Significance of handcrafted features in human activity recognition with attention-based RNN models

Original Scientific Paper

Sonia Abraham
Department of Computer Science, Cochin University of Science and Technology,
Kochi, India
sonianidhi@gmail.com

Rekha K James
Division of Electronics, School of Engineering, Cochin University of Science and Technology,
Kochi, India.
rekhajames@cusat.ac.in

Abstract *– Sensors incorporated in devices are a source of temporal data that can be interpreted to learn the context of a user. The smartphone accelerometer sensor generates data streams that form distinct patterns in response to user activities. The human context can be predicted using deep learning models built from raw sensor data or features retrieved from raw data. This study analyzes data streams from the UCI-HAR public dataset for activity recognition to determine 31 handcrafted features in the temporal and frequency domain. Various stacked and combination RNN models, trained with attention mechanisms, are designed to work with computed features. Attention gave the models a good fit. When trained with all features, the two-stacked GRU model performed best with 99% accuracy. Selecting the most promising features helps reduce training time without compromising accuracy. The ranking supplied by the permutation feature importance measure and Shapley values are utilized to identify the best features from the highly correlated features. Models trained using optimal features, as determined by the importance measures, had a 96% accuracy rate. Misclassification in attention-based classifiers occurs in the prediction of dynamic activities, such as walking upstairs and walking downstairs, and in sedentary activities, such as sitting and standing, due to the similar range of each activity's axis values. Our research emphasizes the design of streamlined neural network architectures, characterized by fewer layers and a reduced number of neurons when compared to existing models in the field, to design lightweight models to be implemented in resource-constraint gadgets.*

Keywords*: Attention mechanism, deep learning, Gated Recurrent Units*

1. INTRODUCTION

Human activity recognition (HAR) is a widely researched area due to the availability of cutting-edge technologies and miniature devices incorporating various sensors capable of transmitting time-series data forming patterns that help users monitor motion. Of the available sensors, tri-axial accelerometer and gyroscope sensors are commonly used in determining human context. Various activities ranging from simple motion to highly complex activities can be accurately determined from the sensor data.

Studies carried out a decade ago utilized the potential of machine learning classifiers. With the advent of deep learning models, features are automatically extracted from raw sensor data for classifying the activities. The more the number of layers and hidden neurons, the more time for training. Another pitfall is that the data collected from sensors forms a spare matrix as the continuous signal flow is disturbed due to the subtle movement of the sensor, resulting in less well-labeled data and an imbalanced class [1]. Data augmentation techniques help in the formation of synthetic data computed from the available annotated data blocks, which reduces overfitting in models with fewer well-labeled data. Of the experimented data augmentation strategies, the moving average smoothing and exponential smoothing, the latter increased the accuracy of the RNN models considerably when applied to temporal data.

The recurrent neural network variants, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), exhibit high-performance accuracy for time-series classification problems [2]-[4]. When subtle movements

are to be considered, deep learning networks fail to capture the context information required for generalization from a domain with an unbalanced class distribution, in particular. Simplified and context-based attention is applied to hidden layers in a two-stacked GRU architecture concatenated and applied to fully connected layers to learn weights for features automatically computed from attention. When the layers of these networks are enhanced with an attention mechanism, the accuracy improves with the benefit of reduced model parameters [5].

Later studies extracted temporal and frequency dimension features from time-series data and proved that the deep learning models trained with these features give comparable accuracies. A total of 202 temporal and frequency domain features extracted from raw accelerometer and gyroscope data stream with selected base classifiers recognized human motion in a context-aware scenario employing an incremental learning model [6]. An ensemble model updated through weighted majority voting when run with user and position-independent streaming data generates personalized context recognition applications.

To understand the decision taken by a model in classification, the knowledge of features is required. To learn the impact of each feature on the performance of a supervised learning model, a technique called feature importance ranking is used in deep learning models. The existence of irrelevant features in the input may breach the generalization of a learning system. An optimal subset of features results in better model performance with optimum resources. In this regard, binary and multi-class classification is conducted using features selected based on their correlation diversity from different activities [7]. Pair-wise correlation between temporal features extracted from the raw accelerometer and gyroscope sensors of each activity is determined and extended to compare binary subsets. Classification using top-ranked features gave comparable performance against all with much reduction in data space. The study in [8] correctly separates dynamic activities using bidirectional LSTM architecture, deploying a grid search strategy for the selection of layers and their depth.

This work uses LSTM and GRU models to classify human context using features computed from raw accelerometer sensor data available in UCI-HAR, a public dataset with 6 activities. The contributions of the study include the following:

- The construction of single and two stacked recurrent neural network variants with attention added to the RNN layer before the fully connected layer. The input to the models is 31 temporal and frequency domain statistical features extracted from raw accelerometer sensor data.
- The work delves into feature selection, a critical task to optimize model performance and efficiency.
- The choice of optimal features from the handcrafted features computed from accelerometer sensor data, employing feature importance ranking measures.
- The comparison of the accuracy of models trained with all features and optimal features. The evaluation shows that training the attention-based classifiers with significant features improved model performance.
- GRU models' performance is better than LSTM models considering the accuracy metric.

Following is the structure of the remaining paper: Section 2 describes the related literature in time-series classification using deep learning models. Section 3 gives a detailed explanation of various architecture, attention mechanisms and feature importance measures. Section 4 gives an insight into the proposed method, while experiments and results are discussed in Section 5.

2. RELATED WORKS

An extensive study on various deep learning architectures apropos speed, accuracy and memory is conducted in [9] using accelerometer and gyroscope sensor data to determine human context. The models employed hidden units in the range of 100 to 600 that span across three layers. The article concluded that CNN captures sensor correlations and temporal dependencies efficiently. The model proposed in [10] used two stacked LSTM with 64 neurons connected to two convolution layers with 64 and 128 filters and a global average pooling layer which reduced model parameters to a great extent without compromising recognition rates. The performance effect of filter count, optimizers and batch size are investigated and the final model is trained utilizing the selected optimal hyper-parameters.

According to a multi-input CNN-GRU model introduced in [11], the same input is fed into three heads with convolution layers having various filter sizes, allowing the model to collect feature vectors with local correlations at various scales. Two GRU layers that automatically extract features from the input dataset and categorize activity data are applied after the convolution layers. To distinguish between everyday activities and accidents involving falls, this study [4] used independent GRU models with forward and backward cells that each had 200 hidden units of data gathered from a mobile phone accelerometer sensor.

A hierarchical framework constructed with an SVM classifier on 12 statistical features from time and frequency domain differentiated coarse-level activities as a GRU network input with RSSI data discriminated fine-level similar activities [12]. The RSSI data was acquired from sensors mounted at different locations in two different environments utilizing the sniffer technique, thereby creating a low-cost, device-free HAR system. Accelerometer and angular velocity signal values from

the Shimmer platform are gathered to build low-energy wearable devices for fall and activity recognition [13]. These values are then subjected to a compressed sensing method. Machine learning classifiers are used to extract and analyze 44 temporal characteristics. According to the study, the use of compressed sensing algorithms led to an increase in battery life of 2.55 times the higher duration without sacrificing precision.

Fall and non-fall events are categorized using four recurrent networks, and their performance is analyzed using an embedded device [14]. The advantage of these wearable fall detectors apropos energy consumption is provided in the paper. The findings demonstrate that when implemented in a real-time microcontroller that runs on small batteries, a single RNN is capable of differentiating multi-class falls.

The model proposed in [15] compared a linear and a nonlinear dimension reduction and feature selection technique using LSTM in binary and multiclass classification for monitoring malicious activity in a network. The calculation of the mutual information score helped select those features from among the 53 available features that had the most significant impact in training the model. A novel dimensionality reduction technique that randomly organizes a small number of feature vectors from each class is proposed in [16]. The selection of features is determined based on the Euclidean distance of feature vectors that fall in a specific range. The proposed method chose 11% of features, and the classification techniques resulted in good accuracy and low response time, paving the way to low computational cost.

The LSTM architecture's inception is covered in detail [17], along with a potential fix for the vanishing gradient issue that involves maintaining a constant error flow by controlling internal cell states. The potential of LSTM in overcoming the vanishing gradient problem substantiated by its performance in a handful of domains is portrayed [18]. The review covers in detail the main components of the network, their interaction and the determination of the weight matrix using a gradient-based method. When the input is sparse dataset, a data-driven model like LSTM is utilized to simulate flood flow by analyzing sequential data streams and identifying long-term dependencies [19]. The study in [20] examined whether memristor-based LSTM might be used as a solution to the low speed of these models caused by their parallel structure and sequential behavior.

GRU networks with weighted averaging, which give more weights to the middle range, determine representations using handcrafted features extracted from raw input vectors, thereby minimizing the need for expert-level knowledge in feature design [21]. A real-time model that recognizes complex activities is designed [2] by concatenating multiple convolution kernels with different scaling and max-pooling layers. Four such Inception-like network is connected with two GRU layers, enabling the model to extract sequential temporal dependencies. The classification performance of gated recurrent units in emotion recognition is examined [22] using clean speech data overlay with various environmental noises. According to the research, GRU produced results similar to LSTM while taking 18.6% less time to run. Network intrusions can be effectively identified using a single or bi-directional GRU with 128 hidden units followed by three layers of multilayer perceptron with 48 hidden nodes and softmax regression. According to the study [3] BGRU can do better in related domains. The work in [5] suggests that augmenting convolution models with LSTM moderately improves the performance of time-series datasets. The refinement phase that iterates with change in learning rate and batch size improved model accuracy although with increased computational complexity.

This study applies different RNN variants with attention to time and frequency domain features that are extracted from raw accelerometer sensor data to identify human activity. These features are ranked according to relevance, and the best features are used to train the model.

3. BACKGROUND

Recurrent neural networks (RNN) set themselves apart from other types of neural networks by having a memory structure that uses earlier input data to influence current input and output and finds use in ordinal or temporal problems involving sequential data. The units in each layer share the same weight parameters and leverage the backpropagation algorithm to determine the gradients that appropriately fit the model parameters. This process may result in smaller gradients, which ceases the network's ability to learn by generating insignificant weight parameters through constant updates. Long short-term memory (LSTM) and Gated recurrent units (GRU) mitigate the short-term memory problem of RNN models. Internal gating mechanisms in these variants control information flow, eliminate unnecessary input, and retain what is necessary for accurate prediction.

3.1. LONG SHORT-TERM MEMORY (LSTM)

Long short-term memory (LSTM) networks are well suited for more general sequence learning tasks like activity identification because they can learn from sequences of data through recurrent processing, where the input at the present step influences the output at subsequent time steps. The gates in LSTM regulate the control flow by learning throughout the training procedure what is significant and what can be allowed on the cell state [23]. To learn which data should be retained and which ought to be removed based on relevance, gates integrate sigmoid activations that push values between 0 and 1 rendering them suited for backpropagation [24].

The output of the LSTM is decided by the current long-term memory of the network, the cell state, the prior hidden state, and the input data at the current time step. The forget gate, f_t, determines which part of

the long-term memory should be deleted or retained at this time, given the prior hidden state and the current data point in the sequence.

$$f_t = \sigma\left(W_i[h_{t-1}, x_t] + b_f\right) \quad (1)$$

Given the previous hidden state and the new input data, the new memory network and input gate, i_t, decide what new information should be added to the network's long-term memory. The tanh-activated neural network, which squishes values between -1 and 1, creates a vector that indicates how much to update each component of the cell state given the new data.

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \quad (2)$$

$$\check{c} = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \quad (3)$$

The output from the input gate is used to update the cell state, c_t, to new values that the neural network deems appropriate.

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \quad (4)$$

The output gate, o_t, which is applied to the newly updated cell state, is a filter that accepts the same input as the forget gate along with a sigmoid activation to ensure that only necessary information is output. Hence, the output gate determines the next hidden state, which aids in prediction.

$$o_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_o\right) \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

where W_x, gate(x) neuron weights, h_{t-1}, the output of prior block, x_t, input at current time step, b_x, gate(x) biases, c_t, cell state at time step(t) and $\check{c}$, candidate for cell state at time step (t). The following time step is subsequently updated with the new hidden state and cell states.

3.2. GATED RECURRENT UNITS (GRU)

A gated recurrent unit (GRU) is designed similarly to LSTM and, for the most part, yields results that are just as good. Two vectors - update and reset gates – decide information transfer and the model trains relatively quickly with fewer parameters.

The update gate, z_t, assists the model in determining how much of the past information should be transmitted into the future, thereby eliminating the possibility of the vanishing gradient problem.

$$z_t = \sigma\left(W^{(z)} x_t + U^{(z)} h_{t-1}\right) \quad (7)$$

The model uses the reset gate, r_t, to decide how much of the past information to forget.

$$r_t = \sigma\left(W^{(r)} x_t + U^{(r)} h_{t-1}\right) \quad (8)$$

The candidate activation, $\hat{h}_t$, which uses the reset gate to store the apposite information from the past is computed.

$$\hat{\mathrm{h}}_t = \tanh(W x_t + r_t \odot U h_{t-1}) \quad (9)$$

Finally, the activation at time t, which contains the information for the current unit is determined and transmitted to the network.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{\mathrm{h}}_t \quad (10)$$

where h_{t-1}, the output of the prior vector, x_t, the input vector, h_t, the output vector, W and U, weight matrices and $\odot$, the Hadamard product.

3.3. ATTENTION MECHANISM FOR ACTIVITY RECOGNITION

The attention mechanism enhances the accuracy of time series classification models by emphasizing the more pertinent temporal features produced by recurrent neural networks through a weighted combination of all hidden state vectors to focus the models's attention on the most significant part of the input sequence. The hierarchical context-based attention mechanism employs an adaptive focusing technique to produce a context vector capable of utilizing a hierarchy of time-dependent features in the sensor data.

Alignment scores, weights, and the context vector are computed iteratively as part of the attention process. The alignment score, e_{ij}, computed using hidden state, h_j, and previous output, s_{i-1}, indicates how closely the parts of the input sequence match with the current output.

$$e_{ij} = a\left(s_{i-1}, h_j\right) \quad (11)$$

where a(.) represents the alignment model of a feedforward neural network.

A softmax function is applied to the previously computed alignment scores to determine the weights.

$$\alpha_{ij} = \frac{e^{(e_{ij})}}{\sum_{k=1}^{T_x} e^{(e_{ik})}} \quad (12)$$

The weighted sum of the hidden states then decides the context vector, c.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (13)$$

Incorporating attention to the layer before the dense layer in a single RNN and on layers of a stacked network improves accuracy, reduces the computational cost and helps to learn the decision process.

3.4. FEATURE IMPORTANCE

3.4.1. CORRELATION AMONG FEATURES:

Algorithms for predictive modeling need data adequately representational of the target domain. Identifying and extracting discriminating features helps construct a model that not only performs effectively but also lessen the likelihood that decisions will be based on outliers. The selection of appropriate features can be determined statistically using correlation analysis techniques like Pearson product-moment correlation and Spearman rank-order correlation.

Pearson's correlation coefficient concisely summarises the relationship between two data points. It is calculated by dividing the covariance of the two points by the product of their respective standard deviation. For a given feature variable, x, and target variable, y, the Pearson coefficient, ρ_{xy}, is defined as:

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x-\bar{x})(y-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y-\bar{y})^2}} \quad (14)$$

where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$

Spearman's correlation coefficient is a good measure when the nature of the distribution and relationship between the data points remain unclear. For the given variables, the Spearman correlation, σ_{xy}, is calculated as:

$$\sigma_{xy} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad \textbf{(15)}$$

where d_i is the difference between the ranks of the data points. Fast Fourier Transform's first and third components are most correlated from among the 31 features calculated on the UCI-HAR dataset, with correlation values of -0.803 and 0.622 respectively, Fig. 1.

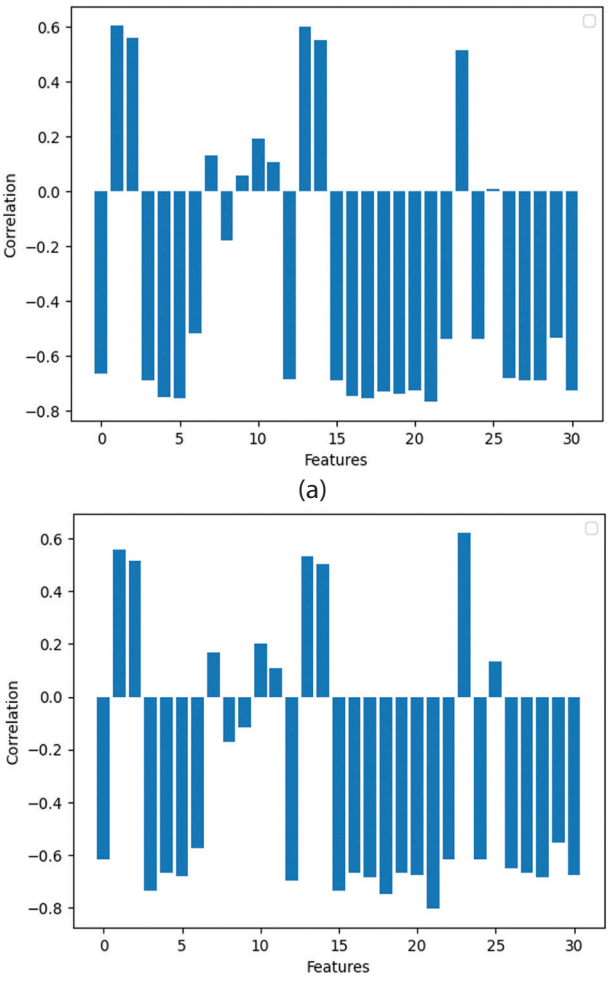


Fig. 1. Correlation between feature and target data points in UCI-HAR dataset (a) Pearson Correlation (b) Spearman's rank

3.4.2. Permutation Feature Importance:

Permutation feature importance inspects the estimator by randomly shuffling one feature and the feature's importance is determined by calculating the model's prediction error. Suppose the model error increased after permuting the feature, in that case, it indicates that the feature contributed to prediction and if the error did not increase, it implies that the feature is not significant.

Compute the score, $s_{n,j}$, of a fitted predictive model, m, on a feature matrix, F, using a scoring argument that accepts multiple scores (like RMSE) for each iteration n from 1 to N and for each feature, f_j, permuted randomly.

The computation of importance, i_j, for a given feature, f_j, is defined as:

$$i_j = s - \frac{1}{N}\sum_{n=1}^{N} s_{n,j} \quad (16)$$

Feature importance gives a reasonable interpretation of how the model will behave, provided the original prediction is accurate. It requires that features be uncorrelated since highly correlated features reduce the importance of the feature in question by spreading the importance between both features.

3.4.3 Shapley Values

The contribution of a feature in a prediction is given by the Shapley value, which evaluates the relevance of a feature by including and excluding it in the prediction. More precisely, understanding how the model behaves for every potential combination of features is necessary to calculate Shapley values. The Shapley value, $\emptyset_i(v)$, for a given feature is calculated as:

$$\emptyset_i(v) = \frac{1}{|N|!}\sum_R [v(P_i^R \cup \{i\}) - v(P_i^R)] \quad (17)$$

where R, is the permutation order of features, $v(P \cup \{i\})$, is the contribution of features to the outcome including the ith feature and $v(P)$, is the contribution of features on the outcome excluding the ith feature. This calculates each feature's average contribution by adding the marginal contribution of an individual feature to the result of all possible permutations of the order of the features.

4. PROPOSED METHOD

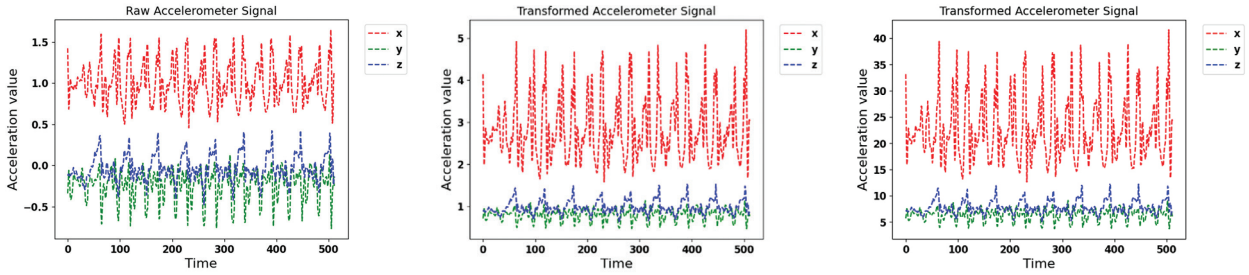
The study predicts human context using data acquired from smartphone accelerometer sensors. The raw signals are analyzed to extract several temporal and frequency domain features, which are then applied to deep learning models with an attention mechanism added to the layers before the dense layer. The model interpretability is studied using the permutation feature importance and shapely values.

4.1 DATASET DESCRIPTION

The public dataset, Smartphone-based Recognition of Human Activities and Postural Transitions Data Set (UCI-HAR https://archive.ics.uci.edu/dataset/240) [25], is used to evaluate the recurrent neural network variants.

Tri-axial accelerometer and gyroscope data from 30 users between the ages of 19 and 48 were collected at a constant rate of 50 Hz for the samples in a semi-controlled environment. The signal data is available for 6 activities, three sedentary activities (*Sitting, Standing, and Lying*), three dynamic activities (*Walking, Walk Upstairs and Walk Downstairs*) and postural transitions.

The dataset contains an equal proportion of representative samples from all activities.



(a) (b) (c)

Fig. 2. Accelerometer Signal Transformation (a) Raw signal (b) e^x of axis values (c) Fibonacci (e^x) for n=4

4.2. DATA PREPROCESSING

The accelerometer sensor data is considered a three-dimensional vector, $A_i=(a_{x_i}, a_{y_i}, a_{z_i}) \in R^3$, where A_i represents the ith signal data. The raw signal data is smoothened using a simple exponential window function.

$$A_k = \alpha A_k + (1-\alpha) A_{k-1} \text{ where } \alpha=0.001$$

The continuous stream data is divided into blocks of 128 sample points, called an example. The raw values are transformed to a larger range by first computing the exponent of each axis value, $y=e^x$, and then by generating the Fibonacci series with the initial value obtained from the earlier step for $n = 4$. The transformations are applied in succession to separate linearly inseparable values; see Fig. 2. 31 temporal and frequency domain features are computed from a 50% overlapped window. The complete feature list is given in Table 1. The standard deviation indicates how far the signal deviates from its mean value. Skewness quantifies the dispersion of a signal around the mean value and is computed as the ratio of average deviation from the mean cubed by the standard deviation cubed. The availability of peaks in a normal distribution is measured by kurtosis. It is the fourth central moment divided by the square of the variance.

Table.1. Temporal and frequency domain feature list

No	Feature
1 – 3	Mean along each axes
4 – 6	Standard Deviation along each axes
7 – 9	Skewness of the component signal
10 – 12	Kurtosis of the component signal
13 – 15	Root Mean Square of ith acceleration vector
16 - 18	Mean Absolute Deviation
19 – 21	Range of each axes
22 – 25	Fast Fourier Transformation components of block
26	Mean of magnitude vector
27	Standard deviation of magnitude vector
28	RMS of Standard Deviation along each axes
29	Standard Deviation magnitude in the horizontal plane
30	RMS of axes data in the horizontal plane
31	Maximum peak-to-peak acceleration amplitude

Root Mean Square (RMS) determines the signal amplitude and energy in the time domain.

$$RMS(\vec{a}[k]) = \sqrt{a_x^2[k] + a_y^2[k] + a_z^2[k]} \quad (18)$$

The mean absolute deviation is the average distance between each data point, a_{k_i}, and the mean, μ.

$$MAD = \sum \frac{(a_{ki}-\mu)}{N}; k \in (x, y, z) \quad (19)$$

The range is calculated as the difference between maximum and minimum axes value. Fast Fourier Transform is applied to blocks to determine the amplitude spectrum. The standard deviation of the magnitude vector is determined as

$$M(\vec{a}[k]) = \sqrt{\sigma_x^2[k] + \sigma_y^2[k] + \sigma_z^2[k]} \quad (20)$$
$$\text{with } \sigma_i = std(\vec{a}[k])$$

The standard deviation of the horizontal plane and RMS of axes data in the horizontal plane is computed with data points in the x, a_{k_x}, and z, a_{k_z}, axes.

$$SD_{hor}(\vec{a}[k]) = \sqrt{\sigma_x^2[k] + \sigma_z^2[k]} \quad (21)$$

$$RMS_{hor}(\vec{a}[k]) = \sqrt{a_x^2[k] + a_z^2[k]} \quad (22)$$

Maximum peak-to-peak acceleration amplitude gives the maximum positive or negative signal deviation from its reference level.

$$P_k - P_k = RMS\,(\max(\vec{a}[k]) - \min(\vec{a}[k])) \quad (23)$$

4.3. LSTM AND GRU NETWORKS:

The study uses single and two-layer stacked LSTM and GRU architectures as well as LSTM-GRU combination networks with attention applied to the hidden state output of the layer immediately preceding the dense layer. The proposed model is depicted in Fig. 3.

Single-stacked models used 64 hidden neurons in the LSTM or GRU layer and a fully connected dense layer with 32 neurons. A dropout of 50% is applied after the first layer to aid with the de-correlation of weights

by selectively eliminating neurons, thus forming a better representation of input data.

The rectified linear activation unit (ReLU) that works on a simple calculation that drops the value to 0 for non-positive inputs is employed as the activation function in layers. The final fully connected output layer is applied with a softmax activation to classify human activities. The two stacked variants and the combination models used 64 neurons each in their architecture layers, followed by a dense layer with 32 neurons. The stacked variant model is given in Fig. 4.

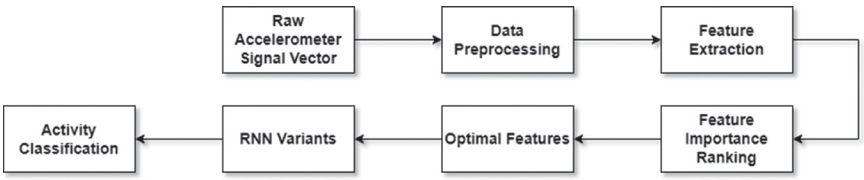


Fig. 3. Proposed Architecture

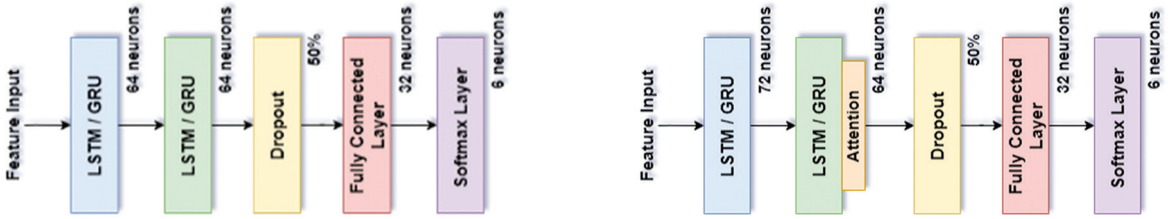


Fig. 4. Two stacked layers model architecture (a) without attention (b) with attention

For LSTM and GRU stacked models, a dropout of 50% is introduced after the stacked layers. However, for the combination model, a dropout of 50% is added to the first layer and 20% to the second layer. All layers except the final fully connected layer used the ReLU activation function. The architecture diagram of the LSTM-GRU combination model is given in Fig. 5.

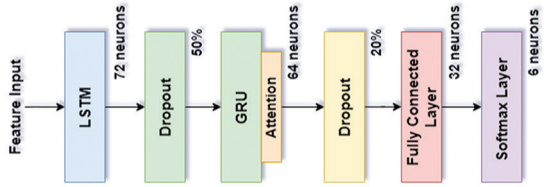


Fig. 5. LSTM-GRU Combination model architecture

In every architecture, an attention mechanism is added to the layer just preceding the dense layer. When computing the weights for single stacked models, the attention mechanism uses the only RNN layer available. For stacked and combination models, weights are computed with the second RNN layer preceding the fully connected layer. All models gave better fit when trained with Adam optimizer and a batch size of 32. The models converged at epochs in the range of 25 to 40 when the early stopping regularization technique was implemented. Validation loss is the monitoring metric to terminate the training based on validation performance. The learning rate is the hyperparameter that determines how much of the model should be changed each time the weights are updated with the estimated error. Single-stacked models performed better by setting the learning rate to 0.0025, while stacked models worked with a value of 0.001.

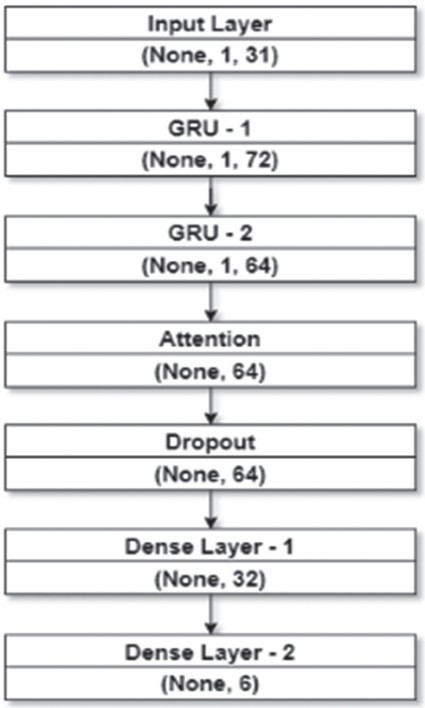


Fig. 6. Network architecture diagram of the two stacked GRU model

The input sequence to the network in Fig. 6 is the features vector computed from each example of the transformed accelerometer signal. The GRU layers process the sequential data and capture temporal dependencies. The attention mechanism added to the second GRU layer focuses on specific elements of the sequence. The dense layers process the output from the GRU layers and classify the activities.

5. EXPERIMENTS AND RESULTS

The proposed models in the activity recognition domain are evaluated using the UCI-HAR public dataset. The metrics used in evaluating the model classification performance are accuracy, the ratio of correct predictions to total predictions; precision, a measure of the accuracy of positive predictions; recall, a measure of the completeness of positive predictions and F1-score, the harmonic mean of precision and recall.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (24)$$

$$Precision = \frac{TP}{TP+FP} \quad (25)$$

$$Recall = \frac{TP}{TP+FN} \quad (26)$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (27)$$

5.1. EVALUATION OF MODELS USING ALL COMPUTED FEATURES

The two stacked GRU architectures exhibited the highest accuracy of 99% when the model is trained with attention, compared to 98% when the attention mechanism is not used. Table 2 compares the recognition rate for models with and without attention when trained with all 31 features extracted from the raw accelerometer signal.

Table. 2. Recognition rate of classifiers in the UCI-HAR dataset

Model Architecture	Performance (Accuracy)	
	Model without Attention	Model with attention
Single Layer LSTM	0.85	0.89
Single Layer GRU	0.89	0.87
Two Stacked LSTM	0.94	0.93
Two Stacked GRU	0.98	0.99
LSTM - GRU	0.96	0.95
GRU - LSTM	0.93	0.93

The performance of stacked GRU architecture, when trained with attention, is given in Table 3. The dynamic activity, walking, and all the sedentary activities achieved the highest scores.

Table. 3. Performance of stacked GRU with attention

Activity	Precision	Recall	F1-Score
Walking	1.00	1.00	1.00
Walk Up	0.94	0.98	0.96
Walk Down	0.98	0.94	0.96
Sitting	1.00	0.99	0.99
Standing	0.99	1.00	1.00
Lying	1.00	1.00	1.00

The confusion matrix for the two stacked GRU models with and without attention are given in Fig. 7. The misclassification of walking upstairs and downstairs is due to the similarity in signal patterns gathered from the accelerometer sensor.

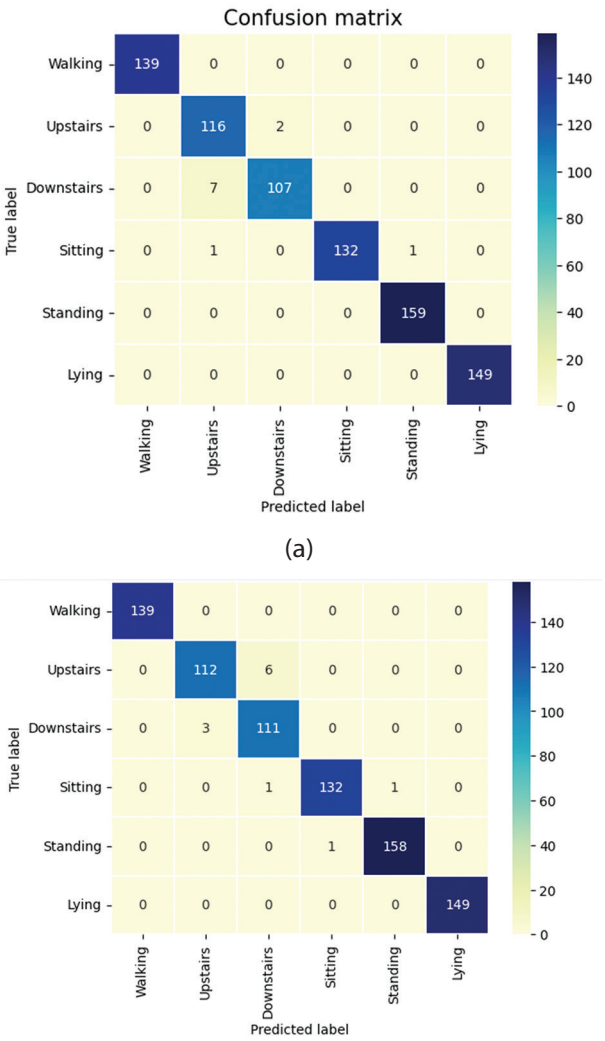
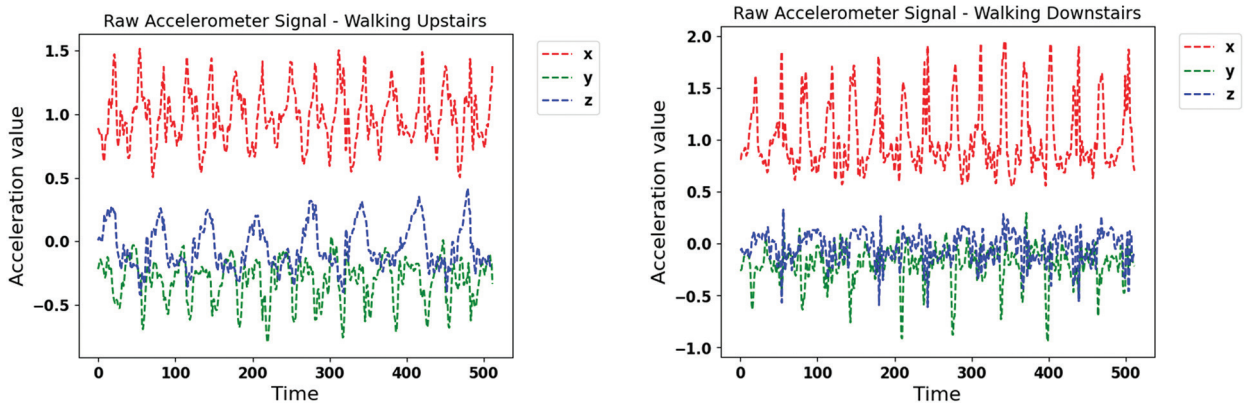


Fig. 7. Confusion Matrix for the two stacked GRU architecture (a) Model with attention (b) Model without attention

The raw signal pattern of these activities is given in Fig. 8. The models are trained with fewer parameters as compared to the baseline architectures. The total trainable parameters used in two stacked GRUs with attention is 51519, whereas the model trained without attention used 51454 parameters. There is only a < 1% increase in parameters when trained with attention. Fig. 8 shows that both activities' axis values fall within a similar range, which can cause misclassification when trained with handcrafted features. The factor that makes the proposed model distinct is fewer neurons in the hidden layers and a single dense layer before the output layer. The training time is considerably reduced in training using features extracted rather than the automatically learned features from the accelerometer vector.

To comprehend how and why a complex model reached a particular conclusion, it helps to analyze the importance of various features in the model. The permutation feature importance values are calculated to learn the model's interpretability. Feature importance arranged in increasing order of significance is given in Fig. 9.



(a) (b)

Fig. 8. Raw accelerometer signal patterns (a) Walking Upstairs (b) Walking Downstairs

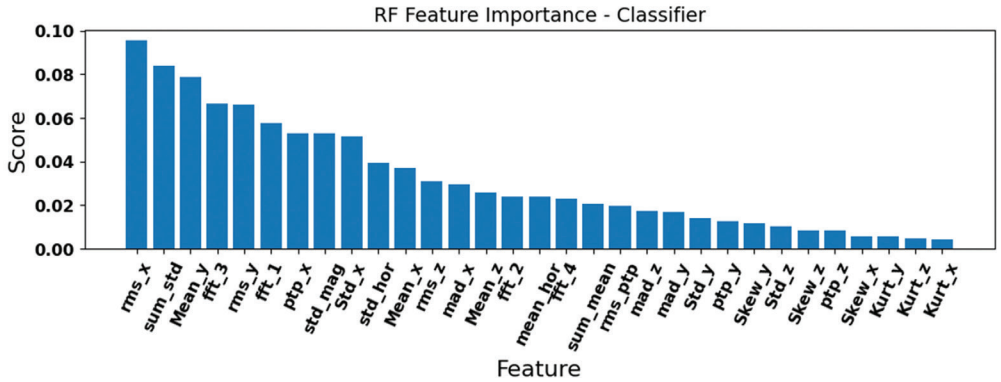


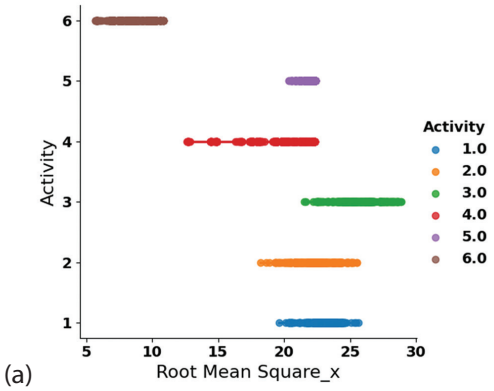
Fig. 9. Permutation feature importance scores in the UCI-HAR dataset

The magnitude vector computed for the x-axes component of the acceleration vector gave the highest score of 0.09565 when feature ranking is performed using the Random Forest Classifier. Permutation feature importance may often lead to misleading interpretations in the presence of strongly correlated features. Figure 1 shows that the features used in the study are correlated. Hence, to determine how a feature contributes to making a prediction, Shapley values, shown in Fig. 10, are a better choice, but their implementation is more expensive. The area of the force plot to the left side from the mean position are the features that helped in prediction, whereas those on the right side would have decreased the likelihood of prediction.

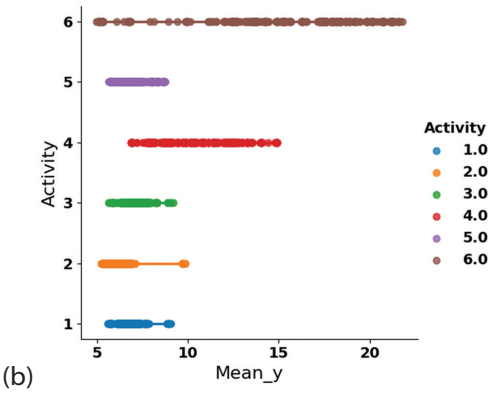
Fig. 10. SHAP explanation force plot for single layer GRU model

The four features with the highest score, namely, magnitude vector along the x-axis, signal mean along the y-axis, root mean square of standard deviation along each axis and fast Fourier transform component, plotted against the activity classes, are shown in Fig. 11.

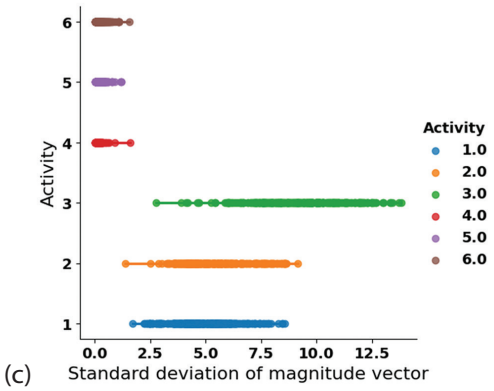
It is observed that lying (activity 6) in Fig. 11 (a) does not overlap with any other activities for the magnitude vector along the x_axis. This is the most separable activity for all models. The dynamic activities of Walking (activity 1), Walking Upstairs (activity 2) and Walking Downstairs (activity 3), which overlap significantly for all features, make prediction hard. Similar to this, the root mean square of the standard deviation along each axis, Fig. 11 (c), and the fast Fourier transform component, Fig. 11 (d), entirely overlap the sedentary behaviors of Sitting (activity 4) and Standing (activity 5), making classification difficult.



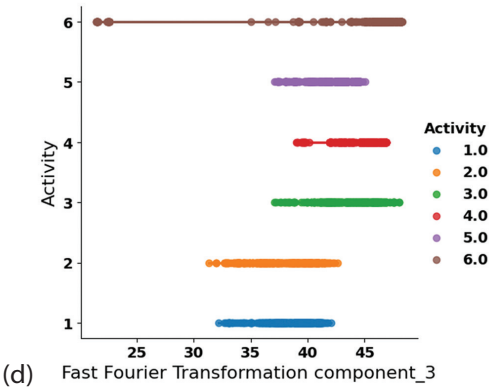
(a)



(b)



(c)



(d)

Fig. 11. Most important feature values against activity classes (a) Magnitude vector along the x-axis (b) Signal mean along the y-axis (c) RMS of standard deviation along each axis (d) Fast Fourier Transform component (Activity: 1 – Walking, 2 – Walking Upstairs, 3 – Walking Downstairs, 4 – Sitting, 5 - Standing, 6 – Lying)

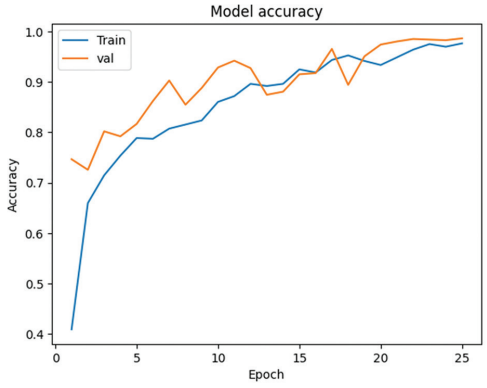
5.2. EVALUATION OF THE MODELS USING OPTIMAL FEATURES

Among the 31 features computed from raw accelerometer sensor data, 23 features representing three-fourths of the total features, arranged according to relevance using permutation feature importance measure, are used to train the models.

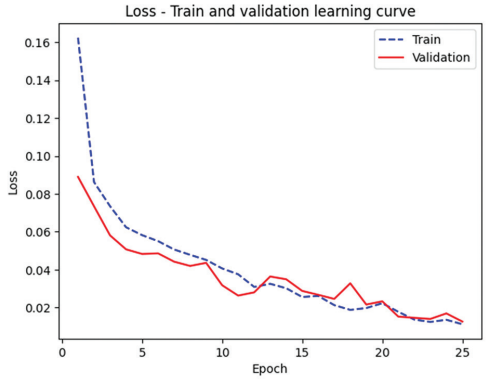
Table 4 shows the comparison between recognition rates of classifiers with and without attention mechanism using three-fourths of the highly relevant features. The introduction of the attention mechanism has improved the recognition rates of all models. The accuracy of models trained with GRU shows high performance compared to LSTM architectures. Figure 12 depicts the training and validation accuracy and loss curves for the GRU model having an attention-based stacked layer.

Table. 4. Recognition rate of classifiers using 75% features selected based on feature importance

Model Architecture	Performance (Accuracy)	
	Model without Attention	Model with attention
Single Layer LSTM	0.85	0.88
Single Layer GRU	0.87	0.88
Two Stacked LSTM	0.93	0.93
Two Stacked GRU	0.95	0.97
LSTM - GRU	0.92	0.93
GRU - LSTM	0.92	0.97



(a)



(b)

Fig. 12. Train and validation curve for stacked attention-based GRU model (a) Accuracy curve (b) Loss curve

The training and validation loss decreases with increased epochs and stabilizes at a value without much gap, indicating that the model is a good fit. It is also to be noted that the decrease in the number of features has not affected the performance of the model.

5.3. EVALUATION OF THE MODELS USING FEATURES GIVEN BY SHAPLEY VALUES

The highly ranked features selected by Shapley values, Figure 9, and fast Fourier transform components, making a total of 16 features, are used to train the model. Table 5 shows the recognition rate of classifiers trained with and without attention using half of the total features originally used in the study, selected based on their ranking.

Table. 5. Recognition rate of classifiers using 50% features selected by Shapley values

Model Architecture	Performance (Accuracy)	
	Model without Attention	Model with attention
Single Layer LSTM	0.85	0.96
Single Layer GRU	0.93	0.95
Two Stacked LSTM	0.93	0.96
Two Stacked GRU	0.96	0.96
LSTM - GRU	0.96	0.95
GRU - LSTM	0.96	0.96

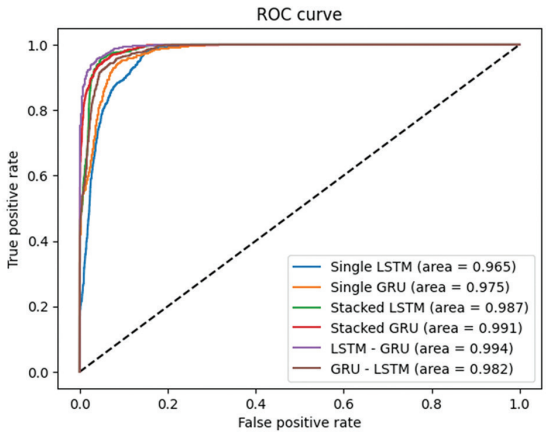
Comparing the results obtained from various study using the lists of all significant features reveals that correlated features minimally impact the performance of the classifiers. When trained using 16 features and attention mechanisms, all models performed comparably. The time overhead of the attention mechanism in single-layer models is less than 2 seconds, and in stacked models is less than 10 seconds.

Table. 6. Statistical analysis of classifiers

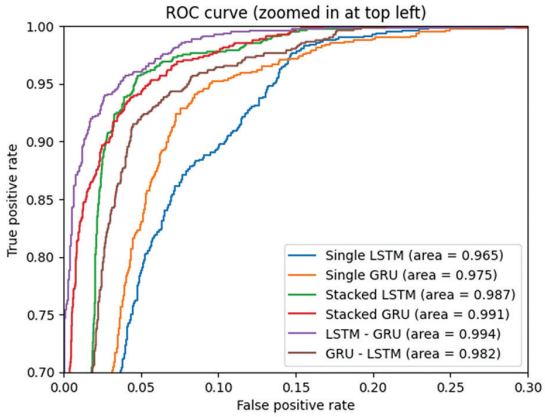
Model Classifiers	Independent t-test	
	T-statistic	p-value
All features	0.06	0.95
75% features	0.88	0.39
50% features	1.43	0.18

An independent t-test is carried out with a significance threshold set at 0.05 to ascertain whether the observed performance difference between the attention-based approach and the non-attention-based strategy is statistically significant. The T-statistic and p-value obtained in the statistical analysis of the classifiers are given in Table 6.

The Receiver Operating Characteristic (ROC) curve is shown for all models trained with optimal features in Fig. 13, illustrates the classification performance. This measure depicts the model's ability to accurately predict the positive class when the result is positive. The Area Under the Curve (AUC) measure in Fig. 12 indicates that all models are good at discriminating various activities.



(a)



(b)

Fig. 13. ROC analysis (a) models trained with optimal features (b) ROC curve zoomed in at top left

The LSTM-GRU combination model has the highest AUC, followed by the two-stacked GRU model. Less false positives are indicated by smaller values on the plot's x-axis, and greater values on the y-axis show more true positives. Table 7 gives the proposed two stacked GRU models compared with other baseline architectures in the literature on the UCI-HAR dataset. The proposed model used a compact architecture with minimum layers and fewer neurons, thereby reducing the number of parameters, to design a lightweight RNN model.

6. CONCLUSION

In conclusion, we have conducted a comprehensive investigation into the classification of human activities using single and stacked recurrent neural network variants with an attention mechanism built into the layer preceding the fully connected dense layer. The attention mechanism enhanced the classification performance of all models, particularly those trained with GRU. Our approach relies on analyzing temporal and frequency domain features calculated from raw accelerometer signal data, offering a robust foundation for human activity classification. The classifier uses fewer neurons and a single dense layer, distinguishing our studies from others in the field that often employ more complex architectures.

The results obtained in the study are given below.

i. GRU performed better than LSTM with the selected handcrafted features to determine human context.

ii. The attention mechanism, when used with stacked variants, improved the overall accuracy of all models.

iii. The significance of features in prediction is determined using permutation feature ranking measures and shapely values. The model is retrained with optimal features that contributed the most, still attaining comparable accuracies.

iv. The accuracy of the GRU stacked models, which reached 99% when trained with all features, dropped by less than 2% when trained with three-fourths of the optimal features and by 3% when only half of the features were used.

In future work, more feature importance measures to attain the interpretability of models having features with high correlation will be attempted.

Table. 7. Comparison of proposed model with baseline architectures

Paper	Year	Model	Layer	Neurons/ layer	Train example	Test Example	Accuracy
[7]	2020	LSTM-CNN	4	(32,32,64,128)	7319	3069	95.8
[26]	2019	GRU with Attention	2	-	7352	2947	94.16
[27]	2023	GRU-INC	2	(144, 128)	-	-	96.4
[28]	2023	CNN-GRU with Attention	4	(128,128)	-	-	94.19
Proposed Model		Stacked GRU with Attention	2	(64,64)	2092	813	99

- The paper does not report these parameters

7. REFERENCES

[1] L. Alawneh, T. Alsarhan, M. Al-Zinati, M. Al-Ayyoub, Y. Jararweh, H. Lu, "Enhancing human activity recognition using deep learning and time series augmented data", Journal of Ambient Intelligence and Humanized Computing, 12, 2021, 10565 – 10580.

[2] C. Xu, D. Chai, J. He, X. Zhang, S. Duan, "InnoHAR: A Deep Neural Network for Complex Human Activity Recognition", IEEE Access, vol. 7, 2019, pp.9893-9902. https://doi.org/10.1109/ACCESS.2018.2890675

[3] C. Xu, J. Shen, X. Du, F. Zhang, "An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units", IEEE Access, vol. 6, 2018, 48697-48707. https://doi.org/10.1109/ACCESS.2018.2867564

[4] T. Alsarhan, L. Alawneh, M. Al-Zinati, M. Al-Ayyoub, "Bidirectional Gated Recurrent Units For Human Activity Recognition Using Accelerometer Data", IEEE SENSORS, Montreal, QC, Canada, 2019, pp.1-4. https://doi.org/10.1109/SENSORS43011.2019.8956560

[5] F. Karim, S. Majumdar, H. Darabi, S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification", IEEE Access, vol. 6, 2018, pp. 1662-1669. https://doi.org/10.1109/ACCESS.2017.2779939

[6] P. Siirtola, J. Röning, "Context-aware incremental learning-based method for personalized human activity recognition", Journal of Ambient Intelligence and Humanized Computing, 12, 2021, 10499–10513. https://doi.org/10.1007/s12652-020-02808-z

[7] A. Tsanousa, G. Meditskos, S. Vrochidis, L. Angelis, "A novel feature selection method based on comparison of correlations for human activity recognition problems", Journal of Ambient Intelligence and Humanized Computing, 11, 2020. https://doi.org/10.1007/s12652-020-01836-z

[8] F. Hernandez, L.F. Suarez, J. Villamizar, M. Altuve, "Human Activity Recognition on Smartphones Using a Bidirectional LSTM Network", in Proceedings of XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), Bucaramanga, Colombia, 2019, pp. 1-5. https://doi.org/10.1109/STSIVA.2019.8730249

[9] E.S.R. Montoliu, O.B. Fernández, "A study of deep neural networks for human activity recognition", Computational Intelligence, 36, 2020.

[10] K. Xia, J. Huang, H.Wang, "LSTM-CNN Architecture for Human Activity Recognition", IEEE Access, vol. 8, 2020, 56855-56866. https://doi.org/10.1109/ACCESS.2020.2982225

[11] N. Dua, S.N. Singh, V.B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors", Computing, 103, 7 (Jul 2021), 1461–1478. https://doi.org/10.1007/s00607-021-00928-8

[12] J.Chen, X. Huang, H. Jiang, X. Miao, "Low-Cost and Device-Free Human Activity Recognition Based on Hierarchical Learning Model", Sensors, 2021, 21(7):2359. https://doi.org/10.3390/s21072359

[13] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing", Journal of Ambient Intelligence and Humanized Computing, 11(1), 2020, 349-361. https://doi.org/10.1007/s12652-019-01214-4

[14] F. Luna-Perejon, M. Dominguez-Morales, A. Civit, "Wearable Fall Detector Using Recurrent Neural Networks", Sensors, 2019, 19. 4885. https://doi.org/10.3390/s19224885

[15] F. Laghrissi, S. Douzi, K. Douzi, B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)", J Big Data, 8, 65, 2021. https://doi.org/10.1186/s40537-021-00448-4

[16] B.A.M. Hashim, R. Amutha, "Human activity recognition based on smartphone using fast feature dimensionality reduction technique", Journal of Ambient Intelligence and Humanized Computing, 12, 2020, 2365-2374.

[17] R.C. Staudemeyer, E.R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks", ArXiv, 2019, arXiv: 1909.09586.

[18] G.V. Houdt, C. Mosquera, G. Nápoles, "A review on the long short-term memory model", Artif Intell Rev, 53, 2020, 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

[19] X. Le, H.V. Ho, G. Lee, S. Jung, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting", Water, 2019, 11(7):1387. https://doi.org/10.3390/w11071387

[20] K. Smagulova, A.P. James, "A survey on LSTM memristive neural network architectures and applications", Eur. Phys. J. Spec. Top., 228, 2019, 2313–2324. https://doi.org/10.1140/epjst/e2019-900046-x

[21] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks", IEEE Transactions on Industrial Electronics, vol. 65, no. 2, 2018, pp. 1539-1548. https://doi.org/10.1109/TIE.2017.2733438

[22] R. Rana, J. Epps, R. Jurdak, X. Li, R. Goecke, M. Breretonk, J. Soar, "Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech". ArXiv, 2016, ArXiv abs/1612.07778.

[23] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, "LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, Vol. 28, No. 10, 2017, pp. 2222-2232.

[24] Y. Yu, X. Si, C. Hu, J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures", Neural Computation, vol. 31, no. 7, 2019, pp. 1235-1270. https://doi.org/10.1162/neco_a_01199

[25] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

[26] M.N. Haque, M.T.H. Tonmoy, S. Mahmud, A.A. Ali, M.A.H. Khan, M. Shoyaib, "GRU-based Attention Mechanism for Human Activity Recognition", in Proceedings of 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-6. https://doi.org/10.1109/ICASERT.2019.8934659

[27] T.R. Mim, M. Amatullah, S. Afreen, M.A. Yousuf, S. Uddin, S.A. Alyami, K.F. Hasan, M.A. Moni, "GRU-INC: An inception-attention based approach using GRU for human activity recognition", Expert Systems with Applications, Volume 216, 2023, 119419, ISSN 0957-4174.

[28] U. Verma, P. Tyagi, M.K. Aneja, "Multi-Branch CNN GRU with attention mechanism for human action recognition", Engineering Research Express, vol. 5, 2, 2023.https://doi.org/10.1088/2631-8695/acd98c