass filtered resampled interpolated data to see if filtering techniques are able to improve the sensor classification results.

Next, the sensor data is converted from the time domain to the frequency domain. The first 5 peaks of the fast fourier transform, the power spectral density and auto-correlation are extracted from each signal to form a sample of 180 features. A random forest classifier is then trained on the sample set. The parameters for the neural network and random forest classifier are shown in Table 2.

TABLE 2
PARAMETERS FOR INERTIA MODEL

Model	Parameters
LSTM	Hidden nodes: 100 Timesteps: 300 Input dimensions: 6 Max epochs: 500 Total parameters (raw): 1800
GRU	Hidden nodes: 50 Timesteps: 300 Input dimensions: 6 Max epochs: 500 Total parameters (raw): 1800
Random Forest Classifier	Input dimensions: 180 No. of estimators: 1000

B. Depth

Because of the large number of pixels features from depth data, XGBoost model was chosen to predict the activity performed by the subjects in each trial using the 3 DDMs as explained in the previous section. XGBoost model was chosen due to its scalability to process large dataset in a fast and efficient way. The model was integrated into scikit-learn for processing in python. The parameters used is as shown in Table 3. Softmax activation function was used to minimise log loss.

TABLE 3
PARAMETERS FOR DEPTH MODEL

Model	Parameters
XGBoost	Booster: gbtree Objective: multi:softmax\ Classes: 27 Learning rate: 0.1 Max depth: 4 Max epochs: 100

C. Skeleton

The experimental setup for activity classification using skeleton data is as follows: The LSTM, GRU and TCN models were chosen to predict the activity performed by the subjects in each trial using the entire sequence of frames in the trial, with zero padding done as explained in the previous section. These models were chosen due to their ability to handle sequences as inputs and developed using Keras, an open source neural network library written in Python, running on a Tensorflow backend. Two different sets of results were obtained, by using the raw 3-D coordinates without feature engineering and using the transformed displacement vectors as shown in the previous section. The parameters used for both sets as shown in Table 4. Early stopping was used such that the training stops when the validation loss does not improve after 30 epochs, so as to prevent overfitting. The best model with the lowest validation loss was selected.

TABLE 4
PARAMETERS FOR SKELETON MODEL

Model	Parameters
LSTM	Hidden nodes: 50 Timesteps: 125 Input dimensions: 57 Max epochs: 10,000 Total parameters (raw): 23,628 Total parameters (transformed): 23,028
GRU	Hidden nodes: 50 Timesteps: 125 Input dimensions: 57 Max epochs: 10,000 Total parameters (raw): 18,078 Total parameters (transformed): 17,628
TCN	No. of filters: 50 Kernel size: 20 No. of stacks of residual blocks: 1 Dilation factors: 1, 2, 4 Max epochs: 10,000 Total parameters (raw): 162,278 Total parameters (transformed): 162,128

D. Ensemble

The best models built using each of the modalities (inertia, depth, skeleton) were identified and chosen as candidates for developing an ensemble model. This was done to assess if predictions from models built using different feature sets can be combined to make an even accurate prediction of the activity. Ensemble approaches considered were: (1) Soft Voting of all 3 models; (2) Soft Voting of top 2 models; (3) Stacking all 3 models using multinomial logistic regression; (4) Stacking top 2 models using multinomial logistic regression. Soft voting involves computation of the average of the predicted probabilities from the base models involved and predicting the target class based on the class with the highest average probability. Stacking was performed by using predicted probabilities from the base models as inputs to a meta-classifier (i.e. multinomial logistic regression) to predict the target class.

V. RESULTS

A. Inertial

Initially, the team used zero padding to fill in the front of the sensor sequence for the neural network models, which resulted in accuracies of around 40%. Next, the data was scaled, interpolated, and resampled in order to retain the same waveform on the graph in the sensor sequence. As shown in Table 5, the accuracy of the LSTM was 56.92% for the test set, and the accuracy of the GRU neural network was 65% on the resampled data.

Next, a bandpass filter with a filter bandpass of 0.1 Hz to 5 Hz was used to smoothen the waveforms and remove noise from the results. Accuracy on the test set for the LSTM improved from 56.92% to 66.15%, and from 65% to 77.69% for the GRU model.

TABLE 5
PERFORMANCE OF INERTIAL MODELS

	Resampled data		Bandpass Filtered	
Model	Accuracy (Training)	Accuracy (Test)	Accuracy (Training)	Accuracy (Test)
LSTM	88.58%	56.92%	75.81%	66.15%
GRU	80.13%	65%	100.0%	77.69%
Random Forest with signal decomposition	FFT, PSD, Auto-Correlation			
	100.0%	93.46%	100.0%	95.0%

Finally, signal processing techniques were applied on the sensor waveform to extract salient information on the signal. As the sensor waveform is decomposed into its constituent signals in the frequency domain without time sequence features, a random forest classifier is used instead. The accuracy of the random forest classifier was 93.46% on the signal decomposition of the resampled data, and 95.0% on the signal decomposition of the bandpass filtered resampled data.

The best results of the random forest classifier (bandpass filtered) is shown in Table 6 below. The model is unable to accurately predict the activities of *Throw*, *Basketball Shoot*, *Draw Circle (CW)*, *Tennis Serve*, and *Push* as the motions may be similar to each other and hard to distinguish with a single inertia sensor.

TABLE 6
PRECISION AND RECALL (INERTIA MODEL)

Activity	Precision	Recall	F1-score	Support
Swipe Left	1	1	1	8
Swipe Right	1	0.82	0.9	11
Wave	0.92	1	0.96	12
Clap	1	1	1	12
Throw	0.75	0.5	0.6	6
Arm Cross	1	1	1	8
Basketball Shoot	0.73	0.89	0.8	9
Draw X	1	1	1	12
Draw Circle (CW)	0.73	1	0.84	8
Draw Circle (CCW)	1	0.9	0.95	10
Draw Triangle	1	0.88	0.93	8
Bowling	1	1	1	9
Boxing	1	1	1	8
Baseball Swing	1	0.91	0.95	11
Tennis Swing	1	1	1	5
Arm Curl	1	1	1	13
Tennis Serve	0.8	0.89	0.84	9
Push	0.79	1	0.88	11
Knock	1	0.93	0.97	15
Catch	1	0.8	0.89	10
Pickup and Throw	1	1	1	7
Jog	1	1	1	9
Walk	1	1	1	13
Sit to Stand	1	1	1	7
Stand to Sit	1	1	1	8
Lunge	1	1	1	8
Squat	1	1	1	13

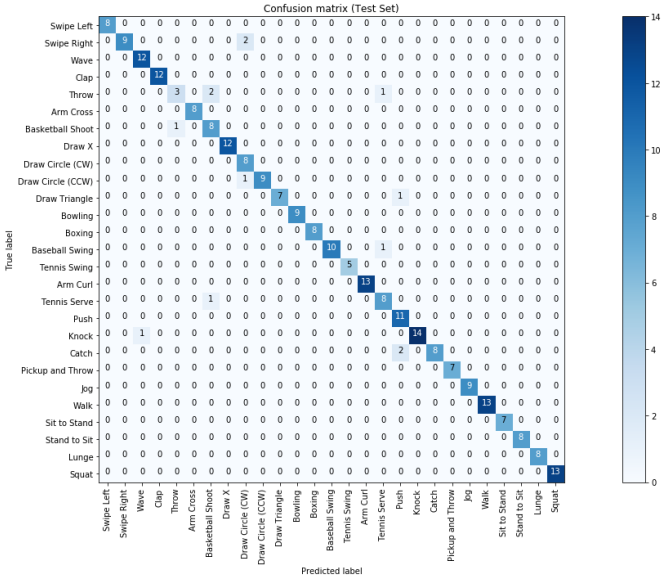

Figure G: Confusion Matrix (Inertia Model)

B. Depth

XGboost model has achieved an accuracy of 81.9% in predicting the activity of the subject in the test set. The confusion matrix as reflected in Figure H indicates that the model performed satisfactory in predicting the activity based on DDMs, achieving 100% recall rate for Arm curl, *Draw circle*, *Boxing*, *Baseball swing*, *Tennis serve*, *Pickup and throw*, *Jog*, *Sit to stand* and S*tand to sit*. Due to minor change in depth, the depth model is unable to accurately predict activities such as *Push* (45%) and C*lap* (58%).

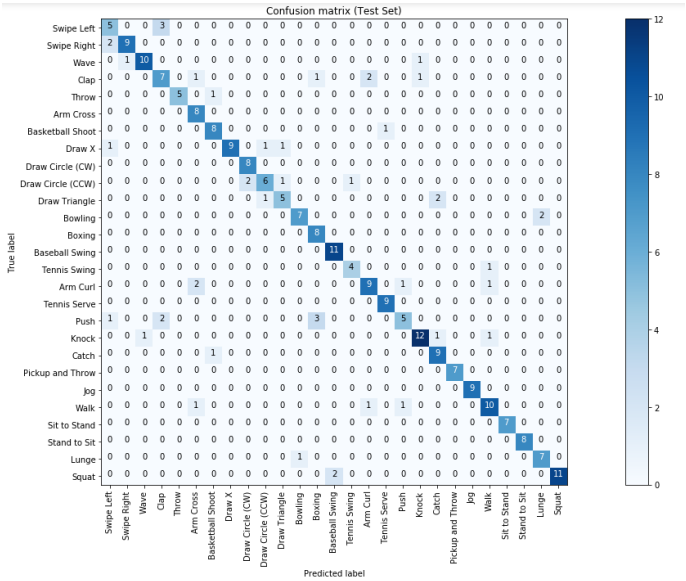

Figure H: Confusion Matrix (Depth Model)

TABLE 7
PRECISION AND RECALL (DEPTH MODEL)

Activity	Precision	Recall	F1-score	Support
Swipe Left	0.56	0.62	0.59	8
Swipe Right	0.90	0.82	0.86	11
Wave	0.91	0.83	0.87	12
Clap	0.58	0.58	0.58	12
Throw	1	0.83	0.91	6
Arm Cross	0.67	1	0.8	8
Basketball Shoot	0.80	0.89	0.84	9
Draw X	1	0.75	0.86	12
Draw Circle (CW)	0.80	1	0.89	8
Draw Circle (CCW)	0.75	0.62	0.67	10
Draw Triangle	0.71	0.62	0.67	8
Bowling	0.88	0.78	0.82	9
Boxing	0.67	1	0.80	8
Baseball Swing	0.85	1	0.92	11
Tennis Swing	0.80	0.80	0.80	5
Arm Curl	0.75	0.69	0.72	13
Tennis Serve	0.90	1	0.95	9
Push	0.71	0.45	0.56	11
Knock	0.86	0.80	0.83	15
Catch	0.75	0.90	0.82	10
Pickup and Throw	1	1	1	7
Jog	1	1	1	9
Walk	0.77	0.77	0.77	13
Sit to Stand	1	1	1	7
Stand to Sit	1	1	1	8
Lunge	0.78	0.88	0.82	8
Squat	1	0.85	0.92	13

TABLE 9
PRECISION AND RECALL (SKELETON MODEL)

Activity	Precision	Recall	F1-score	Support
Swipe Left	1	1	1	8
Swipe Right	1	0.91	0.95	11
Wave	1	0.92	0.96	12
Clap	1	0.92	0.96	12
Throw	1	1	1	6
Arm Cross	1	1	1	8
Basketball Shoot	1	1	1	9
Draw X	1	1	1	12
Draw Circle (CW)	0.89	1	0.94	8
Draw Circle (CCW)	0.83	1	0.91	10
Draw Triangle	1	0.62	0.77	8
Bowling	1	1	1	9
Boxing	1	1	1	8
Baseball Swing	1	1	1	11
Tennis Swing	1	1	1	5
Arm Curl	0.93	1	0.96	13
Tennis Serve	1	1	1	9
Push	1	1	1	11
Knock	0.88	1	0.94	15
Catch	1	1	1	10
Pickup and Throw	1	1	1	7
Jog	0.9	1	0.95	9
Walk	1	0.92	0.96	13
Sit to Stand	1	1	1	7
Stand to Sit	1	1	1	8
Lunge	1	1	1	8
Squat	1	1	1	13

C. Skeleton

The results are shown in Table 8 below. It is clear that feature engineering (transforming raw coordinates into displacement vectors) resulted in better classification accuracy by allowing for better generalization which is location-invariant and size-invariant. Amongst the models built using the engineered features, the LSTM model performed the best based on a test set accuracy of 97.3%. The confusion matrix of the best LSTM model built using engineered features is shown in Figure I.

TABLE 8
PERFORMANCE OF SKELETON MODELS

Model	Raw Coordinates		Feature Engineering	
	Accuracy (Training)	Accuracy (Test)	Accuracy (Training)	Accuracy (Test)
LSTM	90.1%	79.2%	99.5%	97.3%
GRU	89.4%	82.7%	100%	95.8%
TCN	100%	89.6%	100%	96.9%

As seen from the confusion matrix, the model is able to perform satisfactorily using only the skeleton features in the dataset. Table 9 presents the precision and recall rates of each of the 27 activities. We can see that the model has slight difficulties identifying drawing circle (clockwise and counter clockwise) and drawing triangle. It has also wrongly classified walking as jogging in one test instance, possibly due to close resemblance of both activities.

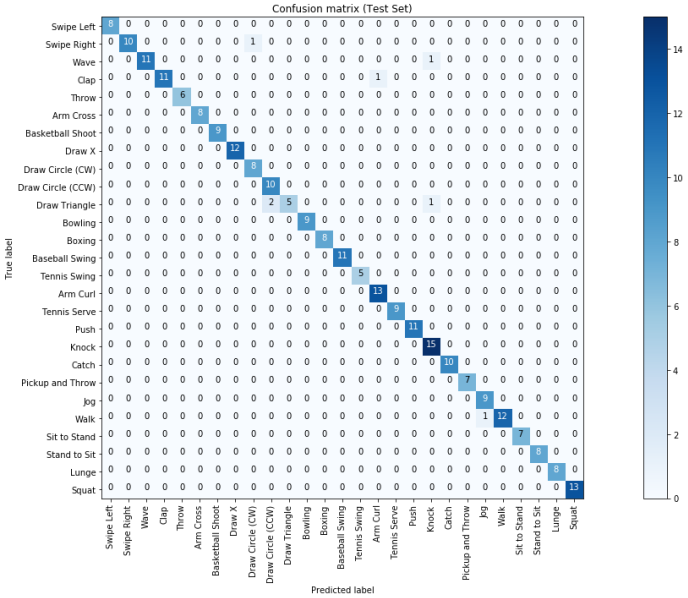

Figure I: Confusion Matrix (Skeleton Model)

D. Ensemble

Several ensemble models were built using selected models built using each of the 3 modalities. Both soft voting and stacking were performed with all 3 models, and just the top

2 models with the best accuracies. The top 2 models are: the random forest model built using inertial information (95.0%) and the LSTM model built using transformed displacement vectors from skeleton information (97.3%).

The performance of the ensemble models is tabulated in Table 10. As seen in the table, the highest accuracy on the test data was 94.6% using either soft voting or stacking involve the top 2 models (inertia and skeleton model). However, this is worse off than using solely skeleton data for prediction.

TABLE 10
PERFORMANCE OF ENSEMBLE MODELS

Model	Accuracy (Train)	Accuracy (Test)
Soft Voting (All 3)	100%	92.3%
Soft Voting (Top 2)	100%	94.6%
Stacking (All 3)	100%	91.2%
Stacking (Top 2)	100%	94.6%

VI. DISCUSSION

Our proposed framework of feature extraction, model learning on individual sensor modalities and finally ensembling the predictions from each modality may be applied in other application areas involving multi-modal sensor data.

For the UTD-MHAD dataset, we have achieved high accuracy in activity recognition using sensor data, with feature engineering techniques. The results suggest that using skeleton data alone is sufficient to achieve a high degree of accuracy in activity classification, using a LSTM model with feature engineering.

There are a few limitations to the study. Firstly, the dataset was randomly partitioned into a training/test set and as a result, the same subjects can be found in both the training and test set. Future work may try to validate model on subjects who are unseen by the training set, by splitting the dataset based on subject IDs. This would allow for a fairer evaluation of model performance on unseen subjects. Secondly, a fixed set of hyperparameters was used for model training. Hyperparameter tuning with the use of grid search and cross validation may be used to fine-tune the models. Finally, due to the limited dataset on hand, data augmentation techniques may also be used in subsequent works to generate "more data". For example, skeleton joint coordinates may be rotated in random amounts around the vertical axis, or displaced from the original coordinates by small random amounts. Given that deep learning techniques tend to be large amounts of parameters, it is important to have a proportionate amount of training examples to have good performance. This also would help as the original dataset was generated using highly controlled conditions whereas the actual application may consist of much more variations in the subject movement patterns, positions, orientations, brightness, etc.

VII. CONCLUSION

In conclusion, an ensemble approach has been proposed in this paper to perform activity classification using multimodal sensor data, i.e. depth, skeleton and inertia. Feature engineering techniques such as the use of Fast Fourier Transform, Depth Motion Maps and 3D coordinate transformations have been used to improve classification performance. The best model from each modality was then included in an ensemble model in order to assess if further improvements in classification accuracy could be achieved. The results show that the use of skeleton data alone is sufficient to achieve a high level of accuracy in this study.

REFERENCES

[1] Kim, T. S., and Reiter, A. 2017. Interpretable 3d human action analysis with temporal convolutional networks. In BNMW CVPRW

[2] O. Kramer, Computational Intelligence - Eine Einführung. Berlin, Heidelberg: Springer, 2009.

[3] R. Lun and W. Zhao, "A survey of applications and human motion recognition with microsoft kinect," International Journal of Pattern Recognition and Artificial Intelligence, vol. 29, no. 05, p. 1555008, 2015.

[4] M. P. V. Kellokumpu and J. Heikkila, "Human activity recognition using sequences of postures," in Proc. IAPR Conf. Mach. Vision Appl., 2005, pp. 570–573

[5] X Chen, M Koskela. Skeleton-based action recognition with extreme learning machines, Neurocomputing, 2015(149): 387-396.

[6] Agahian, Saeid & Negin, Farhood & Köse, Cemal. (2018). Improving bag-of-poses with semi-temporal pose descriptors for skeleton-based action recognition. The Visual Computer. 10.1007/s00371-018-1489-7.

[7] C. Li, Y. Hou, P. Wang and W. Li, "Joint Distance Maps Based Action Recognition With Convolutional Neural Networks," in IEEE Signal Processing Letters, vol. 24, no. 5, pp. 624-628, May 2017. doi: 10.1109/LSP.2017.2678539

[8] S Liu, H Wang. Action Recognition Using Key-frame Features of Depth Sequence and ELM, International journal of advanced computer science and applications. 2017,8(10): 52-56.

[9] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1110–1118.

[10] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 4041–4049.

[11] W. Zhu et al., "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in Proc. AAAI Conf. Artif. Intell., 2016, pp. 3697–3704

[12] C. Núñez, Juan & Cabido, Raúl & Pantrigo, Juan & S. Montemayor, Antonio & F. Vélez, José. (2018). Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition. 76. 80-94. 10.1016/j.patcog.2017.10.033.

[13] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[14] Damla Arifoglu, Abdelhamid Bouchachia, Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks, Procedia Computer Science, Volume 110, 2017, Pages 86-93, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.06.121.

[15] Kyunghyun Cho, Bart van Merienboer, Caglar Gulcehre, Dzmitry Bah- ¨ danau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Ed., Doha, Qatar, 2016, pp. 1724–1734.

[16] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271, 2018

[17] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. and Blake, A., 2011, June. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 1297-1304). Ieee.

[18] Xia, L., Chen, C.C. and Aggarwal, J.K., 2012, June. View invariant human action recognition using histograms of 3d joints. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (pp. 20-27). IEEE.

[19] Li, W., Zhang, Z. and Liu, Z., 2010, June. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (pp. 9-14). IEEE.

[20] Chen, C., Liu, K. and Kehtarnavaz, N., 2016. Real-time human action recognition based on depth motion maps. *Journal of real-time image* processing, 12(1), pp.155-163.

[21] Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[22] UTD Multi-view Action Dataset. Retrieved from: http://www.utdallas.edu/~kehtar/MultiViewDataset.pdf

[23] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.