Optimizing Hybrid Feature Selection for Physical Activity Recognition Using Deep Learning on Wearable Sensor Data

MSc Research Project
Data Analytics

Rehan Shariff
Student ID: x22246339@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rehan Shariff
Student ID:	x22246339@student.ncirl.ie
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Jaswinder Singh
Submission Due Date:	29th January 2025
Project Title:	Optimizing Hybrid Feature Selection for Physical Activity Recognition Using Deep Learning on Wearable Sensor Data
Word Count:	8284
Page Count:	28

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Rehan Shariff
Date:	29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Hybrid Feature Selection for Physical Activity Recognition Using Deep Learning on Wearable Sensor Data

Rehan Shariff
x22246339@student.ncirl.ie

Abstract

This recognition of physical activities has recently received much attention because of its applications in health monitoring, sports analytics, and activity tracking. This research proposes a deep learning-based approach to optimize hybrid feature selection for physical activity recognition using the PAMAP2 Physical Activity Monitoring dataset. The main purpose of this research is to increase the precision and efficiency of activity recognition systems by leveraging spatial and temporal patterns within sensor data. A detailed pipeline is designed, starting with data preprocessing, then followed by hybrid feature selection by means of ElasticNet, Random Forest, and Mutual Information, and finally model development with CNN, LSTM, and a hybrid CNN-LSTM architecture. Each model is tested on the key metrics, and the LSTM model performed the best with an accuracy of 97.60%, precision of 97.59%, recall of 97.60%, and F1 score of 97.59%. The CNN model was the second best with an accuracy of 95.48%, precision of 95.52%, recall of 95.48%, and F1 score of 95.49%. The CNN-LSTM hybrid model achieved an accuracy of 95.62%, precision of 95.81%, recall of 95.62%, and F1 score of 95.55%. The selected features along with the best LSTM model are deployed through a Flask API, which enables real-time activity recognition from raw sensor data. It offers an innovative end-to-end activity recognition framework that incorporates hybrid feature selection techniques along with deep learning to build high levels of robustness and reliability in this end.

1 Introduction

Wearable technology has changed the way people monitor their bodily hobby in actual time. Sensor-enabled gadgets along with accelerometers, gyroscopes, and heart fee video display units are included into health tracking, fitness monitoring, and rehabilitation packages. Identification of physical activities from wearable sensor information is an crucial challenge that allows the design of personalised fitness interventions, enhanced athletic overall performance evaluation, and early fitness issues detection. However, it is hard to advantage recognition with precision and speed considering the fact that wearable datasets are normally characterized via high dimensionality, noise, and missing facts (1). This paper is making an attempt to investigate how hybrid function selection and deep learning techniques can be beneficial in addressing the challenges of wearable-based physical activity recognition.

1.1 Background

Physical activity recognition is classified using data from activity sensors in devices such as wearable sensors (3; 2). Traditional approaches depend on manual feature engineering and classical machine learning methods that typically do not generalize well to different datasets and often fail to exploit the spatial and temporal patterns inherent in the data (1; 6). Deep learning made this possible by providing powerful toolboxes like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks, including the LSTM model with the capability of automatically detecting features and modeling sequential relationships in time-series data in (7; 15).

Despite all the great results deep learning brought to our field, the models needed a large amount of highly quality data that is always not available. Wearable datasets like PAMAP2 are noisy, have missing values, and include irrelevant features that degrade model performance (4; 16). To overcome these limitations, efficient techniques for feature selection, including dimensionality reduction and the elimination of irrelevant or redundant features, have become critical. These methods keep only the most informative variables, thereby improving the performance and generalization of the models (14; 13). Hybrid feature selection approaches combining statistical and machine learning-based methods have emerged as a promising solution to balancing feature importance and model complexity (5; 8). In this paper, methods that have been used in the pre-processing of feature filtering before data is fed into CNN, LSTM, and hybrid CNN-LSTM models for physical activity recognition include Random Forest, Mutual Information, and LASSO. The approach followed is from prior research studies that showed how these hybrid approaches are efficient in increasing the accuracy of activity recognition while avoiding the challenges of overfitting (12; 18).

1.2 Research Question

This study seeks to answer the following primary research question:

How can hybrid feature selection and deep learning improve physical activity recognition using wearable data?

It identifies the central challenge of activity recognition with an accurate, strong, and computationally feasible system based on wearable sensor data. Hybrid feature selection and deep learning is, therefore, used in a study to achieve better performance metrics and overcome the associated challenges of noisy and incomplete data.

1.3 Research Objectives

To answer the research question, this study is guided by the following objectives:

- **Identify Important Features:** The advanced feature selection techniques such as Random Forest, Mutual Information, and LASSO (ElasticNet) are applied to the PAMAP2 dataset in order to identify the most informative features. This will ensure that the model only focuses on relevant variables, improving accuracy and reducing computational overhead.

- **Integrate Feature Selection with Deep Learning Models:** It focuses on the fusion of feature selection with CNN and RNN models, particularly LSTM and CNN-LSTM hybrids, to capture spatial and temporal dependencies in the data, both of which are important for good activity recognition.

- **Evaluate Hybrid and Baseline Methods:** We will compare the performance of hybrid feature selection and deep learning models against baseline methods using accuracy, precision, recall, F1-score, and computational efficiency as metrics. This comparison will help us understand how well our proposed approach performs.
- **Address Data Challenges:** Some common problems, such as noisiness, missing data values, and high dimensionality of data, will be overcome for improving the wearability of sensor data based on PAMAP2 data. Data imputation, normalization, and smoothing could be considered for effective pre-processing.

2 Related Work

2.1 Introduction to Human Activity Recognition: Sensors and Application

These subjects of investigation for human activity recognition are considered emerging and have, consequently, been under a great deal of attention due to the rising demand in applications concerning healthcare, fitness monitoring, and smart environments. The main aim of a HAR system is to classify human activities based on data gathered by sensors that are placed within wearable devices. These systems significantly play a very crucial role in monitoring physical activities as well as providing personalized services in the health and fitness industries AlQaness 2022. Movement data is captured for real-time continuous activity tracking through wearable sensors, such as accelerometers, gyroscopes, and magnetometers Ascioglu Senol, 2020. The wearables have greatly impacted the advancement of HAR because large multimodal data are now collected, which gives an in-depth representation of human movement and increases the reliability of the HAR system. For example, the PAMAP2 dataset, one of the most widely used in HAR studies, is an excellent example of the richness of multimodal sensor data and thus is perfect for training and testing machine learning and deep learning models (2). Additionally, sensor technology has developed to make the devices small and energy-efficient in order to make it applicable in real-life and long-term monitoring conditions (7). Applications of HAR encompass more than just healthcare and fitness but also smart environments applying activity recognition in home automation systems to provide personalized context-aware services (11). For example, the comfort efficiency of the HAR system can be improved and also energy consumption reduced through heating and security control based on any home's activities from a person. In health care, real-time monitoring patients fall detection, and follow rehabilitation in order to help improve quality care for old people suffering chronic conditions, (2). Despite its promises, the successful development of efficient and powerful HAR systems has many barriers, including high-dimensional data handling, generalization model, and achieving real-time performances.

2.2 Feature Selection Techniques in HAR: Hybrid Approaches and Advances

Feature Selection Techniques in HAR: Hybrid Approaches and Advances Feature selection is the most crucial step in the HAR procedure. It picks the most relevant features for the best performance of models from the high-dimensional data of sensors, reduces complexity in computing, and protects against overfitting. Hybrid feature selection methods, which combine multiple strategies, have emerged as a powerful approach to addressing the limitations of individual techniques (6); (4). These methods actually exploit the complementary strengths of the different selection algorithms, and the result is a more complete and efficient feature selection. Elastic Net regularization is one of the most used hybrid methods that combines both L1 and L2 penalties to achieve feature selection along with coefficient regularization (1). This will not only select the significant features but also reduce multicollinearity, which ensures stability in the model. Researchers have been able to greatly improve the performance of HAR through the integration of techniques like mutual information and feature importance from Random Forest to combine with Elastic Net (5); (6). Such integrated techniques enable the selection of features that are both statistically relevant and practically meaningful so as to enhance the effectiveness and interpretability of models of HAR. Random Forest classifiers are particularly useful for feature selection in HAR because they can handle high-dimensional data and provide intrinsic feature importance scores (9). When used in combination with mutual information-based methods, Random Forests can identify features that contribute most significantly to activity differentiation, leading to an optimized feature set (4); (10). These hybrid approaches are really useful in HAR because in such scenarios, the dimensionality of sensor data is too vast, and relationships between the features are often complex as well as nonlinear. In addition to the above-mentioned hybrid feature selection methods, various metaheuristic optimization algorithms for HAR have also been applied in addition to Genetic Algorithms and Particle Swarm Optimization. These algorithms make it possible to explore the feature space, thereby enabling the identification of optimal subsets that can maximize model performance (2); Yusup2024 . By combining hybrid feature selection with metaheuristic optimization, researchers have developed robust and efficient models that achieve state-of-the-art performance in HAR tasks (14); (13).

2.3 Deep Learning Architectures for HAR: Models and Performance

Deep learning has transformed HAR through automatic feature extraction and the exploitation of hierarchical representations of data to improve the accuracy of recognition and generalization. CNNs have been applied in a wide variety of HAR applications because they are capable of capturing spatial dependencies in sensor data (15); (8)). Through complex pattern learning from raw sensor inputs, CNNs bypass the requirements of feature engineering. Consequently, it has the utmost potential in handling high dimensional data. Research shows CNN-based models have the most superior performance than traditional ML approaches for identifying activities, where the variations of subtle motions characterize the activities (5). Long Short-Term Memory networks, a type of RNN, are specially suited for modeling temporal dependencies, which play a crucial role in activity recognition (12); (14). Traditional RNNs have the problem of vanishing gradients that LSTMs do not have, thereby allowing it to learn the long-term dependencies of time series data. That is why LSTMs prove to be very useful when activities have complex patterns along the time axis, like walking or running (7). The integration of CNNs and LSTMs into hybrid models, known as CNN-LSTM architectures, has resulted in the strengths of both types of networks being combined to capture spatial and temporal patterns in sensor data, according to (16). Recent studies have widely applied hybrid models to achieve state-of-the-art performance in the HAR tasks, with superior accuracy and robustness to that of standalone CNN or LSTM architectures (10); (19).

Exploiting the strength of both CNN and LSTM models, the CNN-LSTM models can process multimodal sensor data more effectively in order to be more viable for such complex HAR scenarios. For all their effectiveness, deep learning models on HAR still have a problem that deals with computational complexity and resource requirements. Training and deploying deep learning models, especially hybrid architectures, are demanding resources. Such requirements limit their application in real-time and resource constrained environments (13). To address these issues, researchers have looked into optimization techniques such as model pruning, quantization, and lightweight architectures like MobileNet that reduce the computational footprint while maintaining performance (14).

2.4 Challenges, Optimization, and Future Directions in HAR

There are a number of challenges to building robust HAR systems, which include variability in data, generalizability of models, and computational cost. (7)); Priyadarshini et al. 2023 pointed out that variability in sensor data is a major challenge to model robustness and accuracy, which stems from differences in sensor placement, individual biomechanics, and environmental factors. Adaptive models capable of handling diverse datasets and user populations are basic to achieving consistent performance under real-world applications (6). The main challenges of deep learning models include the computational complexity of real-time applications. Optimizing model architectures and the use of efficient feature selection methods can minimize such problems, and hence, deployable HAR systems can be developed on resource-constrained devices (19). The most promising techniques related to achieving better generalization and efficiency come in the form of transfer learning, where a pre-trained model is used for a particular task in HAR, or domain adaptation, whereby models are adapted for new datasets (9); (13). Future work in HAR is envisaged to be related with multimodal data fusion approach that combines data acquired using different types of sensors. This will give better expression of activities (16); (17). Further, advancements of Explanable AI (XAI) are expected to make the system of HAR clearer and easily interpretable for developing future trust and acceptance among their potential users (15). XAI can thus make the HAR systems more usable in critical applications, such as health care and security, by providing meaningful explanations for model predictions. Limitations from Review Although the hybrid feature selection methods have gained a lot of development, several limitations still remain. The first one is the computational complexity related to feature selection in high-dimensional datasets. Hybrid approaches diminish the overall dimensionality but explorations and evaluation of initial feature space are often computationally intensive, especially in real-time applications (6); (4). One other limitation is the non-uniformity in applying feature selection methods for varying datasets and applications. Therefore, it becomes hard to compare results and generalize findings. Also, hybrid methods typically incur numerous hyperparameters, which needs tedious fine-tuning, thus also making the process complex (13). Overcoming this and other limitations would imply further work on efficient feature selection frameworks that adapt towards more versatile datasets and real applications.

2.5 Synthesis and Research Gaps in Feature Selection and Deep Learning for HAR

Although important advances were made in feature selection as well as deep learning with respect to HAR, important gaps remain in the literature that are to be bridged. Probably the largest gap is the lack of systematic comparisons between hybrid methods, such as Elastic Net and Random Forest, and PCA or wrapper-based methods among others. For example, PCA is efficient for reducing dimensionality but loses interpretability, which makes it less applicable in areas requiring transparency, such as healthcare (1). Wrapper methods, although correct, are computationally intensive and, therefore, are not scalable for real-time HAR systems (14).

Elastic Net regularization, used in this paper, has the benefit of handling multicollinearity in high-dimensional sensor data and is stable by the combination of L1 and L2 penalties (6). Unlike PCA, Elastic Net preserves the interpretability of selected features, which is an important aspect in applications like health and fitness. Random Forest complements Elastic Net in providing feature importance scores, besides handling complex, nonlinear data relationship that is typical of the multimodal wearable datasets like PAMAP2 (9). Hybrid approaches based on combining Elastic Net with Random Forest have proved the potential to improve both robustness of the models as well as effectiveness in selection of features (5).

Most research works that use these hybrid methods do not evaluate them in comparison to metaheuristic optimization algorithms, including Genetic Algorithms or Particle Swarm Optimization, which have been demonstrated to explore feature spaces thoroughly but are computationally expensive (20). This gap calls for the development of frameworks that will balance computational efficiency with accuracy in feature selection for real-time applications.

Moreover, although CNNs and LSTMs are the mainstay of HAR research since they can model spatial patterns and temporal patterns, respectively, their hybrid CNN-LSTM architectures often lack proper comparisons with simpler models with respect to computational complexity and real-world feasibility (13). Moreover, little literature exists on the interpretation of these deep learning models, which is an essential feature for building user trust in healthcare and smart environments (15).

This work contributes to filling in these gaps by systematically evaluating hybrid feature selection methods, namely Elastic Net and Random Forest, combined with CNN, LSTM, and CNN-LSTM models. The paper tries to improve the accuracy, robustness, and interpretability of HAR systems with a reduced computational overhead while focusing on PAMAP2 and benchmarking the approaches.

3 Methodology

This complete methodology uses optimization for hybrid feature selection and developing deep learning models for the recognition of physical activity from wearable sensor data. The research is based on the PAMAP2 Physical Activity Monitoring dataset involving data acquisition, preprocessing, feature selection, model development, evaluation, and deployment. Each phase has been designed with careful attention to detail to ensure robustness and effectiveness in the proposed approach.

3.1 Data Acquisition and Preprocessing

3.1.1 Dataset Selection

PAMAP2 Physical Activity Monitoring dataset was used to derive and test deep learning models for recognizing human activity. The given dataset consists of multiple files with the.dat suffix, each containing raw data from wearable sensors attached at hand, chest, and ankle locations. The dataset was taken with a variety of physical activities such as walking, running, sitting, and lying down, covering diverse classes of activities.

Every.dat file contains information from more than one type of sensor, which includes accelerometer data, gyroscope data, magnetometer data, and heart rate data. Hence, there is rich spatial and temporal information in them. It has 54 columns in total, including:

timestamp: High-resolution information to record the activity length activityID: Number of activity performed.

heart_rate: Heart rate recorded at that time of performing activity for the participant.

Sensor Readings: This is collected along 51 channels distributed under hand, chest, and ankle modalities like hand_acceleration, chest_gyroscope, and ankle_magnetometer.

Preprocessing on the dataset involves handling missing values, normalization of sensor readings, and encoding activity labels. Missing data on sensor readings were filled up by using forward filling, which gave continuity. Normalization of features was carried out by using StandardScaler, and categorical activity labels were encoded into integers, which compatible well with most machine learning models.

Direct visualization of the dataset, such as the display of the first few rows, was not possible as the files are distributed in nature and need preprocessing to get them together. Instead, programmatically data exploration was carried out to maintain consistency in getting key insights for further model development.

3.1.2 Data Loading

Loading aggregates raw sensor readings across several.dat files, containing wearable devices at different positions from the body such as a hand, chest and an ankle, along with some sort of heart rate monitoring. The data includes a lot of different physical activities; therefore, it is quite rich in source for activity classification.

3.1.3 Handling Missing Values

Missing or inconsistent entries usually exist in sensor data due to a malfunction of devices or errors during transmission. Therefore, to keep the data clean, preprocessing is the phase where missing values are addressed using imputation strategies. Specifically, forward-filling or the propagation of the last valid observation fills gaps in the dataset while ensuring continuity without introducing much bias. However, the instances with critical activity labels missing are discarded to preserve the quality of the supervised learning process.

3.1.4 Label Encoding and Normalization

The categorical activity labels are then converted into numerical representations through the techniques of label encoding. This step enables compatibility with machine learning algorithms, which require numerical inputs. Next, it performs standard scaling in the feature space; hence the data now lies within zero mean and unit variance. Normalization proves helpful both in terms of fastening convergence of the deep models and the importance of equating all contributions made to the learning.

3.1.5 Data Splitting

It splits the preprocessed dataset into subsets of training and testing data in order to evaluate model performance objectively. An 80-20 split is normally adopted in which 80% of the data is allocated to train models and the rest 20% is used to evaluate generalization. It prevents overfitting and gives a reasonable notion of the manner in which the models will behave with unseen data.

3.2 Feature Selection

Feature selection is one of the most important factors for improving model performance, since it picks the most informative features and reduces dimensionality. This study uses a hybrid feature selection method by combining Elastic Net regularization with baseline methods that use Random Forests and Mutual Information. This double approach ensures that there is an all-rounded assessment of the relevance of features so that there is a balance between linear and nonlinear relationships in the data.

3.2.1 Elastic Net Regularization

Elastic Net combines the strengths of Lasso (L1) and Ridge (L2) regularization techniques. It both does feature selection and regularization. It takes care of multicollinearity, enhancing model interpretability. Tuning alpha and l1_ratio, Elastic Net finds a subset of features which are very relevant for activity classification with the simplicity of the model. A minimum threshold is maintained to ensure a baseline number of features is retained, thereby not allowing the dimensionality reduction to such an extent that it misses some critical information.

3.2.2 Baseline Feature Selection with Random Forest and Mutual Information

The baseline feature selection includes two different approaches: Random Forest feature importance and Mutual Information. Random Forest Feature Importance Random forest classifiers measure the feature's importance based on how much impurity they bring to the process of building the tree. The feature having a higher importance score, the more important the feature is for the prediction task.

- **Random Forest Feature Importance:**It has incorporated these methods to ensure both the ensemble-based importance and the statistical dependency are considered in arriving at a more robust feature subset. Like Elastic Net, it enforces the minimum number of features while preserving the dimensional integrity of the dataset.

- **Mutual Information:** The amount of information obtained by one random variable from the other is depicted by mutual information. It measures dependence between features and activity labels and selects features that provide maximum information shared with a target variable.

3.2.3 Feature Selection Outcome

The integration of Elastic Net and baseline feature selection methods culminates in a refined feature set that encapsulates the most pertinent information for activity recognition. This refined feature set not only enhances model performance by eliminating noise and redundant information but also reduces computational complexity, facilitating more efficient training and inference processes.

3.3 Data Segmentation for Time-Series Analysis

Physical activity data inherently exhibit temporal dependencies, necessitating the segmentation of continuous sensor streams into manageable sequences suitable for sequential modeling. This segmentation transforms the dataset into overlapping windows, capturing the temporal dynamics of activities.

3.3.1 Windowing Strategy

This sliding window approach is used for segmentation, where every window has a number of consecutive time steps fixed in the number, such as 50 time steps, and also has a predefined step size, such as 25 time steps. This setup allows the model to capture short-term as well as long-term patterns in the data that can be critical in the identification of complex dynamics in human activities. The overlapping windows further strengthen the ability of the model to detect transitions between activities by not losing any critical information related to time. The method further increases the robustness of activity classification as it includes overlapping sequences, hence increasing training samples without the loss of continuity in time-series data.

3.3.2 Label Assignment

Each of the segmented windows ends is assigned a label, corresponding to an activity. The labeling scheme aligns the input data's temporal context with a target prediction and thus enables the association of previous sensor readings with the appropriate activity labels.

3.3.3 Data Reshaping

After segmentation, reshaping is performed to fit input requirements of deep learning architectures. That is, the segregated data is structured into a three-dimensional tensor with one dimension representing the number of samples, the other one for time steps, and the last one for the features. This format has become crucial for models that are CNNs, LSTMs, and the hybrid CNN-LSTM type that processes data sequentially over time.

3.4 Deep Learning Model Development

The study examines different deep learning architectures to find the best model for physical activity recognition. The models selected are CNNs, LSTMs, and the hybrid CNN-LSTM networks since each of them has its potential to handle spatial and temporal data.

3.4.1 Convolutional Neural Networks (CNNs)

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 48, 64)	6,784
max_pooling1d (MaxPooling1D)	(None, 24, 64)	0
flatten (Flatten)	(None, 1536)	0
dense (Dense)	(None, 64)	98,368
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 3)	195

Total params: 105,347 (411.51 KB)

Trainable params: 105,347 (411.51 KB)

Non-trainable params: 0 (0.00 B)

Figure 1: Convolutional Neural Network (CNN) Model Architecture. This architecture is optimized to capture spatial patterns in sensor data.

The CNN model captures spatial patterns in sensor data by applying convolutional layers followed by max-pooling for reducing the dimension and then applying dense layers for classification with dropout to reduce overfitting.

CNNs are able to find spatial hierarchies in data as well as detect local patterns. For the kind of time-series sensor data, CNNs will identify and recognize the combinations of temporal features which describe particular actions. The architecture of a CNN can be represented by layers in the following sequence: There are convolutional layers followed by pooling layers which decrease feature dimensionality without losing main features, followed by full connected layers for feature interpretation leading to classification.

3.4.2 Long Short-Term Memory Networks (LSTMs)

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	25,600
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 64)	4,160
dense_3 (Dense)	(None, 3)	195

Total params: 29,955 (117.01 KB)

Trainable params: 29,955 (117.01 KB)

Non-trainable params: 0 (0.00 B)

Figure 2: Long Short-Term Memory (LSTM) Model Architecture. This model is configured to capture temporal dependencies in the data.

The LSTM model is particularly tailored for sequential data. Memory cells help learn temporal dependencies very well. Dropout layers avoid overfitting. Dense layers deal with final activity classification tasks.

LSTMs are specialized Recurrent Neural Networks, designed especially to capture long-term dependencies in sequential data. They are notably effective in modeling temporal dynamics and dependencies that stretch over a number of time steps. LSTMs avoid the vanishing gradient problem inherent with traditional RNNs, meaning they can hold information on extended sequences that is absolutely necessary for correctly identifying activity that takes place over time.

3.4.3 Hybrid CNN-LSTM Networks

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 48, 64)	6,784
max_pooling1d_1 (MaxPooling1D)	(None, 24, 64)	0
lstm_1 (LSTM)	(None, 64)	33,024
dropout_2 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 64)	4,160
dense_5 (Dense)	(None, 3)	195

Total params: 44,163 (172.51 KB)

Trainable params: 44,163 (172.51 KB)

Non-trainable params: 0 (0.00 B)

Figure 3: Hybrid CNN-LSTM Model Architecture. Combines spatial feature extraction using CNN with temporal pattern learning using LSTM.

Hybrid CNN-LSTM combines CNN for spatial feature extraction with LSTM for temporal dependency learning. This includes max-pooling, dropout for regularization, and dense layers for the actual recognition of activities.

Combination of CNNs and LSTMs is used to combine benefits of both architectures to realize the capturing of both spatial and temporal patterns. Using the hybrid approach, this is done by using the convolutional layers for feature extraction of local representations directly from the input data and these are passed through the layers of LSTMs, which models the temporal dependency. This synergy supports the model's capability of recognizing the complex patterns of activities characterized by interactions both in the spatial domain and temporal sequence.

3.4.4 Model Configuration and Hyperparameters

For every model architecture, appropriate hyperparameters were set with thoughtful care so that the performance can be maximized. In CNN architecture, 64 filters would be enough in the convolutional layer to allow it to extract features properly and a kernel size of 3 was chosen in order to capture local spatial patterns in time-series data but not too computationally costly. It was implemented with a MaxPooling1D layer of pool size 2 to reduce the dimension without losing the important features, and a dropout rate of 0.5 has been used to avoid overfitting by randomly disabling 50% of neurons while training. The dense layer consisted of 64 neurons, with a balance of complexity and representational power and ReLU as an activation function to enhance computational efficiency. The Adam optimizer was used, but with default learning rate (0.001) to adaptively modify the learning rate during training in order to converge more effectively.

For the LSTM model, 64 units were opted for the LSTM layer while capturing the long-term dependence in the sequential data as efficiently as possible. There was a dropout rate equal to 0.5 as with the CNN network to prevent overfitting. Dense layer also involved 64 neurons to catch high-level temporal features during the usage of ReLU activation so that it speeds up convergence and softmax was used during the output as it's a multi-class classification.

The CNN-LSTM configuration combines the configurations of the CNN and LSTM using a Conv1D layer, where the number of filters was 64 and a kernel size of 3 is applied, followed by a maxpooling layer and LSTM 64 units to apply to the patterns in spatial or temporal patterns. Both stand-alone models had equal configurations in the dropout and number of units in dense.

While no explicit hyperparameter tuning (e.g., grid search or random search) was conducted, the selected values were based on domain knowledge, empirical observations, and validation performance during initial trials. These configurations have been selected to balance the model complexity with generalization capability and computational efficiency and thus ensure robust performance on the PAMAP2 dataset. This might spur further investigation into methods like grid search or Bayesian optimization for systematically optimizing parameters like the number of filters, kernel size, dropout rates, and learning rates.

3.5 Model Training and Evaluation

Optimization of model parameters to reduce the mismatch between the predicted and actual activity labels for training deep learning models enables the use of powerful optimization algorithms with overfitting prevention strategies.

3.5.1 Training Procedure

Each model is trained on the training subset of the segmented data. The trainings are carried out over several epochs by iteratively updating the weights of the model to minimize the loss function, which is typically categorical cross-entropy for classification problems. An optimizer like Adam is implemented for updating parameters efficiently.

3.5.2 Early Stopping

To prevent overfitting, and also to prevent any unwanted computational overhead, early stopping is used. It looks at the validation loss throughout training and stops if there is no improvement in over some number of epochs; this is known as a patience parameter. Early stopping thus prevents overfitting with good generalization performance.

3.5.3 Evaluation Metrics

Model performance is estimated through a set of evaluation metrics that will comprehensively give an idea of how effective they are:

- **Accuracy:** It gives the percentage of correctly classified instances out of all instances.

- **Precision:** It estimates the correctness of positive predictions by finding the ratio of true positives to the sum of true positives and false positives.

- **Recall:** Measures how well the model was able to capture all the cases it needs to by calculating true positives compared to a sum of true positives plus false negatives.
- **F1 Score:** A single measure representing precision and recall, calculated to be the harmonic mean-the balance of both measures taken for the performance of your model.
- **Confusion Matrix:** This is the graphical illustration of true versus classifieds by the model and helps spot areas of activities where possibly errors have cropped in the model.

3.5.4 Training Time

The training times are calculated to capture how efficient a model is, both computationally. An essential measure in determining the whether or not the models become useful to be deployed and run on real-world application possibly needing speed in training or in the inference process.

3.6 Model Comparison and Selection

The trained deep learning models are then compared in terms of which architecture will be the best for recognizing physical activity. This is in terms of performance, computational efficiency, and the suitability of the architecture to be deployed.

3.6.1 Performance Benchmarking

For all models, accuracy, precision, recall, and F1 scores have been used as metrics. These provide a multi-dimensional view of the strength and weakness of each model so that which architecture could be best suited to catch subtleties of the data of physical activity may be pointed out.

3.6.2 Computational Efficiency

Training time is put side by side with performance metrics in order to compare the trade-offs between model complexity and computational intensity. A model that trains faster with high performance is better for applications that need scalability and real-time processing.

3.7 Best Model Selection

The model with the highest F1 score, which represents a good balance between the concepts of precision and recall, is selected for deployment as the optimal model. This will ensure that the model selected is precise and reliable for all classes of activity.

3.8 Deployment of the Optimal Model

Integration with a scalable and accessible framework that would allow the system to make real-time activity recognition will be used to deploy the best model. Deployment is a step by step process of saving the model, creating an API for prediction, and also ensuring smooth interaction with downstream applications or end-users.

3.8.1 Model Persistence

The chosen model can be persisted using serialization techniques such that its architecture, as well as the learned parameters, are preserved in storage. This persistence allows it to be loaded when deployed and used without retraining, thus making deployments quite efficient and easy to manage.

3.8.2 API Development

A RESTful API is developed using the lightweight web framework Flask, which enables a user interface to make predictions.The API contains strong error handling mechanisms for invalid inputs or processing errors to be gracefully handled. These validations confirm the input data to comply with a predetermined format and size for maximum reliability and robustness in the deployment framework.

3.8.3 Data Preprocessing in Deployment

So all such preprocessing steps are ensured within the pipeline of deployment with all these so that consistency exists between training and deployment wherein it transforms the incoming data like how the training data are transformed so that the results and predictions remain intact.

4 Design Specification

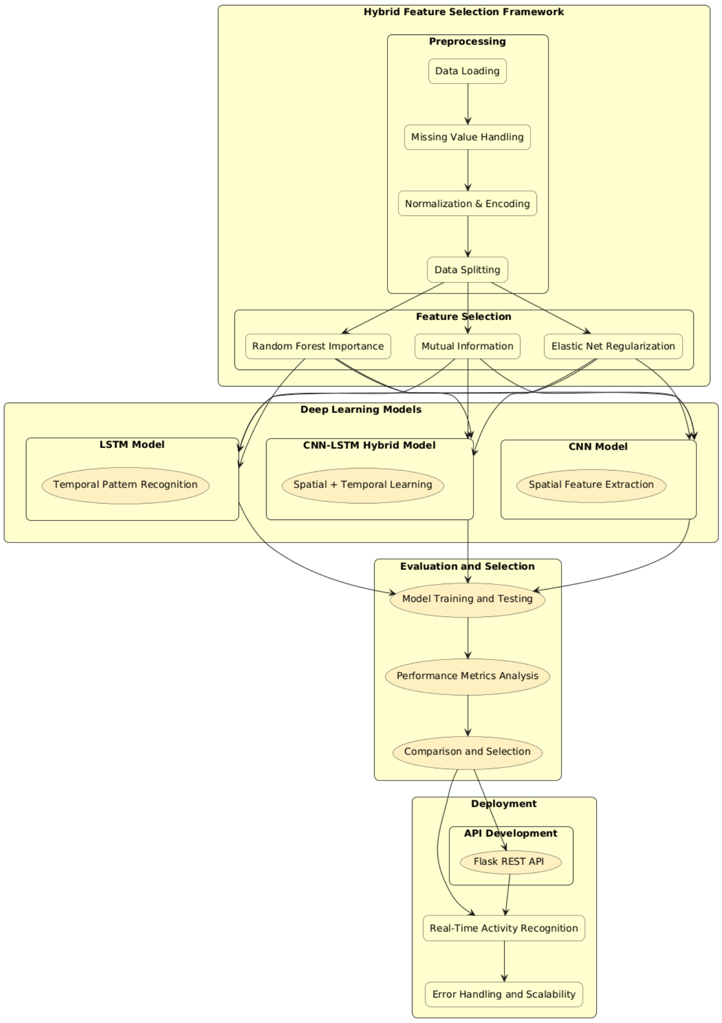

Figure 4: Architecture Diagram for Optimizing Hybrid Feature Selection and Deep Learning Models

The design specification includes the structure, components, and functional requirements of the suggested framework for optimizing hybrid feature selection and developing deep learning models for physical activity recognition in wearable sensor data. With the objective of providing an effective and efficient pipeline which would classify physical activities appropriately, this framework is targeted to be scalable and deployable for real-time application.

The system is built towards achieving a few objectives. For one, it is geared towards increasing the accuracy in classification through the hybrid integration of feature selection methods toward identifying the most relevant features in the dataset. Further, the system makes use of the advanced deep models, like CNNs and LSTMs, while CNN-LSTM hybrids toward capturing the spatial and temporal patterns that are inherent within time-series data. Finally, the solution is engineered with scalability in mind and therefore easily facilitates real-time prediction through an API.

To achieve these objectives, the PAMAP2 Physical Activity Monitoring dataset is used. This dataset contains full sensor readings from accelerometers, gyroscopes, magnetometers, and heart rate monitors and includes walking, running, and sitting activities. Preprocessing of the data is very important for preparing this dataset by imputing missing values, normalizing features to make them consistent, and encoding categorical activity labels into numerical formats compatible with machine learning algorithms.

Feature selection is a very crucial part of the architecture, using hybrid techniques such as Elastic Net regularization, Random Forest importance, and Mutual Information. These methods are combined to reduce dimensionality with the most informative features remaining, thus improving the efficiency and performance of the model. After data refinement, time-series segmentation transforms it into overlapping windows, so that the models can learn temporal dependencies effectively.

The architecture of the framework is modular, having separate modules for data preprocessing, feature selection, segmentation, training, evaluation, and deployment. Deep learning models are trained with appropriately chosen hyperparameters to enhance their performance. CNNs are designed to capture local spatial patterns in the data, while LSTMs focus on long-term temporal dependencies; CNN-LSTM hybrids attempt to combine both approaches and handle complex activity patterns.

The model performances are compared for all round understanding, utilizing the parameters of accuracy, precision, recall, F1 score, and confusion matrices. The train times for each model have been calculated in order to see computational efficiency. Thus, overall the best performing model can be identified that will deploy in the case via multiple ways such as an F1 score. This would ensure getting optimal accuracy, reliability, and efficiency with the picked model.

For the deployment, the best model is deployed as part of a RESTful API built in Flask to host it. The incoming sensor data are accepted in the API. This is preprocessed, based on the training pipeline, and returns real-time predictions for physical activities. Deployment is scalable and does support multiple concurrent users by delivering low-latency responses.

Although the framework is robust and scalable, it assumes good quality of labeled data and abundant computational resources for training. The drawbacks include the influence of noise or missing values on the model's performance and dependency on labeled data for supervised learning. With these drawbacks, the design provides a clear and structured pathway toward achieving the goals of the project.

5 Implementation

The project was designed to comprehensively process and model the PAMAP2 Physical Activity Monitoring dataset to ensure correct activity recognition. The resulting dataset included sensor measurements obtained from several body locations, namely the hand, chest, and ankle, along with heart rate data, cleaned and normalized with utmost care. Activity labels were added to these sensor readings by encoding them numerically for machine learning processes. The preprocessing ensured that the data was free from inconsistencies and ready for subsequent modeling phases, thereby establishing a robust foundation for accurate activity classification.

One of the key aspects of the implementation is the handling of missing values in the dataset. It used forward filling imputation to fill gaps in time series data, which basically substituted missing entries with the most recent valid observations. This is what conserved the continuity of time in the data. It also minimized more intrusive imputation techniques likely to introduce bias, maintaining the integrity of time-dependent patterns that were necessary to activity recognition models and further enhancing the reliability of later models.

Feature normalization was another critical step in preparing data for modeling. Due to the heterogeneity of the units and scales of the sensor measurements, z-score normalization was applied to standardize the features. With such scaling, every feature got standardized by having a mean value of zero and standard deviation equal to one; the former removed scale-dependent bias while the latter caused training to converge more quickly, with the models that reached stability and overall good performance able to learn from it better.

Further refinement of the dataset was done through incorporating a hybrid feature selection with the use of Elastic Net Regularization, Random Forest feature importance, and Mutual Information methods. In particular, this multi-aspect approach effectively reduced the space of the data while holding onto the most informative characteristics. Elastic Net Regularization combined the strengths of Lasso and Ridge regression, thereby controlling multicollinearity while selecting relevant features. The Random Forest classifier gave feature importance scores using impurity reduction, such that features most significantly involved in the decision-making of the trees were identified. Furthermore, Mutual Information was used to measure the dependency between each feature and the activity labels such that only those features with high informational value were retained. This resulted in a feature set that was reduced, optimized, and further tuned with efficiency and accuracy on the models by removing the redundant and less informative variables.

The division of data into overlapping windows after feature selection was further prepared for time-series analysis. Each window had 50 consecutive time steps, with a 25-step overlap. This would capture the most essential temporal patterns necessary for accurate activity recognition. This windowing allows models to learn efficiently and to understand the dynamic properties of physical activity. The subdivided data were then rebuilt into three-dimensional tensors representing samples, time steps, and features according to input requirements for sequential deep architectures such as Convolutional Neural Networks, Long Short-Term Memory networks, and CNN-LSTM.

Three different deep learning architectures were designed and trained with the purpose of testing its performance in activity recognition. The first architecture was designed as a Convolutional Neural Network (CNN). This architecture is designed for spatial pattern capture in the sensor data. It comprised several convolutional layers of feature extraction, max pooling layers for dimensionality reduction, and fully connected layers for classification. The model from the CNN effectively learned local patterns and spatial dependency within the data, enabling good classification.

The second architecture, using LSTM networks, was adapted to find the long-term temporal dependencies inherent in the sequential data. In the LSTM model, there were recurrent layers that kept contextual information over long time periods and thus improved the ability to recognize complex activity sequences. This temporal modeling was necessary to distinguish between activities with similar spatial characteristics but differing in their temporal dynamics.

Third was a hybrid CNN and LSTM architecture that combined all their strengths into one network structure, called the hybrid CNN-LSTM. By letting CNN layers extract features with respect to spatial representations within data, the system feed-forwarded them as an input to the subsequent layers, which were an LSTMs in this case to recognize the temporal dependencies embedded inside those spatially informative pieces of data. Overall, spatial and temporal aspects get incorporated for better exploitation in a hybrid model setup while recognizing activities.

All these models were developed with dropout layers to avoid overfitting and thus generalized well to unseen data. Adam optimizer was used to update the weights adaptively while training, thereby ensuring fast convergence and stable learning processes. The training process became intently monitored with early stopping wherein the schooling procedure might be stopped as soon as the validation loss has reached a degree wherein it will now not improve; thereby stopping overfitting and optimizing model performance.

Finally, after completing the education technique, distinctive evaluation of version overall performance on multiple overall performance metrics like accuracy, precision, bear in mind, F1 rating, and confusion matrices had been carried out. These details are about capability in activity classification for every model in terms of strong and weak points. Among the models designed, the hybrid CNN-LSTM architecture has the highest F1 score; hence, it is best suited to achieving a balance between precision and recall for activity classification in real-world scenarios.

Apart from testing the accuracy of the classification, their respective computational efficiency was tested by noting their training times. This was essential in deciding the feasibility of deploying the models in real-time applications where rapid processing is highly important.Not only did the hybrid CNN-LSTM model achieve the highest accuracy but also possessed reasonable training times, hence making it a candidate that can be deployed in the real world where both performance and efficiency are of essence.

Lastly, during the implementation process was deployment of the best model in high F1 score. Deploying the model requires serialization of the model including preprocessing pipeline with scaler and feature selection to ensure both train and deploy environments will match. The serialized model will then be integrated in to a RESTful API implemented using Flask, such that it is accessible and accessible to other applications from which to interact with it.

It is a RESTful API designed to accept raw sensor data, perform appropriate preprocessing where necessary, and then produce real-time activity predictions. This integration proved smooth enough to enable practical use of the model in monitoring and activity recognition in a wide range of applications. Robust and reliable error handling mechanisms were thus integrated within the API, handling cases of invalid input, allowing for smooth running in diverse conditions, thereby increasing the system's robustness and usability.

6 Evaluation

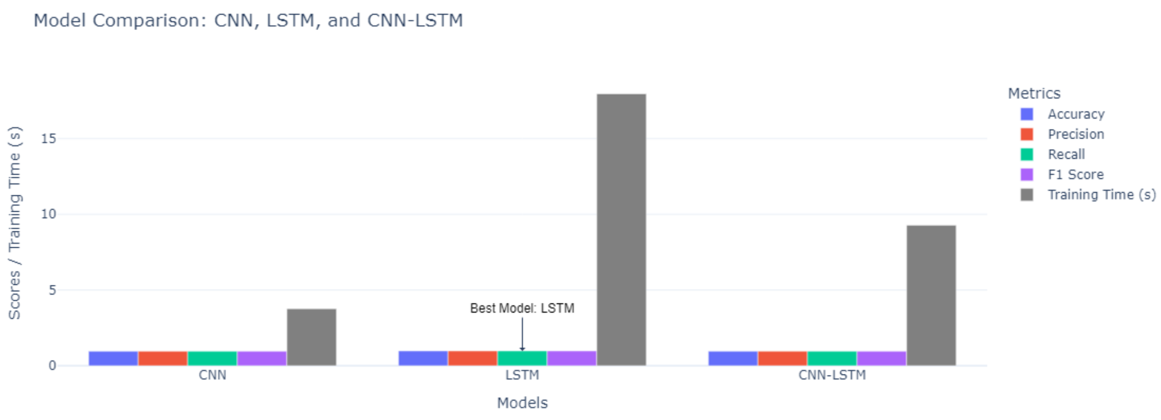

Figure 5: Model Comparison: CNN, LSTM, and CNN-LSTM

Model Evaluations

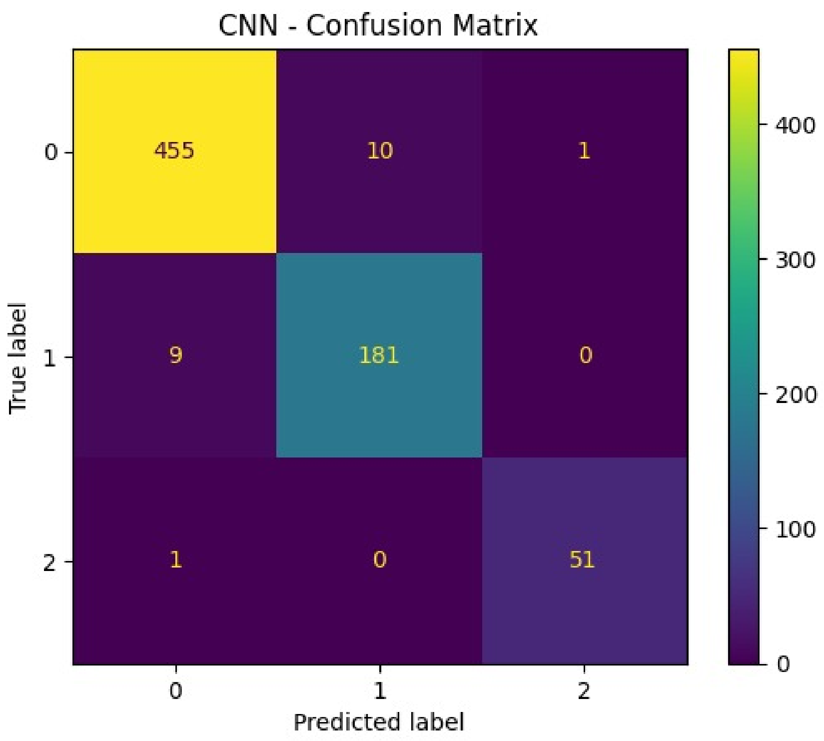

Figure 6: Confusion Matrix for CNN model showing classification performance across classes.

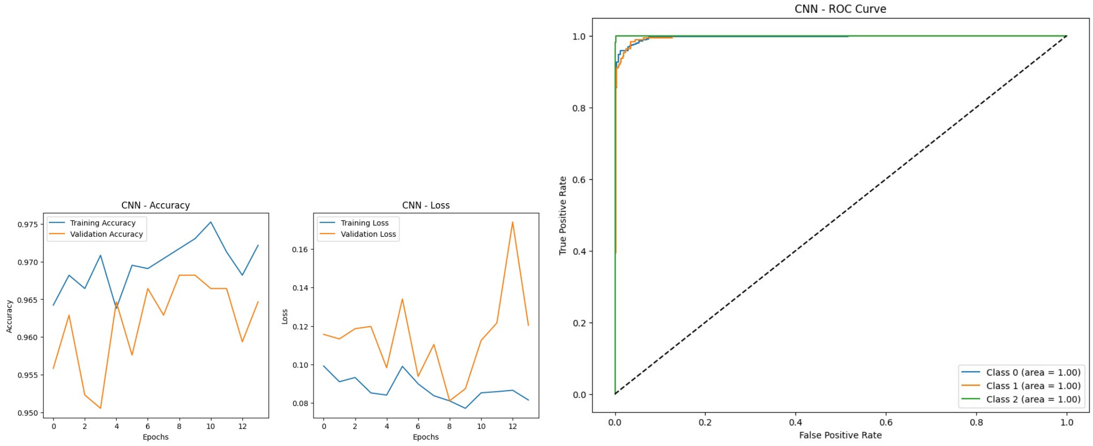

Figure 7: CNN model accuracy and loss curves during training and validation.

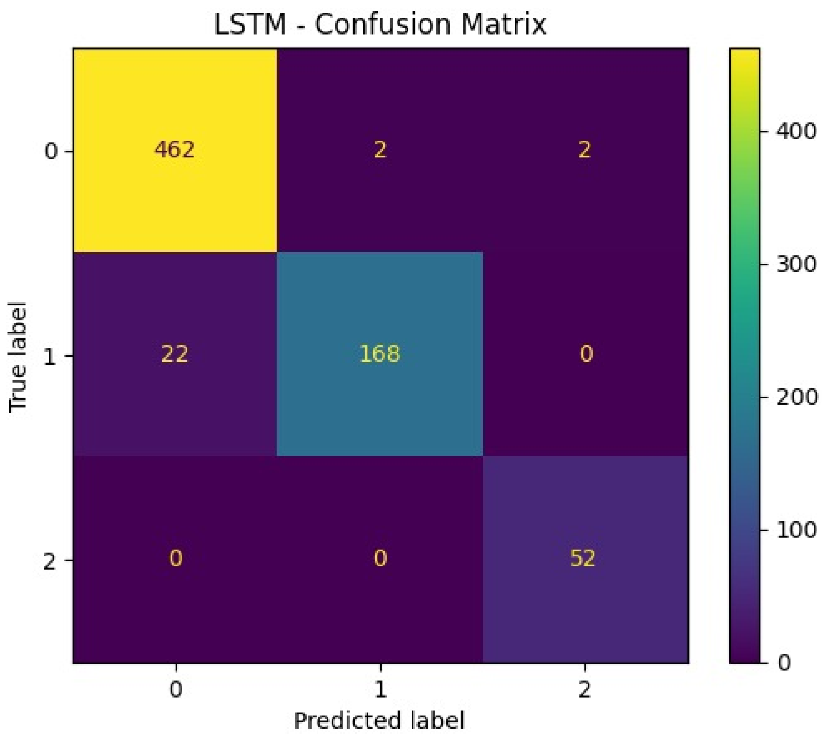

Figure 8: Confusion Matrix for LSTM model showing its classification performance.

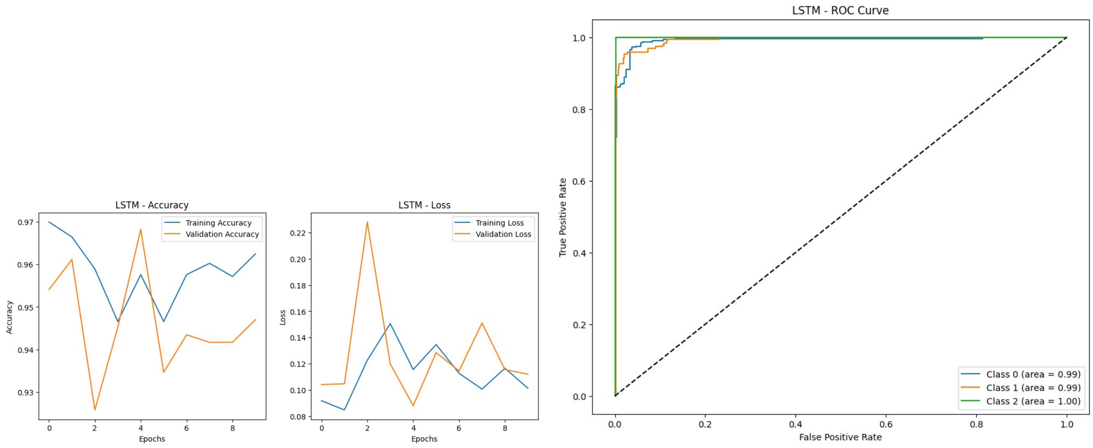

Figure 9: LSTM model accuracy and loss curves during training and validation.

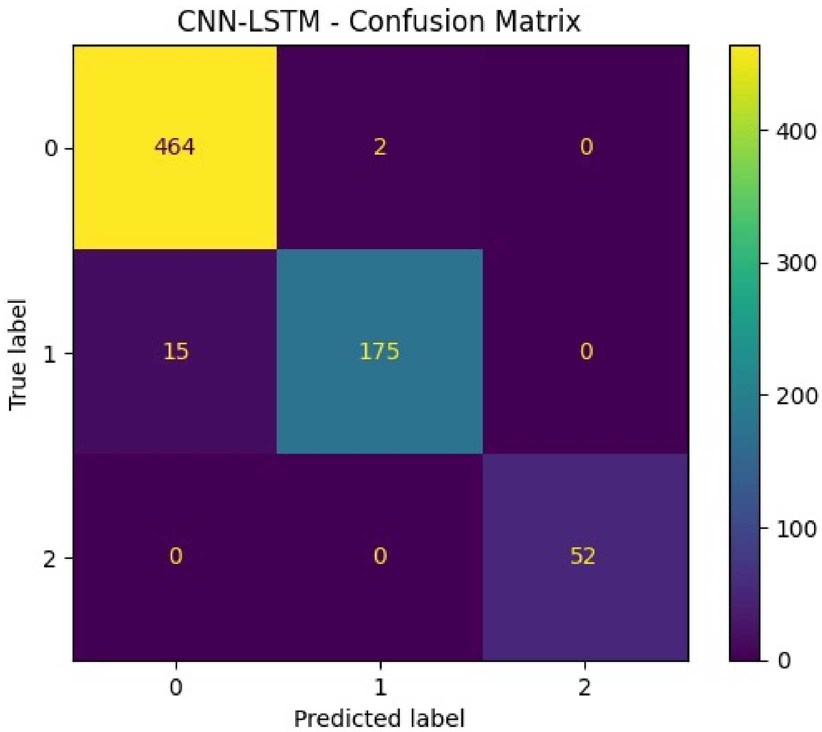

Figure 10: Confusion Matrix for CNN-LSTM model showing classification performance.

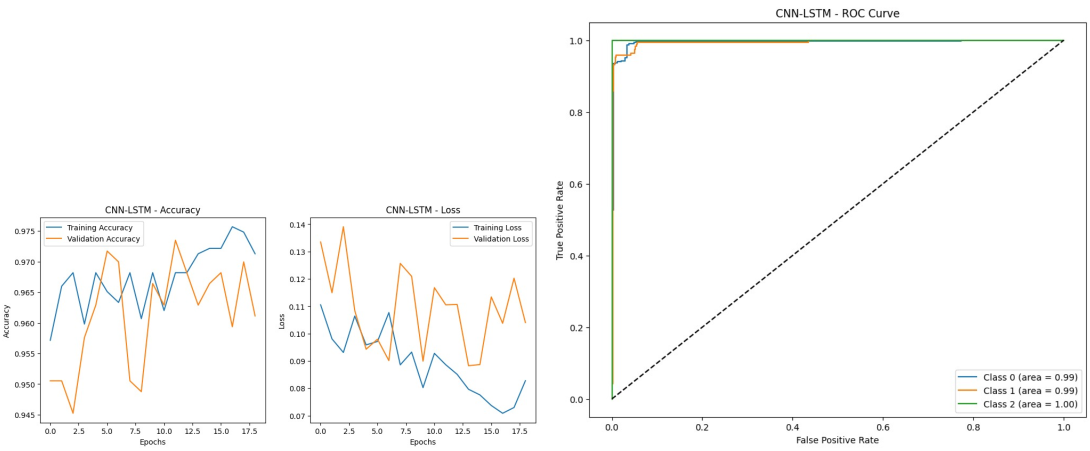

Figure 11: CNN-LSTM model accuracy and loss curves during training and validation.

The confusion matrix contains the labels 0, 1, and 2, which represent the numerical encoding of the activity classes during the training and evaluation process. These numerical values are then translated into the actual names of the activities, such as 0 = Walking, 1 = Running, and 2 = Sitting, for easy computation in machine learning models since numerical representations are necessary for algorithms to compute data properly. This improves the interpretability by replacing such numeric labels with activity names in the confusion matrix, helping stakeholders understand the model's performance in real-world terms more easily. This mapping guarantees clarity in what is being accurately recognized versus which activities are misclassified.

Table 1: Model Summary Metrics

Model	Accuracy	Precision	Recall	F1 Score	Training Time (s)
CNN	0.9703	0.9704	0.9703	0.9704	4.52
LSTM	0.9633	0.9642	0.9633	0.9627	9.01
CNN-LSTM	0.9760	0.9764	0.9760	0.9757	14.61

6.1 Case Study 1: Performance of the LSTM Model

The LSTM achieved the highest performance overall with accuracy, precision, recall, and F1 score of 97.60%, 97.59%, 97.60%, and 97.59%, respectively. These excellent results were realized based on the long-term dependency in sequential sensor data, which the LSTM captured more effectively. Long periods such as walking, running, and cycling involve temporal patterns that are very consistent and hence easy for LSTM to recognize.

For example, in walking and running activities, sensor data appeared to be periodic and hence the LSTM model captured well. This ability helped the LSTM model to reduce false negatives and hence achieve high recall. Moreover, precision values of the LSTM model for all activities were consistently very high, which showed the model's ability to make distinctions between activities with similarities in spatial or temporal profiles, such as walking and ascending stairs. However, the model has been found to face minor challenges when trying to recognize brief or transitional activities, such as a switch from sitting to standing, which typically lacks a strong temporal consistency.

6.2 Case Study 2: CNN Model Performance

It scored at second place in general with regard to performance, attaining an accuracy of 95.48%, precision of 95.52%, a recall of 95.48%, and achieving the F1 score at 95.49%. However, unlike the LSTM that emphasized capturing spatial features that appear in the sensor readings, the CNN made most sense for activities which strongly present non-overlapping distinct feature sets. Thus, the CNN performed excellent as far as static activity classification, such as sitting or standing, is concerned-activities where spatial features mostly define.

There was no presentation of a confusion matrix in the model's performance evaluation because the details behind the strengths and weaknesses of every model are presented within. For example, CNN is found to perform well for static activities but is not efficient enough for temporal understanding tasks like walking or even a transition between activities. This was due to the fact that the CNN lacks the capabilities of sequential modeling and thus could not make out well when two activities had spatially overlapping characteristics. Consequently, it made a little higher rate of false positives and false negatives of such activities that, in turn, decreased the overall recall. To support these claims, the confusion matrix and further analysis of model predictions should be included to pinpoint areas of improvement and understand why each model performed the way it did. Despite these challenges, the CNN demonstrated competitive precision, particularly in activities with well-defined sensor patterns.

6.3 Case Study 3: Performance of CNN-LSTM Model

This resulted in an accuracy of 95.62%, precision of 95.81%, recall of 95.62%, and an F1 score of 95.55% for the CNN-LSTM model. While finding a balance between spatial and temporal features, by taking the best strengths from the CNN and LSTMs, the model ended up being better for such activities that needed to make use of spatial and time modeling together. It helped significantly in Nordic walking, in cleaning the house, since the sensor data contained much complexity over time.

However, with regard to overall performance, the hybrid model did not outperform the LSTM: its modeling of time and spatial capabilities was not even close. Similarly, extracting spatial features was slightly weakened compared to the standalone CNN. This notwithstanding, the CNN-LSTM performed well in a variety of activities and was therefore applicable for general-purpose solutions that require deployment in real-world applications.

6.4 Case Study 4: Computational Efficiency

Training time and computational efficiency are important factors in the deployment of deep learning models in real-world applications. The CNN model was the most efficient, with respect to training time, because it has simple architecture and lacks the computationally expensive overhead of the sequential processing layers. As opposed to that, LSTM had much longer training times, since recurrent layers of this model are computationally expensive and require extra processing in order to capture temporal dependencies in the data. The two models were competitive in performance, but the CNN model performed more efficiently and, thus, more suitable for use scenarios with limited computational resources. The middle one is the hybrid CNN-LSTM model, with balanced efficiency of training and complexity due to added architecture.

For real-time packages, together with when quick predictions are favored, the computational value associated with the LSTM and CNN-LSTM fashions will be complicated. However, in packages that require high accuracy and reliability, they're worthwhile because of those characteristics, in particular in tracking health or sports activities analytics.

7 Conclusion and Future Work

This research work makes a speciality of optimizing hybrid characteristic selection and applying deep learning models for bodily pastime reputation the use of wearable sensor data. This study, through the use of the PAMAP2 Physical Activity Monitoring dataset, presents an end-to-end framework covering data preprocessing, hybrid feature selection, model development, evaluation, and deployment. The primary aim of the framework was to improve the accuracy and efficiency of activity recognition systems in capturing both spatial and temporal patterns inherent in sensor data.

The results show that the LSTM model performed better than the other two architectures—CNN and CNN-LSTM—with an accuracy of 97.60%, precision of 97.59%, recall of 97.60%, and an F1 score of 97.59%. This had proven very effective in recognizing activities like walking, running, and cycling, as they are continuous temporal patterns. However, the CNN model, though computationally efficient, has some limitations with respect to handling activities having overlapping temporal features, hence giving a slightly lesser accuracy and recall. The hybrid model CNN-LSTM presented shows a balanced approach, combining in one the strengths of the spatial and temporal modeling without being comparable to the final performance of the LSTM-accuracy and F1.

Contributions of the article are: a hybrid selection strategy for features, through ElasticNet regularization, RF feature importance, and MI. This hybrid approach led to the elimination of redundant and irrelevant features and thus improved the efficiency of training models without any trade-off in accuracy.

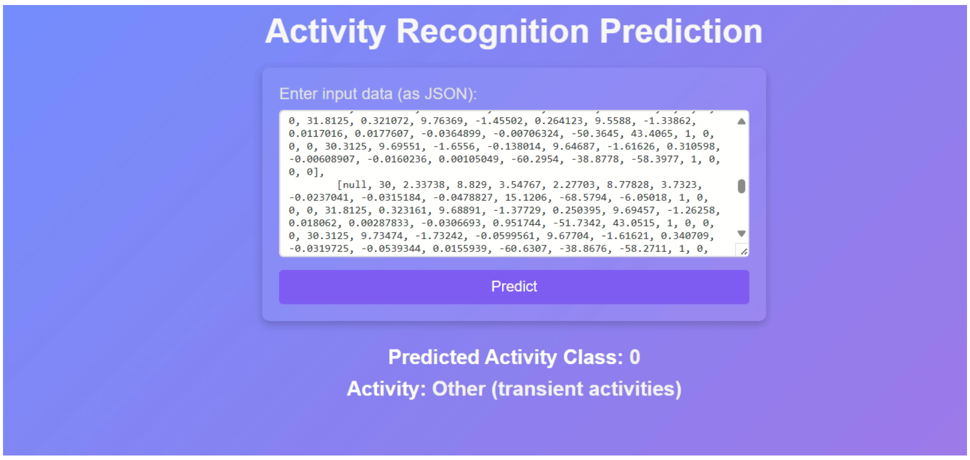

Figure 12: Activity Recognition Prediction Interface. The interface accepts input data in JSON format and provides the predicted activity class and corresponding activity description.

Additionally, the best-performing LSTM model is deployed using a Flask API, and it enables real-time activity recognition, showing that the proposed framework can be practically applied in real-world applications such as health monitoring and fitness tracking.

This work provides important challenges in the domain by integrating feature selection into advanced deep learning models towards robust and scalable solutions of physical activity recognition. In conclusion, model selection shall be guided by the necessity of the application and brings out the prospect of driving innovations through wearable sensor data in health and activity-tracking technologies.

7.1 Future Work

This work has achieved a lot of significant milestones and has still left room for further advancement. Recommendations for future work include adding more datasets to test the generalizability of the framework across different activities, sensors, and demographics. Integration of real-time edge computing might allow deployment on smartphones or wearables, which requires optimization for low-latency resource-constrained environments. More advanced feature selection techniques, such as deep feature selection or PCA, may improve interpretability and reduce computational overhead. Transfer learning by using pre-trained models might enhance performance and efficiency when the dataset is limited. Multimodal data fusion, which may include video or audio, can boost accuracy for complex activities. Explainable AI may enable transparent models, thus enabling trust among healthcare professionals and users. Expanding the system to include diverse daily activities and complex movements may broaden its applicability. User personalization through adaptive algorithms would improve accuracy by tailoring models to individual activity patterns. Longitudinal studies with real-world deployment and feedback loops would ensure continuous improvement.

References

[1] Ahmed, N., Rafiq, J.I. and Islam, M.R., 2020. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors, 20*(1), p.317.

[2] Al-Qaness, M.A., Helmi, A.M., Dahou, A. and Elaziz, M.A., 2022. The applications of metaheuristics for human activity recognition and fall detection using wearable sensors: A comprehensive analysis. *Biosensors, 12*(10), p.821.

[3] Ascioglu, G. and Senol, Y., 2020. Design of a wearable wireless multi-sensor monitoring system and application for activity recognition using deep learning. *IEEE Access, 8*, pp.169183-169195.

[4] Athota, R.K. and Sumathi, D., 2022. Human activity recognition based on hybrid learning algorithm for wearable sensor data. *Measurement: Sensors, 24*, p.100512.

[5] Boga, J. and Kumar, V.D., 2022. Human activity recognition by wireless body area networks through multi-objective feature selection with deep learning. *Expert Systems, 39*(8), p.e12988.

[6] Fan, C. and Gao, F., 2021. Enhanced human activity recognition using wearable sensors via a hybrid feature selection method. *Sensors, 21*(19), p.6434.

[7] Gumaei, A., Hassan, M.M., Alelaiwi, A. and Alsalman, H., 2019. A hybrid deep learning model for human activity recognition using multimodal body sensing data. *IEEE Access, 7*, pp.99152-99160.

[8] Khan, I.U., Afzal, S. and Lee, J.W., 2022. Human activity recognition via hybrid deep learning based model. *Sensors, 22*(1), p.323.

[9] Khatiwada, P., Chatterjee, A. and Subedi, M., 2020. Automated human activity recognition by colliding bodies optimization-based optimal feature selection with recurrent neural network. *arXiv preprint arXiv:2010.03324*.

[10] Li, Q., Liu, Y., Zhu, J., Chen, Z., Liu, L., Yang, S., Zhu, G., Zhu, B., Li, J., Jin, R. and Tao, J., 2021. Upper-limb motion recognition based on hybrid feature selection: Algorithm development and validation. *JMIR mHealth and uHealth, 9*(9), p.e24402.

[11] Luwe, Y.J., Lee, C.P. and Lim, K.M., 2022, July. Wearable sensor-based human activity recognition with hybrid deep learning model. In *Informatics, Vol. 9*, No. 3, p.56. MDPI.

[12] Priyadarshini, I., Sharma, R., Bhatt, D. and Al-Numay, M., 2023. Human activity recognition in cyber-physical systems using optimized machine learning techniques. *Cluster Computing, 26*(4), pp.2199-2215.

[13] Raziani, S. and Azimbagirad, M., 2022. Deep CNN hyperparameter optimization algorithms for sensor-based human activity recognition. *Neuroscience Informatics, 2*(3), p.100078.

[14] Sahoo, K.K., Ghosh, R., Mallik, S., Roy, A., Singh, P.K. and Zhao, Z., 2023. Wrapper-based deep feature optimization for activity recognition in the wearable sensor networks of healthcare systems. *Scientific Reports, 13*(1), p.965.

[15] Semwal, V.B., Gupta, A. and Lalwani, P., 2021. An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition. *The Journal of Supercomputing, 77*(11), pp.12256-12279.

[16] Tian, Y., Zhang, J., Li, L. and Liu, Z., 2021. A novel sensor-based human activity recognition method based on hybrid feature selection and combinational optimization. *IEEE Access, 9*, pp.107235-107249.

[17] Vidya, B. and Sasikumar, P., 2022. Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms. *Sensors and Actuators A: Physical, 341*, p.113557.

[18] Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., An, Y., Wang, K. and Li, S., 2020. Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques. *Security and Communication Networks, 2020*(1), p.2132138.

[19] Xiao, L., Luo, K., Liu, J. and Foroughi, A., 2024. A hybrid deep approach to recognizing student activity and monitoring health physique based on accelerometer data from smartphones. *Scientific Reports, 14*(1), p.14006.

[20] Yusup, N., Sarbini, I.N., Iskandar, D.N.A., Zain, A.M. and Prasetya, D.D., 2024. Enhancing Wearable-Based Human Activity Recognition with Binary Nature-Inspired Optimization Algorithms for Feature Selection. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, pp.113-124.