SENSOR DATA BASED HUMAN ACTIVITY RECOGNITION USING EVOLUTIONARY ALGORITHM AND 1D-CNN

A PROJECT REPORT

Submitted by

Ch GANESH [CB.EN.U4CSE20115]
K S S KARTHIK [CB.EN.U4CSE20129]
KARTHIK G [CB.EN.U4CSE20130]
MAHESH M [CB.EN.U4CSE20143]

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

MAY 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, COIMBATORE – 641 112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **SENSOR DATA BASED HUMAN ACTIVITY RECOGNITION USING EVOLUTIONARY ALGORITHM AND 1D-CNN** submitted by Ch Ganesh (CB.EN.U4CSE20115), K S S Karthik (CB.EN.U4CSE20129), Karthik G (CB.EN.U4CSE20130), Mahesh M (CB.EN.U4CSE20143) in partial fulfillment of the requirements for the award of Degree **Bachelor of Technology** in Computer Science and Engineering is a bonafide record of the work carried out under our guidance and supervision at the Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore.

Dr.Dhanya M Dhanalakshmy
(Assistant professor)
Department of CSE

Dr.Vidhya Balasubramanian
Chairperson
Department of CSE

Dr.Bagavathi C
(Assistant professor)
Department of CSE

Evaluated on:

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, the undersigned solemnly declare that the project report **SENSOR DATA BASED HUMAN ACTIVITY RECOGNITION USING EVOLUTIONARY ALGORITHM AND 1D-CNN** is based on our own work carried out during the course of our study under the supervision of Dr.Dhanya M Dhanalakshmy, (Assistant professor), Computer Science and Engineering, Co-guide Dr. C Bagavathi, (Assistant professor), Computer Science and Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement has been made wherever the findings of others has been cited.

Ch Ganesh[CB.EN.U4CSE20115]
K S S Karthik [CB.EN.U4CSE20129]
Karthik G[CB.EN.U4CSE20130]
Mahesh M[CB.EN.U4CSE20143]

ABSTRACT

Human Activity Recognition (HAR) is the process of identifying and classifying human activities. HAR has been classified into two types based on the type of data generated: Sensor-based, in which human activities are collected using wearable devices and sensors and Vision-based, in which human activities are captured in the form of images and videos. HAR involves the creation of systems and techniques that can automatically recognize and categorize human activities. HAR has various applications in fields such as sports training, security, entertainment etc. The intention of this work is to identify the best model suitable for HAR and to enhance the accuracy and adaptability of the HAR model, enabling it to automatically classify human activities from sensor data. In order to obtain the most appropriate model in the field of human activity recognition (HAR), this study suggests a method based on sensor data. An evolutionary algorithm is utilized to determine the optimal architecture with the best parameters. Initially an Empirical Comparison of Machine Learning (ML), Ensemble and Deep Learning (DL) Models for Sensor-data based HAR is done. Then, an Evolutionary Algorithm (EA) is used to optimize the Convolutional Neural Network (CNN) and hyper-parameters ensuring optimal architecture. Subsequently, a neural architecture search is tried and found the best suitable architecture for CNN. Finally, the tuned parameters are applied to the CNN with the best architecture found which resulted in a robust and highly performing model. To prove that the findings are properly, this work also introduces the implementation of the obtained best architecture and parameters on a real-time raw sensor data which is collected by ourselves. The best model selection will be based on a performance analysis of all the models. The metrics considered for analyzing the performance are number of accuracy, execution time.The obtained HAR system is expected to provide better accuracy in monitoring the daily activities, contributing to the safety and health of elderly individuals etc.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our beloved Satguru **Sri Mata Amritanandamayi Devi.** This acknowledgment is intended to thank all those people involved directly or indirectly with our project. We express our thanks to **Dr. Vidhya Balasubramanian**, Chairperson, Department of Computer Science and Engineering and Principal, School of Computing, Amrita Vishwa Vidyapeetham, **Dr. C. Shunmuga Velayutham and Dr. N. Lalithamani,** Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guides, **Dr. Dhanya M Dhanalakshmy, Dr. C Bagavathi** for their guidance, support and supervision. We feel extremely grateful to **Dr. C. Shunmuga Velayutham, Dr. G. Jeyakumar, Dr .S. Thangavelu, Dr. Ritwik M, Ms. Anisha Radhakrishnan,** for their feedback and en- couragement which helped us to complete the project. We would also like to thank the entire fraternity of the Department of Computer Science and Engineering. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project. Finally, we would like to thank all those who have helped, guided and encouraged us directly or indirectly during the project work.

CH. Ganesh [CB.EN.U4CSE20115]
K.S.S Karthik [CB.EN.U4CSE20129]
Karthik.G [CB.EN.U4CSE20130]
Mahesh Sai.M [CB.EN.U4CSE20143]

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SD Sensor Data

HAR Human Activity Recognition

ML Machine Learning

DL Deep Learning

1D- CNN One Dimensional Convolutional Neural Network

HPT Hyper Parameter Tuning

NAS Neural Architecture Search

GA Genetic Algorithm

LR Logistic Regression

DT Decision Tree

RF Random Forest

KNN K-Nearest Neighbor

SVM Support Vector Machine

NB Naive Bayes

RNN Recurrent Neural Network

Chapter 1

INTRODUCTION

Human Activity Recognition (HAR) is a field within Computer Science and Artificial Intelligence that focuses on developing algorithms and techniques to identify and interpret human activities based on data from various sensors. The goal is to automatically recognize and classify different activities performed by individuals, typically using data from sensors such as accelerometer, gyroscopes, and sometimes other sources like cameras or microphones. Applications of HAR are diverse and can be found in areas such as:

1. Health and Fitness Monitoring -Tracking physical activities to monitor exercise routines, count steps, or assess overall physical health.
2. Smart Homes -Automating home systems based on the occupants' activities, such as adjusting lighting, temperature, or security systems.
3. Assistive Technologies -Aiding individuals with disabilities by recognizing specific gestures or movements for controlling devices.
4. Sports Analytics -Analyzing and tracking athletes' movements for performance improvement or injury prevention.
5. Security and Surveillance -Identifying suspicious or abnormal activities in security monitoring systems.
6. Context aware computing -Adapting applications or services based on the user's current activity and context and can also be used in Automation and Convenience, Research and Insights, Personalized Experiences, Efficiency in Workplaces, Smart Transportation, Enhanced Human-Computer Interaction.

The general recognition process involves collecting sensor data, pre-processing it, and then using machine learning or pattern recognition algorithms to classify the data into specific activities. Many research works are already existing in the field of HAR which tried out various possible Machine Learning (ML) and Deep Learning (DL) models, and each of the research work favoured different models. All the research works, concluded that some particular models work better that were implemented on some standard datasets. The different datasets available in the field of HAR are UCI-HAR, WISDM, MHEALTH, DEVICE MOTION dataset, KU-HAR, AMU-HAR etc. In this experimental comparison, the ML and DL models considered are Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Ensemble ML models. This work intends to find the best model that can work well for UCI dataset in the field of HAR. For achieving that, the idea is to empirically analyze all the possible models implemented using this dataset. In order to identify the best model, the performance of these models were compared on the basis of some metrics like accuracy, execution time and number of features. The best model is identified by this work and efforts are made to further improve the model by tuning its Hyper Parameters in the future scope. The problem statement for this proposed work can be summarized as 'perform the empirical analysis of popular ML, Ensemble and DL models to identify best model suitable for HAR, finding the best Hyper Parameter setting through Evolutionary Algorithm (EA) so that the results are improved and evolve the best architecture that performs well across various datasets'. The prominent models that are used in HAR have been studied thoroughly and the details are described in Chapter 2. The experimental setup and the methodology for this comparative study is elaborated in Chapter 4. Then, Chapter 5 explains the results, analysis and the observations of all the implemented models. Chapter 6 concludes the paper with major findings from the whole experimental comparison and our future scope.

1.1 Problem Definition

Human Activity Recognition is a process of identifying and classifying human activities. The two types of HAR are: Sensor-based, in which human activities are collected using wearable devices and sensors and Vision-based, in which human activities are captured in the form of images and videos. Nowadays, Human Activity Recognition is emerging as one of the most trending research areas since it supports a wide range of applications, especially in human tracking and sensing and also Health and Fitness monitoring, Smart Homes, Assistive Technologies, Sports Analytics, Security and Surveillance etc. Together with the advancements of embedded devices and sensors, we recognize that smart-phones can be exploited as an efficient way to perform human activity recognition since they have become an essential part of humans daily life. Hence this project proposes a method on sensor data based HAR. However, the accuracy, execution time and memory consumption are some major challenges in developing a recognition algorithm on smart-phones. In this project, an attempt is made to tackle the challenges by adopting the Evolutionary Algorithm to optimize CNN model and also a Neural Architecture Search (NAS) is done on the CNN model using GA to find the best architecture in the field of sensor data based HAR.

1.2 Motivation

Smart phone-based HAR systems that use traditional machine learning algorithms or deep learning techniques are becoming increasingly popular. Since, it extracts the relevant characteristics that are responsible for discriminating diverse activity patterns, feature engineering is a dominant phase in classical ML approaches. The feature extraction of raw signals has a significant impact on the accuracy of HAR solutions. The characteristics are then input into classifiers, which identify human behaviors. Conventional classifiers fail to recognize physical activities competently and correctly without a proper feature engineering approach. As a result, extensive data preparation techniques are necessary to deliver sensory data in a suitable format, and handmade characteristics are derived from the collected sensory data based on expert domain knowledge. Also in this project, improving accuracy and adaptability are given more priority along with reducing the execution time and memory consumption.

1.3 Problem use cases

Health-care:

HAR has dominated the health care industry, enabling applications such as remote care for elderly people living alone.

Surveillance:

HAR is used in surveillance systems to detect abnormal activities and ensure security.

Sports Performance Analysis:

HAR can track the position of human joints, which is useful for tracking various sports performance metrics.

Wellness:

HAR systems can be utilized in wellness applications to monitor and analyse activities related to fitness and well-being.

Smart Home/Office/City:

HAR can be used in smart environments to automate tasks and enhance the overall experience.

1.4 Problem Statement

To enhance the accuracy and adaptability of HAR by conducting an empirical analysis of various ML models, DL model and Ensemble approaches, the initial phase was to focus on evaluating these models to identify the most suitable model for HAR then employing an EA for neural architecture search and to optimize the model architecture and parameters.The optimized model will be implemented to process real time raw sensor data aiming to improve the performance and applicability of HAR in real life applications.

1.5 Project-Flow

In module 1,An empirical comparison is done on UCI HAR dataset by various ML, Ensemble and DL models and the best model is selected based on accuracy. In module 2, the best model's parameters is tuned using EA and the best configuration of parameters are found. Since, the tuned model gave a better result in module 3, NAS is used to find the architecture of the model that performs good across selected datasets. Also to check the performance of the model a custom dataset is prepared with the real time data as input.

The prominent models that are used in HAR have been studied thoroughly and the details are described in Chapter 3. The experimental setup and the methodology for this comparative study is elaborated in Chapter 4. Then, Chapter 5 explains the results, analysis and the observations of all the implemented models. Chapter 5 concludes the paper with major findings from the whole experimental comparison and our future scope.

Chapter 2

LITERATURE SURVEY

2.1 Research Works

2.1.1 Machine Learning

Farzana Kulsoom and Sanam Narejo (1) used ML techniques such as Decision Trees, Random Forests, and Support Vector Machines for human activity recognition using smart-phone sensors.

Zaki et al. (2) in their research made a comparative analysis using five classifiers - KNN, Naive Bayes, Random Forest, Gradient Boosting and Logistic regression and evaluated performance using two publicly available datasets of HAR. They concluded that the logistic regression model outperformed all the classical machine learning models.

Janaki and Geethalakshmi (3) experimented on An Automated System for Human Activity Identification. The paper focuses on developing an automated system for identifying and monitoring human activities in elderly people using data analytics and artificial intelligence techniques. The authors collected multidimensional data and employed machine classification approaches such as KNN, SVM, DT, and NB. The experiments showed that the decision tree algorithm achieved remarkable outcomes.

Priyadarshini et al. (4) proposed "Human activity recognition in cyber-physical systems using optimized machine learning techniques". It used machine learning algorithms like Random Forest, Decision Trees, K-Nearest Neighbors, Convolutional Neural Networks, Long Short-Term Memory, and EAted Recurrent Units for HAR. The study incorporated optimization techniques to improve model performance, using Stochastic Gradient Descent, Adam, and RMSProp optimizers. The evaluation of the models is done based on Accuracy and F-1 score, and the study shows that optimization techniques enhance model performance

2.1.2 Deep Learning

Saurabh Gupta (5) discussed the use of CNN for HAR using sensor fusion. The authors proposed a novel approach that combines multiple sensors to improve the accuracy of HAR.

Ehab Essa (6) suggested two unique architectures for classifying sequences of human activity data from various sensors: the temporal channel convolution with self-attention network (TCC-SNET) and the convolution with self-attention network (CSNet). A collection of convolution layers are used by CSNET to process the input sensor data features in sensor reading. These layers capture local features in sensor reading, and temporally aware channel relationships are captured by TCCSNET.

Ankalaki and Thippeswamy (7) conducted a comparison analysis of the most advanced algorithms suggested on HAR that are based on deep learning and machine learning techniques. A thorough investigation has been carried out to determine how various parameters, such as pooling, activation functions, the number of dense layers, and the dropout percentage of CNN, affect recognition accuracy.

Akter et al. (8) introduced a new HAR method using CNNs that effectively utilizes mobile sensor data for applications in healthcare and monitoring. It combines multi-stage convolutional features with an attention mechanism for refined feature extraction, incorporating CBAM modules for a more informative model. The method, tested on three datasets, showed high accuracy, outperforming previous models.

Shibo Zhang et al. (9) highlighted the transformative role of mobile and wearable devices, equipped with various sensors, enable applications that enhance daily life through activity tracking and wellness monitoring. Deep learning has notably improved human activity recognition (HAR) on these devices, leading to a detailed review of recent advancements and future research directions in the field.

Daniel et al. (10) discusses the classification of five human activities utilizing data from smartphone sensors. It evaluated various ML models, including traditional algorithms and advanced neural networks, to determine their effectiveness in Human Activity Recognition (HAR). The study identified Support Vector Machines (SVMs) and a 1D Convolutional Neural Network as the top-performing models in this domain.

Omar Sh. Ahmed Aboosh et al. (11) focused on the use of hybrid deep learning models that based on convolutional neural networks (CNN) and recurrent neural networks (RNN) methods for fake video detection. The inceptionV3 model was used to extract facial features from the frames, then these features were used to train simple RNN and Gated Recurrent Unit (GRU) models to classify video. Most deepfake detection works fails when tested on a new dataset, especially those that are real and close to reality. Therefore, the most realistic dataset which produced 'in the wild' was chosen in this research. The deepfake detection challenge (DFDC) dataset was used to evaluate the proposed models. Where these models achieved a high detection accuracy, 98.5 percent for Simple RNN and 98.9 percent for GRU. Also, the models achieved 0.979 and 0.986 of AUC respectively.

Seamus Lankford and Diarmuid (12) suggested on using Open NAS's Swarm Intelligence (SI) components to train and optimize CNNs. The PSO and ACO major types of SI algorithms are compared to determine which is better at producing higher model accuracy. Our experimental setup demonstrates that the PSO algorithm outperforms ACO. PSO's performance gain is especially noticeable when dealing with more complicated datasets. In the context of this study, the output of swarm intelligence algorithms has produced impressive performances. Their effectiveness is frequently only slightly superior to adjusted pre-trained VGG models, though. In the image classification of grayscale and color datasets, it has been demonstrated that the accuracy of PSO derived models is higher than that of ACO derived models.

Zainab abed Almoussawi et al. (13) proposed a deep learning using Convolution Neural Network (CNN) with Channel Attention Module. The FER 2013 dataset is utilized in this research for effective classification of face detection. The Data Augmentation is used in this experiment for data pre-processing and feature extraction using high feature generation pyramid (HFGP) and low feature generation pyramid (LFGP). Face detector using Single-Short multi biox detector (SSD) and ResNet 10 based face detection. Then, face detector is given input to channel attention module classifier which utilized Deep Learning (DL) for effective classification. The obtained result show that the proposed DL using CNN model achieves better accuracy of 99.90 percent on FER 2013 dataset which ensure accurate classification compared to other existing methods like Engagement Index, Deep Neural Net and Zoning based model.

Ajay et al. (14) explored the impact of different neural network architectures that are currently in use to comprehend static frames in action perception. Based on a range of metrics, the optimal and most suitable model for action recognition is then determined by utilizing different neural network architectures, such as VGG-16.

2.1.3 Comparison and Ensemble models

Shakya et al. (15) compared ML DL models on WISDM Shoaib SA Datasets. It compares traditional machine learning classifiers such as Random Forest (RF), Decision Tree (DT), K-nearest neighbour (KNN) and DL models explored are RNN and CNN. The major observations are that using raw data from accelerometer sensors reduces the need for hand-crafted feature extraction in ML classifiers and DL models, especially CNN, exhibit higher accuracy when trained on balanced datasets with data from multiple accelerometer sensors.

Muralidharan et al. (16) focuses on classifying human activities like sitting, stand- ing, walking, climbing stairs, and laying down using smartphone sensor data. Machine learning models, including classic algorithms and advanced neural networks, are evaluated for their effectiveness in Human Activity Recognition (HAR), with SVMs and a

1D Convolutional Neural Network emerging as top performers.

Voicu et al. (17) examines the use of smartphone sensors to identify human activities like sitting, walking, and climbing stairs. It compares ML models, emphasizing the effectiveness of Support Vector Machines (SVM) and 1D CNN for Human Activity Recognition. The study also proposes using deep learning with CNN and LSTM on public video datasets for activity detection without wearable sensors.

Balaha and Hassan (18) In order to solve the HAR problem, a thorough investigation of sensor-based human activity identification has been conducted. Machine and deep learning approaches, in addition to several conventional dimensional reduction and TDA feature extraction techniques, are offered.

Rojanavasu et al. (19) focused on Sensor-Based Human Activity Recognition (S-HAR) using wearable inertial measuring instruments. Proposed method employed an ensemble of deep learning networks (CNN, LSTM, and ResNet) with sensors on the waist, chest, leg, and arm. Utilized a public dataset (Smartphone and Supporting Nodes) with eight human actions. The proposed Ens-ResNeXt model outperformed existing techniques in terms of accuracy and F1-score. Ens-ResNeXt demonstrated superior performance in comparison to individual DL models.

Pramila et al. (20) extracted temporal and spatial components from the smartphone signals regarding movement data related to eight different activities and the data was divided into single and double channels. Alongside the time domain, raw data was shown in the Fourier and wavelet domains. Deployed CNN models under-performed compared to a deep neural network using a double-channelled time-domain input.

2.1.4 GA and Hyper Parameter Tuning

Sarkar et al. (21) suggested guided mutation technique to improve chromosomal fitness. Next, a modified version of the Genetic Algorithm (GA) is used to eliminate the features that score lower, leaving only the best set of features. Subsequently, the KNN classifier is employed to classify human activities. Spatial attention assisted CNN was utilized to extract features, and several filter techniques such as Mutual

Information and Relief-F were employed to rank the features derived from the CNN model.

Raziani and Azimbagirad (22) experimented with sensor-based human activity recognition Deep CNN hyperparameter tuning algorithms. In order to determine the CNN model's ideal hyperparameters automatically, the authors looked into seven metaheuristic techniques. The UCI Machine Learning repository's HAR dataset is used to test the algorithms.

Young et al. (23) emphasized the recent popularity of Deep Learning (DL) in voice and image applications due to its comparatively automatic feature generation and, especially, high accuracy classification capabilities for convolutional neural networks (CNNs). Model selection (as architectural creation) through hyper-parameter selections is still a laborious and highly intuitive operation, even though these models learn their parameters through data-driven approaches. In response, a technique called Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL) is put forth to automate network selection on computer clusters via genetic algorithm-based hyper-parameter optimization. The experiment's conclusion is that the number of filters was not as important as the kernel size for the convolutional layers. A significantly lower kernel size produced the greatest outcomes in the

Shrestha and Mahmood (24) explains the idea of mtDNA and how it can be used to enhance GA by incorporating crossover control that is based on heuristics, which can be used to direct the search for the ideal hyperparameter values within the large solution space. Additionally, the training technique and dataset optimization are covered in more detail in this part. It also explains how machine learning (ML) training can be enhanced by using the cloud. The addition of mtDNA and importance sampling resulted in a.4% improvement in overall accuracy.

Guo et al. (25) suggested a unique approach for hyper-parameter optimization is pre-aimed to achieve the effective search for learning algorithm hyper-parameters by combining the benefits of a genetic algorithm and tabu search. The Tabu-Genetic Algorithm is the name given to this technique. The Tabu-Genetic Algorithm offers superior search capabilities and is a useful technique for determining the hyper-parameters of learning algorithms in both low- and high-dimensional spaces. This study presents a novel approach to address the hyper-parameters optimization problem of complicated machine learning models, leading to improved performance of machine learning algorithms in real-world scenarios.

Emmanuel Okewu et al. (26) Using the MNIST database of handwritten images and Python deep learning modules, a neural network technique for deep learning was devised, and the results were experimentally assessed. Even though the training time and loss quality of these extensions of gradient descent as the cardinal optimization method for deep learning neural networks have significantly improved over time, there is still room for improvement. In particular, experimental results showed that optimizers using adaptive moment estimation, such as RMSProp and Adam, are producing better results.

Sehla Loussaief and Afef Abdelkrim (27) looked into CNN as a technique for classifying images. The configuration of the network's hyperparameters, which include the quantity of convolutional layers, the number of filters in each layer, and their corresponding sizes, determines the network's performance. Our specific challenge is to use CNN to create a prediction model for the classification of sign stop images. Initially, we attempt to manually create the CNN architecture. The categorization accuracy of this method is a pitiful thirty percent. Therefore, in order to enhance the caliber of the network training, we decide to add a batch normalization layer after every convolutional layer. The performance of the Enhanced E-CNN-MP surpasses that of the initial design.

Ismail Damilola Raji et al. (28) introduced the SDSGA with the goal of effectively finding the ideal ML model hyperparameters. The performance of the developed SDSGA was then assessed using a comparative analysis approach, and it was compared with existing MOAs, including the genetic algorithm (GA), particle swarm optimization (PSO), and biogeography-based optimization (BBO) approaches. To preserve the computational resources needed for training machine learning (ML) models, we employed small population sizes (CNN and RF). Therefore, the less time spent training the model, the fewer function evaluations that are required to determine the ideal HP values.

Xueli Xiao et al. (29) suggested that the hyperparameters of a CNN be automatically and methodically adjusted using a variable length genetic algorithm (GA) in order to enhance performance. Based on experimental results, our system is able to efficiently find good CNN hyper-parameters. Our tests show that better outcomes may be obtained if more effort was spent adjusting the hyperparameters. The CIFAR-10 dataset was used for the experiments, and the outcomes are contrasted with those obtained through large-scale evolution, classical genetic algorithms, and random search. Based on experimental results, our method is able to find substantially better models in a limited amount of time.

Khader M. Hamdia et al. (30) outlined an approach for optimizing ML model feature combinations and architecture while taking supervised learning into account. The suggested method uses genetic algorithm (GA)-based integer-valued optimization for two machine learning models: adaptive neuro-fuzzy inference system (ANFIS) and deep neural networks (DNN). The amount of hidden layers, neurons, and activation function in DNN optimization issues are the selected factors, but in ANFIS optimization, the type and quantity of membership functions are the design variables. The optimization fitness function is the mean squared error (MSE) between the goal outputs and the predictions.

X.J. Luo et al. (31) To predict week-ahead hourly building energy usage, an integrated artificial intelligence-based approach comprising feature extraction, evolutionary optimization, and adaptive DNN model is proposed. The core forecasting mechanism of the suggested model is the DNN. Using clustering algorithms, the daily weather profile's features are extracted. The year-round daily weather profiles are grouped into several groups using k-means clustering, which extracts representative features from each group. After that, one DNN sub-model's ideal architecture and weighting variables can be found using the datasets in each cluster. In terms of tracking the peak, valley, and abrupt variations in energy consumption, the suggested FE-GA-DNN model performs better than the reference

GA-DNN predictive model with fixed architecture.

Enes Ayan (32) suggested a novel hyperparameter optimization technique based on evolutionary algorithms for CNN models that have already been trained to classify insect pest types. Three CNN models of varying scales (MobileNetV2, DenseNet121, and InceptionResNetV2) were used to test the proposed technique on three insect datasets: Wu's IP102 dataset with 102 classes, Xie2's D0 dataset with 40 classes, and Deng's dataset with 10 classes. On the D0 (99.89 percent) and Deng (97.58 percent) datasets, the optimized CNN models have attained state-of-the-art accuracies, and on the IP102 (71.84 percent) dataset, they have demonstrated the performance that is closest to the literature. The test findings indicate that the suggested approach successfully categorizes different crop pests and can be applied in agriculture to preserve crop fields.

2.1.5 Neural Architecture Search

Yu Xue et al. (33) suggested a self-adaptive mutation neural architecture search algorithm built using Dense Net and ResNet blocks of information. In order to accomplish better exploration, the self-adaptive mutation method allows the algorithm to adaptively alter the mutation techniques during the evolution process. Additionally, users don't need to be experts in CNN architecture design because the entire search process is totally automated. This work presents a comparison between the proposed algorithm and 17 cutting-edge algorithms, including CNN that is manually built and automatic search algorithms on CIFAR10 and CIFAR100. As per the findings, the suggested method exhibits superior classification performance compared to its competitors while utilizing fewer computational resources.

Franchini Giorgia et al. (34) examined how to select the hyperparameters pertaining to the optimization algorithm used to train the CNN, including the step length, the mini-batch size, and the possible use of variance reduction techniques. The architecture of a Convolutional Neural Network (CNN) includes factors like the number of filters and kernel size at each convolutional layer. The paper's primary contribution is the introduction of an automatic machine learning technique for optimizing a CNN performance metric by setting these hyperparameters. Specifically, we suggest a low-cost method to forecast the associated CNN's performance based on its behavior after just a few training steps, given a set of values for the hyperparameters. In order to accomplish this, we create a dataset whose label is the performance corresponding to a full training of the CNN, and whose input samples are supplied by a restricted number of hyperparameter configurations together with the corresponding CNN measures of performance obtained with only a few steps of the CNN training process.

Yuqiao Liu et al. (35) reviews over 200 articles of most recent EC-based NAS methods in light of the core components, to systematically discuss their design principles and justifications on the design. They introduced ENAS from four aspects: population representation, encoding space, population updating, and fitness evaluation. The findings indicate that the outcome aligns with the baseline (e.g., random search), suggesting that there's no need for intricate evolutionary operator designs. The crossover operator is a Mult individual based operator and there is no sufficient explanation to how the crossover operator works well on ENAS.

Edicson Santiago Bonilla Diazn et al. (36) addressed the bearing fault classification by evaluating three neural network models: 1D Convolutional Neural Network (1D-CNN), CNN-Visual Geometry Group (CNN-VGG), and Long Short-Term Memory (LSTM). Utilizing vibration data, our approach incorporates data augmentation to address the limited availability of fault class data. A significant aspect of our methodology is the application of neural architecture search (NAS), which automates the evolution of network architectures, including hyperparameter tuning, significantly enhancing model training. The comparison of results demonstrated that while other studies have achieved high accuracies, the methods and models proposed in this study displayed excellent results in more complex classification scenarios. This suggests that the combination of data augmentation, machine learning models, and NAS optimization can offer more reliable and high-performing solutions for bearing fault classification using vibration data.

Muhammad Junaid Ali et al. (37) suggested using an evolutionary Neural Architecture Search (NAS) method to find robust designs for the classification of medical images. One search algorithm that is employed is the Differential Evolution (DE) algorithm. Moreover, we employ an attention-based search space with sixteen convolution and pooling operations and five distinct attention layers. Tests conducted on many MedMNIST datasets demonstrate that the suggested method outperformed robust NAS and deep learning architectures in terms of performance.

Jeon-Seong Kang et al. (38) intended to give a summary of the fundamental ideas behind NAS as well as a rundown of current research on the technology's uses. It is important to note that the majority of earlier survey research on NAS has been on viewpoints related to hardware or search tactics. To the best of the current authors' knowledge, no previous study has examined NAS via the lens of computer vision. The computer vision domains were task-categorized in the current study, and each study's recent trends on NAS were thoroughly examined.

Lianbo Ma et al. (39) proposed an efficient fuzzy NAS framework for defect recognition, where the searched architecture can effectively handle uncertain information from the given datasets. Specifically, we first design a fuzzy search space and the related encoding strategy for fuzzy NAS. Then, we propose a comparator-based evolutionary search approach, where an online end-to-end comparator is learned to directly determine the selection of candidate architectures from the evolutionary population. The comparator works in an end-to-end way and it transforms the complex ranking problem of evaluating architectures into a simple classification task, which overcomes the rank disorder issue suffered from traditional performance predictors. A series of experimental results demonstrate that the architecture with fewer Params (1.22 M) search by fuzzy neural architecture search framework for defect recognition method achieves higher accuracy (92.26%) compared to the state-of-the-art results (i.e., DARTS-PV) on the ELPV dataset, as well as competitive results (accuracy = 76.4%, Params = 1.04 M) on the CODEBRIM dataset. Experimental results show the effectiveness and efficiency of our proposed method in handling uncertain problems.

Najmeh Fayyazifar (40) make use of Electrocardiogram (ECG) data in order to create an automatic method for detecting AF. For this purpose, we employed a neural architecture search (NAS) algorithm. The efficiency of NAS algorithms on image classification tasks has been well established, however, studies on using NAS methods for ECG classification are very limited. Our experiments show that our automatically designed neural model performs very well and arguably outperforms currently available deep learning models. This model achieved the accuracy and F1-score of 84.15%± 0.6 and 82.45± 0.2 on the publicly available subset of PhysioNet challenge 2017 dataset, respectively.

2.2 Summary

Many research works are already existing in the field of HAR which tried out various possible ML, Ensemble and DL models and each of the research work favoured different models. All the research works, concluded that some particular models work better than others when implemented on some standard datasets. The different datasets available in the field of HAR are UCI-HAR, WISDM, MHEALTH, DEVICE MOTION dataset, KU-HAR, AMU-HAR etc. Significant progress has been made in HAR research recently by utilizing ML and DL approaches. Research results show that DL techniques perform better than conventional ML techniques, especially when it comes to accuracy and performance. All these models frequently include attention mechanisms and meta-heuristic algorithms for hyper-parameter optimization. Additionally, it has been demonstrated that combining optimization methods with neural networks, such as Adam and RMSProp, improves model performance. Novel architectures such as 1D CNN s, fusion techniques, and self-supervised learning frameworks have also helped to advance HAR systems, demonstrating a wide variety of strategies targeted at efficiently using sensor data for activity recognition.

2.3 Research Gaps

1. Choice of hyper-parameters for evaluation and tuning.
    a. It is possible that the results obtained could change if a different number or different types of hyper parameters are tuned.
    b. In building a CNN model, the change of parameters like number of pooling layers, kernel size, number of filters, learning rate and others can have effect on the perfor- mance of the model.
    c. To overcome the choice of parameter selection, we have done hyper-parameter tuning using GA.
2. For different ML, Ensemble and DL models we will get different accuracy and exe- cution times and finding the optimal among all the models is important.
    a. Among all the existing research works, every work proposed a different architecture and there is no such architecture which is suitable for all the other datasets.
    b. Problem of raw-data was not considered in any of the research works.
3. We tried to propose a better architecture which is suitable for most of the datasets in the field of HAR.
    a. This research work is also an attempt to tackle the challenge of raw data. We collected raw sensor data and tried to get the best model for that data as well.

Chapter 3

DATA SET REQUIREMENTS

3.1 Data Set

3.1.1 UCI HAR Dataset

UCI HAR DATA SET (Unsupervised and unstructured) -KAGGLE. Sensor data collected from device -Samsung Galaxy S2 smart phone (Accelerometer and gyroscope) The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. The six activities performed were as follows:

1.Walking

2.Walking Upstairs

3.Walking Downstairs

4.Sitting

5.Standing

6.Laying

Accelerometer – rate of change of velocity with respect to time. A gyroscope measures angular velocity, which is the rate of rotation around a particular axis. It provides information about the device's orientation and changes in orientation over time.

Dataset is mainly depended on the 9 inertial signals obtained from accelerometer and gyroscope.

1.Body Acceleration data:

Includes only linear acceleration due to the device movement

1.body_acc_x_test

2.body_acc_y_test

3.body_acc_z_test

2.Body Gyroscope Data:

1.body_gyro_x_test

2.body_gyro_y_test

3.body_gyro_z_test

3.Total Acceleration data:

It include both linear acceleration and the gravitational acceleration.

1.total_acc_x_test

2.total_acc_y_test

3.total_acc_z_test

X-axis: Changes in acceleration along this axis may correspond to movements in the horizontal plane, such as forward or backward movements (e.g., tilting the device forward or backward).

Y-axis: Changes in acceleration along this axis may correspond to vertical movements, such as lifting the device up or lowering it down.

Z-axis: Changes in acceleration along this axis may correspond to movements in the depth or radial direction, such as shaking the device.

The dataset also includes a very well-engineered set of 561 features calculated from 128 reading of each channel. Our training set consists of 7352 sample and we used 20

3.1.2 Collected Dataset

Device -IQOO neo 7 smart phone (Accelerometer and Gyroscope) The experiments have been carried out with a group of 16 volunteers. 12000 values from 16 individuals each individual 750 values for all 6 activities each activity 125 values for every individual.

The sensor data is recorded by the phone which is positioned near the waist. The Acc x_axis, Acc y_axis, Acc z_axis; Gyr x_axis, Gyr y_axis, Gyr z_axis are the features and the data is collected for the activities. The six activities performed were as follows:

1.Walking

2.Walking Upstairs

3.Walking Downstairs

4.Sitting

5.Standing

6.Laying

3.2 Machine Configuration

In this Experimental study, the software and the hardware systems used are

Software Requirements

Python 3.12.1 is used as programming language and TensorFlow 2.15.0 as the DL framework. To evaluate the time complexity, the models were tested on a 64-bit Windows 10 operating system. Google Colab is used for the code implementation in T4 environment.

Hardware Requirements

Intel Core i5 processor and 8 GB RAM is used.

Chapter 4

PROPOSED SYSTEM

The entire project is divided into 3 phases, phase 1 is experimental comparison, phase 2 is hyperparameter tuning of the best model obtained from phase 1 and phase 3 is doing a neural architecture search to find a model that performs well across various datasets.

Phase -1

An experimental comparison of ML models like Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), K Nearest Neighbour (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Ensemble (hard voting) and DL models like CNN, RNN are implemented on a standard pre-processed dataset (UCI-HAR). Thereafter, a complete analysis of the performance of all the implemented models is done The best model is determined by taking into account two metrics: execution time and accuracy.

Phase -2

1D-CNN is considered as the best model, the model's parameters are all taken into account, adjusted, and then evaluated to see if the model performs well in every scenario then EA is used to fine-tune the best parameters for the model and the performance is evaluated. By further exploring the results, the change of architecture of the model is considered as an important parameter.

Phase -3

The EA is used to find the best architecture (Neural Architecture Search) of the model which showed an improvement in the evaluation metrics. Also, there is a possibility that the model might not be suitable for raw data. Hence, the sensor data has been collected from smart- phone (Accelerometer and Gyroscope) while performing the activities and made into a dataset then a Neural architecture search (NAS) is implemented to find a model that works best with raw data. Both the models are tested on the datasets like WISDM, PAMAP and HHAR and the results have been evaluated.

4.1 Novelty of the solution

This research on HAR, includes the implementation of all the models that are worked with, finding the best performing model from them using few metrics like no. of features, accuracy and time. The model features are taken and are passed through the Optimization Algorithm (GA) (Fine-tuning the best performing model) to improve the accuracy of the model. There were different datasets tried in those works and each of them has used a different model with different architecture and that might not be the best for all other datasets of HAR. The dataset chosen in this project is UCI-HAR. Hence, we have done a neural architecture search using GA, which will help in finding the best possible architecture along with the best parameters. The resulting best model can be used for many other datasets with the same architecture. Although we get the best model with best architecture, still there is a chance that the model may not give good results with raw data. Thus, we have collected real-time raw sensor data from smart-phones and tried to find the best model with best possible architecture using neural architecture search, so that any randomly given un-preprocessed data is expected to provide better performance.

4.2 Work Flow of the project

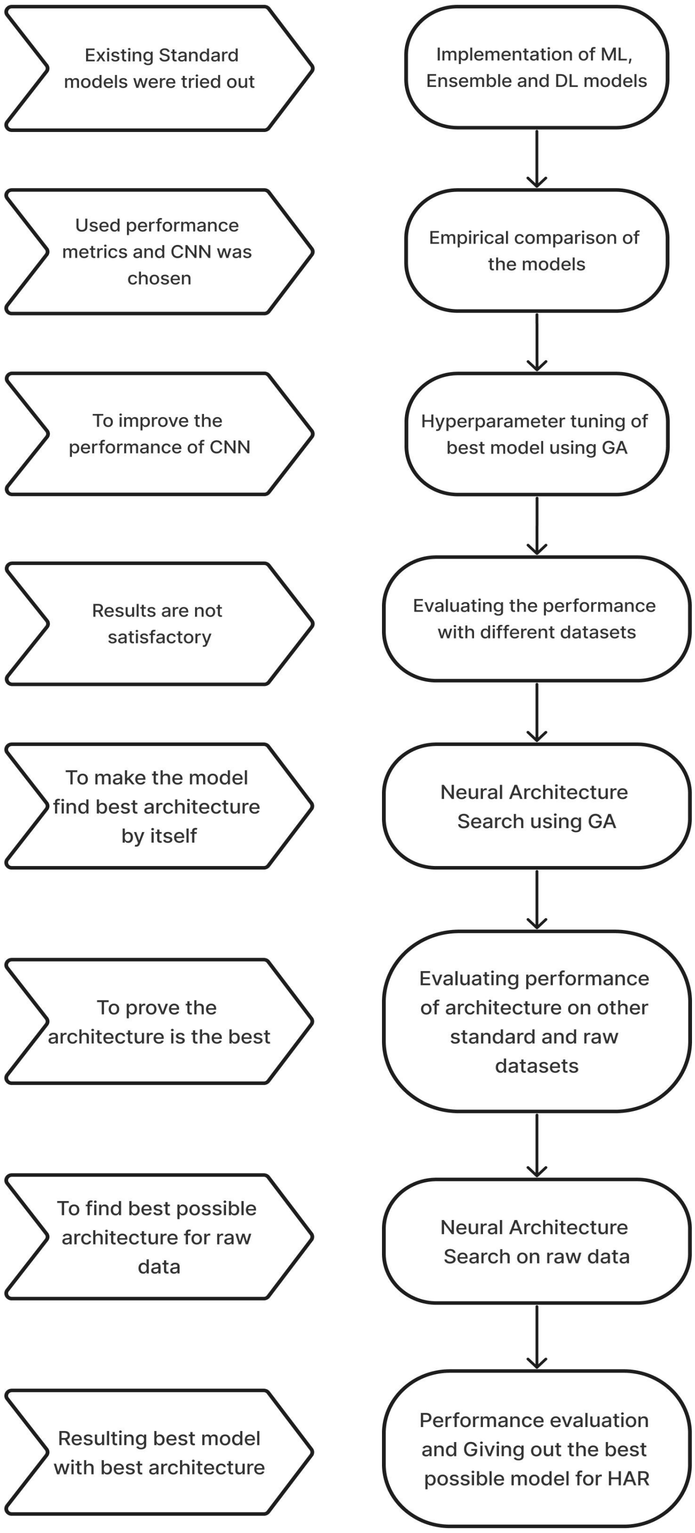

Figure 4.1: Work Flow

Chapter 5

IMPLEMENTATION

5.1 Empirical comparison of ML, Ensemble and DL models

5.1.1 Machine Learning Models

Under the ML module the algorithms considered were LR, DT, RF, NB, SVM, Ensemble learning. The models used in ML module are used with different setups such as like Considering all features, considering important features, Top 20 features and Dimensionality Reduction using Principle Component Analysis (PCA).

Logistic Regression: it is used for the classification of 6 activities using the 'one vs. rest' technique. A standard scalar is used to scale the feature to avoid under fitting and over fitting. A logistic regression model is initialized with 1000 iterations to ensure convergence. The top 20 most significant features based on the absolute values of coefficients are identified and the training and test datasets are reduced to only include these top 20 features.

Decision Tree: it is initialized with a set random state for reproducibility. The classifier is trained on the training dataset. In Decision Tree with selected top 20 features, Feature importance is calculated based on how effectively each feature splits the data to classify the target variable. PCA is applied to both the training and test datasets with the goal of retaining 95% variance in the data.

Random Forest: A random Forest with 100 trees is initialized, using a fixed random state for reproducibility. The top 20 features are selected based on their importance scores. A new random forest classifier is trained on the reduced feature set. A random forest classifier is trained on the PCA-transformed training data.

K-Nearest Neighbor: A KNN classifier is initialized with n neighbors=5, indicating that the classifier considers the 5 nearest neighbors for making predictions. KNN model is initialized, and GridSearchCV is used to find the optimal number of neighbors. The search range is set from 1 to 30.

Gaussian Naive Bayes: The Gaussian Naive Bayes classifier is initialized and trained on the training dataset. This model assumes that the features follow a normal distribution. Gaussian Naive Bayes assumes that the features follow a normal distribution.

Support Vector Machine: An SVM classifier with the RBF kernel is initialized. The random state=42 ensures the reproducibility of results. SVM is a powerful classification model, capable of handling both linear and non-linear data.

5.1.2 Ensemble Learning Model

An ensemble learning model is created using the following ML models LR, DT, SVM, and KNN classifiers is used. A Voting Classifier is created, comprising the four individual classifiers. As 'hard' is the voting parameter set, the final prediction is determined by the classifiers' majority vote. Predictions are made on the test dataset and the accuracy of the ensemble model is calculated.

5.1.3 Deep Learning Models

CNN: 1D convolution layer with 64 FILTERS and a kernel size of 3 is used to extract features from the input data followed by a max pooling 1D layer to reduce the data's dimensionality and help prevent over fitting. The fully connected neural network is followed by a dense layer with 100 neurons and a RELU activation function.

RNN: Labels are converted to one-hot encoding required for multi-class classification. A Standard Scaler is applied to standardize features. A simple RNN layer with 50 units and activation. A dropout layer with a rate of 0.5 is added. A dense layer with units equal to the number of classes and a SoftMax activation function is used for the output layer.

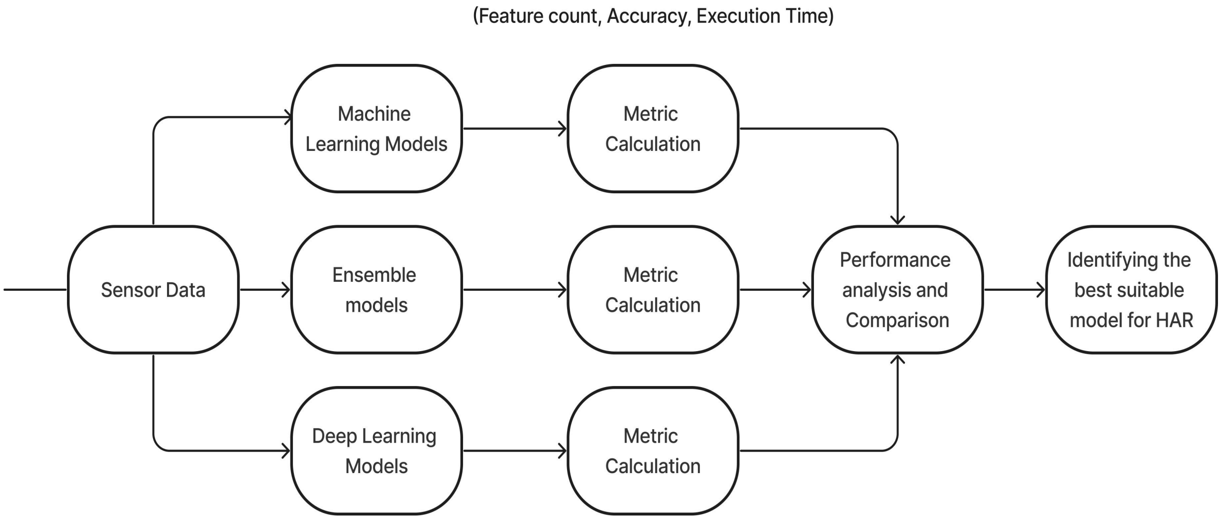

Figure 5.1: Work-flow of phase 1

5.2 Basic CNN architecture

5.2.1 CNN

Input Layer: Serves as the entry point for the 1D CNN, accepting the multi-channel sensor data input and passing it to subsequent layers for feature extraction and learning.
Convolutional Layer: Applies convolution operations to detect and learn hierarchical features in the sequential sensor data, enabling the model to capture patterns relevant to human activity recognition.
Flatten Layer: Reshapes the output from the convolutional layer into a one-dimensional vector, preparing it for input into subsequent fully connected layers for classification.
Dense Dropout Layer: Adds a layer of regularization by randomly dropping out a fraction of neurons during training, preventing overfitting and enhancing the model's generalization ability.
Normalized Layer: Normalizes the output values, typically using techniques like batch normalization, ensuring stable and efficient training by maintaining a consistent scale across the neural network's layers.
Filters = 64, Kernel size = 3, Pool size = 2, Activation function = relu

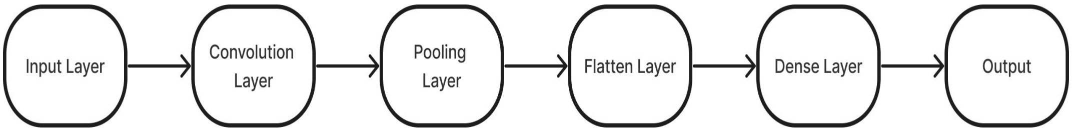

Figure 5.2: Basic CNN Architecture

5.3 Hyper Parameter Tuning using GA

5.3.1 Genetic Algorithm

Genetic Algorithm is used to find the best possible parameter for CNN architecture. Genetic Algorithm is a search and optimization technique to find the approximate solution for complex optimization and search problems using principles of natural selection and genetics. For our proposed method, parameters like, Number of CNN layers, Number of filters in each unit, activation function in each unit, Inclusion or exclusion of Pooling layers and dropout layers, loss function and optimizer.

The main steps involved in genetic algorithm is:

Initialization: Generate a population with potential CNN models that solves our problem. Here CNN models can be generated by randomly selecting the above-mentioned parameters.

Fitness Evaluation function: This function uses the accuracy of test data as a fitness function and can try to maximize the value or can try to lose value of the test data as fitness values and try to minimize its value. A 5 folds cross validation is used to find the optimal split size, epochs and any regularization values.

Selection: It is the process of selecting individual CNN models from the population. The probability of selection would be directly proportional to its fitness value, so better CNN models have better chances of getting selected to make a new generation of population.

Reproduction: The two CNN models from the selected population are crossed. Here the cross involves selecting each parameter with 40 percent probability from each of the two CNN models and 20 percent probability of randomly selecting a value for the parameter. The randomness helps to avoid converging to local minimum i.e., sub optimal solutions. Then a 5-fold cross validation is used to just iterate within two standard deviations from the mean to find best epochs, split size ratio and regularization parameters.

Replacement: Replace 80 percent of the old population with the new generation. This Could be better than the entire replacement so that the best model is among the top 10 to 20 percent of the old population.

Termination: Here termination of the process will be based on the resource available, like time and processing power to find the best model within 5 to 10 generations and stop the process if very little or no improvement in the best model of the population and store the best model in each step and return that model at the end of the process

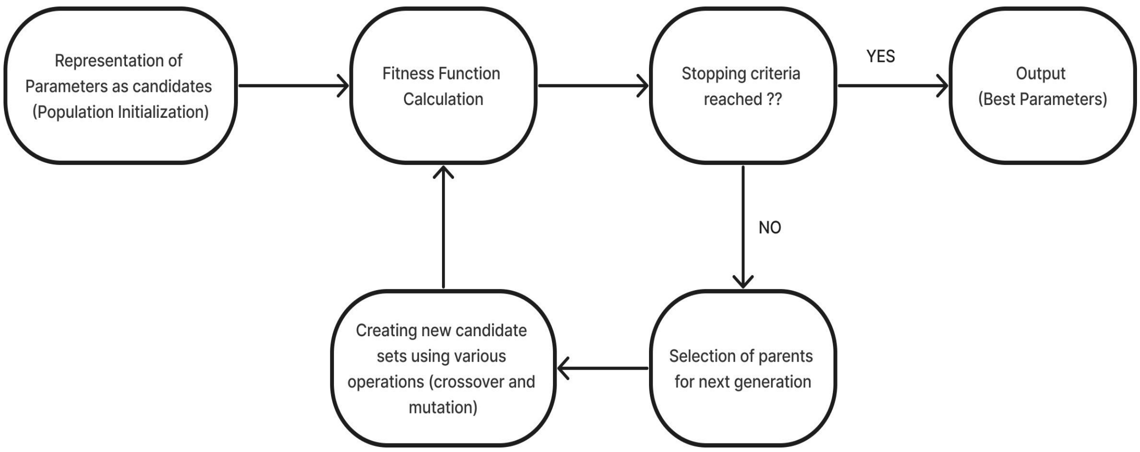

Figure 5.3: GA Architecture

5.4 Modified CNN Structure with fixed parameters

In the initial Basic CNN structure, there was no consideration of parameters and only one convolutional layer taken. But to verify the performance of the model when given multiple convolutional layers and some set of parameters, the model was given two parallel convolutional layers followed by one pooling layer each, then merge layer to merge both the layers. Here max pooling has been used. Following the merge layer another convolutional layer along with pooling is given and then flatten layer, dense layer respectively and finally the best configuration with the fixed set of parameters and a randomly chosen architecture comes out from output layer. The list of parameters considered are :

Filters = [32, 64, 128]

Kernel size = [3, 5, 7]

Pool size = [2, 3, 5]

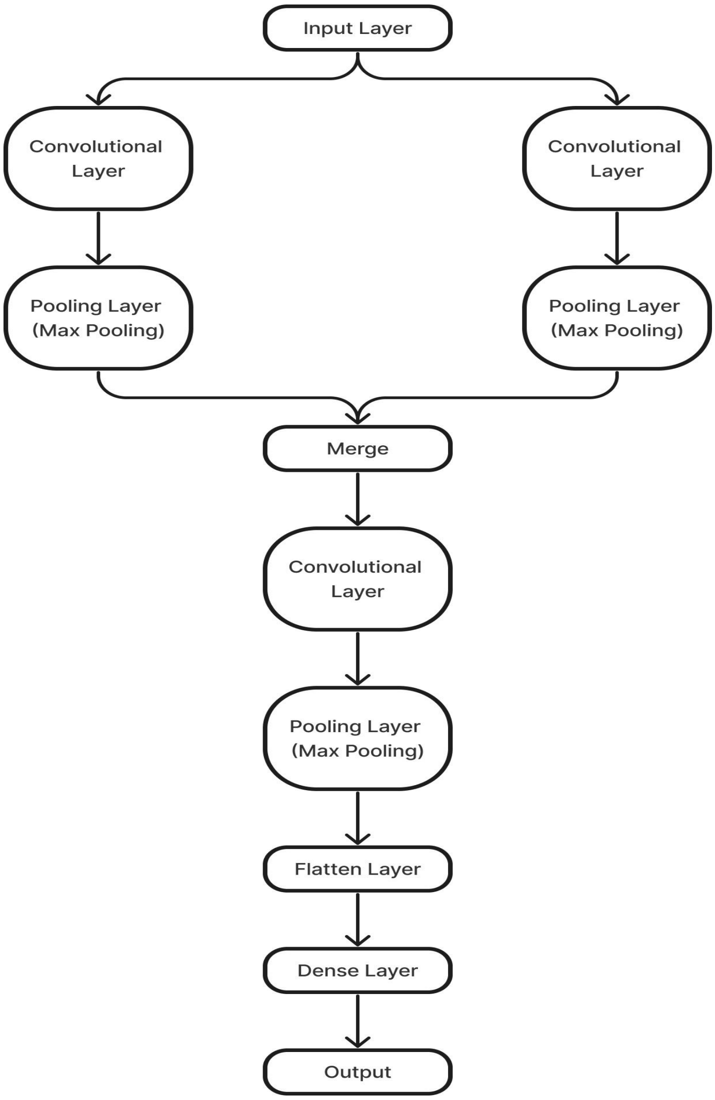

Figure 5.4: Modified CNN Architecture

5.5 Neural Architecture Search

The previous model may show good performance, but the chosen architecture might not be the best one. Hence, in this project a neural architecture search is tried out on the chosen dataset(UCI-HAR) using GA to find the best possible architecture with best parameters. The above structure shows resulted best architecture after NAS. It contains 3 parallel convolutional layers after the input layer, one pooling layer following each convolutional layer. Then merge layer combines all the layers and gives out the result to convolutional layer after that pooling layer, similarly 3 series of convolutional layers and pooling layers consecutively and then flatten layer, dense layer respectively and finally from output layer the best configuration with the best architecture for CNN model along with the possible best parameters which will give better performance than others will be obtained.

List of parameters considered :

convl_layer = [1,2,3], convl_third = [1,2,3]

Filters = [32, 64, 128], Kernel size = [3, 5, 7]

Pool size = [2, 3, 5], activation_func = [sigmoid, relu, elu]

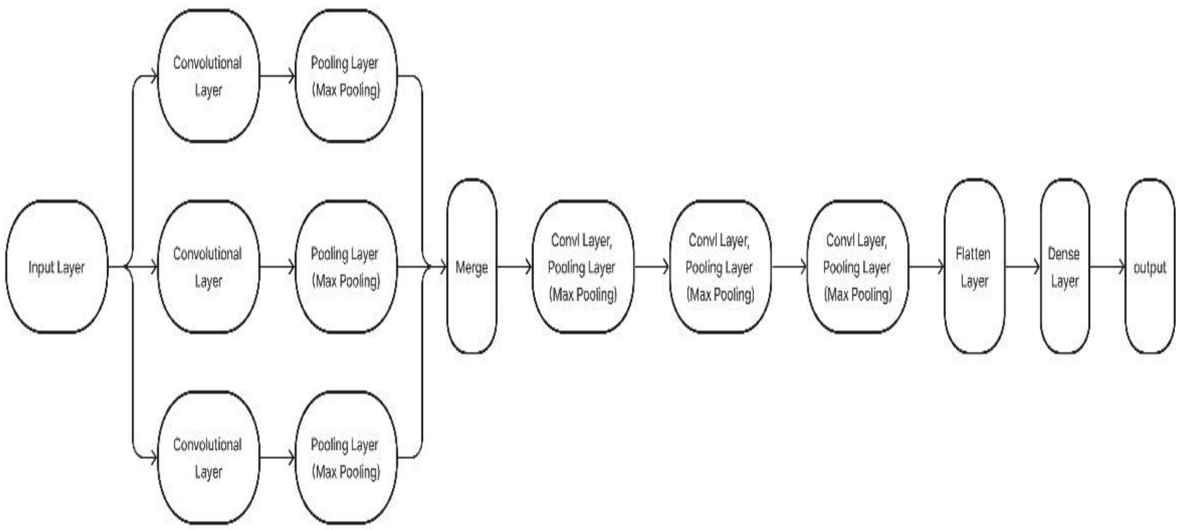

Figure 5.5: CNN model obtained from NAS

5.6 Verifying the best model using custom dataset

Custom dataset: for the purpose of testing the model on raw data, a huge effort has been put to collect the real time raw sensor data from 16 individuals, each performing all the 6 activities. 125 values are collected for each activity from every individual valuing upto a total of 750 values were collected from each individual which resulted in 12000 values, included in our new dataset. The readings are collected using an application called phyphox, through a smart-phone.

Activities: Walking, Walking Upstairs, Walking Downstairs, Standing, Sitting, Laying

The obtained values are accelerometer and gyroscope readings: acc x_axis, acc y_axis, acc z_axis, gyr x_axis, gyr y_axis, gyr z_axis

Then, the dataset was given to multiple models like LR, DT, RF, CNN (Resulted best architecture) to verify the performance of all models when given raw data. As there is limitation to LR that it converts the problem into a supervised learning task, where the input is the set of features extracted from each window and the output is the activity label. Now in the case of LR it treats each input feature independently and does not account for the sequence or order of data points. This is a significant limitation for HAR data, where the sequence of sensor readings is crucial for recognizing activities. Also, LR will not perform well on RAW-DATA.

5.7 Implementing the model on different Data sets

For the proof of validation that the architecture came out of NAS is best, in this project some more standard datasets are considered, and all the models implemented on UCI-HAR dataset are again implemented on the chosen datasets. For comparing their performance, accuracy is considered as main metric Complete result analysis and observations are shown in chapter 6

Datasets considered are: WISDM, PAMAP, HHAR

Models implemented are: LR, DT, RF, CNN (with best configuration obtained)

Chapter 6

RESULTS AND DISCUSSION

This research mainly focuses on finding out the best model for HAR and enhance the model using Evolutionary algorithms to improve the accuracy and adaptability in recognizing the activities. In the initial phase traditional machine learning and deep learning models are explored to find out the best model for HAR performance.This exploration includes LR,DT,RF,SVM,KNN,NB in Machine learning, RNN and 1D CNN in deep learning models.To validate the models performances they were tested across multiple well known datasets,each presenting unique challenges and characteristics. HHAR dataset that contains data from smart phones and smart watches sensors ,WISDM dataset data collected from smartphone sensors during everyday activities. PAMAP dataset contains more physically intense activities recorded using wereable sensors.Raw sensor dataset contains unprocessed data in a less controlled environment.

6.1 A detailed work of the result of UCI HAR data

This comprehensive study evaluates various models for HAR based on features, accuracy, and processing time. Notably, Logistic Regression with 310 important features achieved high accuracy (95.72 percent) in 62.95 seconds. Decision Tree and Random Forest, employing feature importance and top features, demonstrated competitive performance. KNN with optimal parameters (n=8) reached 90.74 percent accuracy in 2.7887 seconds. SVM achieved 95.045 percent accuracy in 5.633 seconds. Ensemble methods and CNN proved effective, yielding 94.57 percent and 95.21 percent accuracy, respectively. The study provides a comprehensive overview, enabling the selection of models based on specific criteria in HAR applications.

S.No.	Model	Features	Accuracy (%)	Time (s)
1	Logistic Regression (LR)	561	95.62	109.24
	LR with top 20 features	20	77.02	48.51
	LR considering all important Features	310	95.72	62.95
	LR + Dimensionality reduction using PCA	67	94.13	7.96
2	Decision Tree (DT)	561	86.22	5.41
	DT top 20 features	20	83.30	6.33
	DT considering all important features	9	81.40	7.17
	DT + Dimensionality reduction using PCA	65	79.84	5.78
3	Random Forest (RF)	561	92.56	16.18
	RF with important features	16	79.57	18.21
	RF with top 20 features	20	82.05	33.05
	RF + Dimensionality reduction using PCA	67	90.53	13.35
4	K-Nearest Neighbor (KNN) (n=5)	561	90.15	1.6569
	KNN (n=8) optimal	561	90.74	2.7887
	KNN + Dimensionality reduction using PCA(n=19 optimal)	67	90.39	44.28
5	Naive Bayes (NB)	561	77.02	19.7
	NB + Dimensionality reduction using PCA	67	87.68	1.86
6	Support Vector Machine (SVM)	561	95.045	5.633
	SVM + Dimensionality Reduction using PCA	67	94.06	4.732
7	Ensemble Method	561	94.57	44.36
8	RNN	561	19.03	14.30
9	CNN	561	95.21	123.38

Table 6.1: Comparison of the results of all the implemented models

6.1.1 Rate of Miss classifications in each model

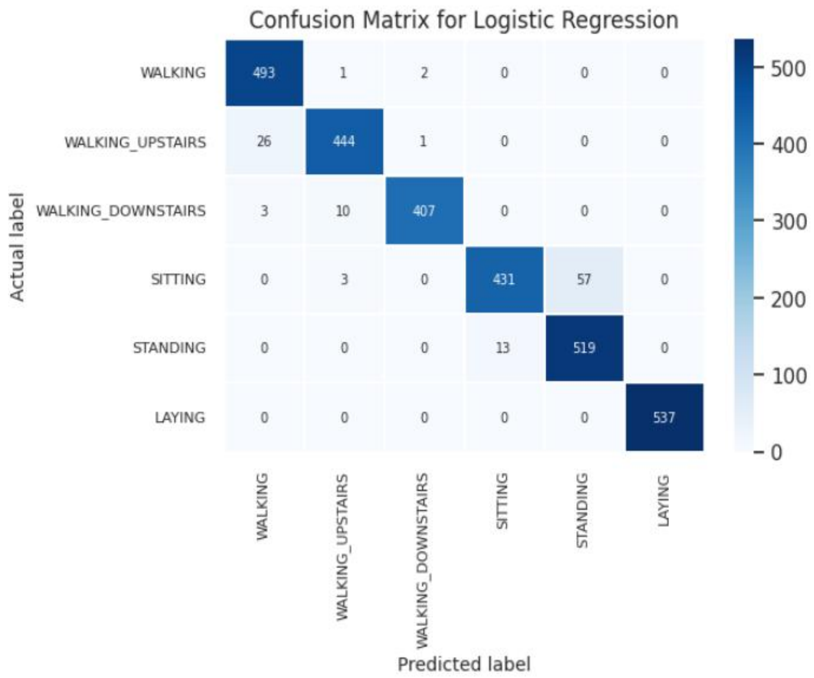

Figure 6.1: Confusion matrix for Logistic Regression

Confusion Matrix : (Logistic Regression)

WALKING was almost perfectly classified with only 1 misclassification as WALKING_UPSTAIRS and 2 as WALKING_DOWNSTAIRS.

WALKING_UPSTAIRS was misclassified as WALKING 26 times and as WALKING_DOWNSTAIRS once.

WALKING_DOWNSTAIRS was confused with WALKING 3 times and WALKING_UPSTAIRS 10 times.

SITTING was misclassified as STANDING 57 times.

STANDING was misclassified as SITTING 13 times.

LAYING was consistently correctly classified with no misclassifications.

The classifier performs exceptionally well in distinguishing 'LAYING' from other activities with perfect classification.

There is some confusion between 'SITTING' and 'STANDING', which could be due to similarities in the static nature of these activities.

The classifier also shows a high degree of accuracy in distinguishing between the walking activities but with some errors, mainly between 'WALKING_UPSTAIRS' and 'WALKING'.

Decision Tree :

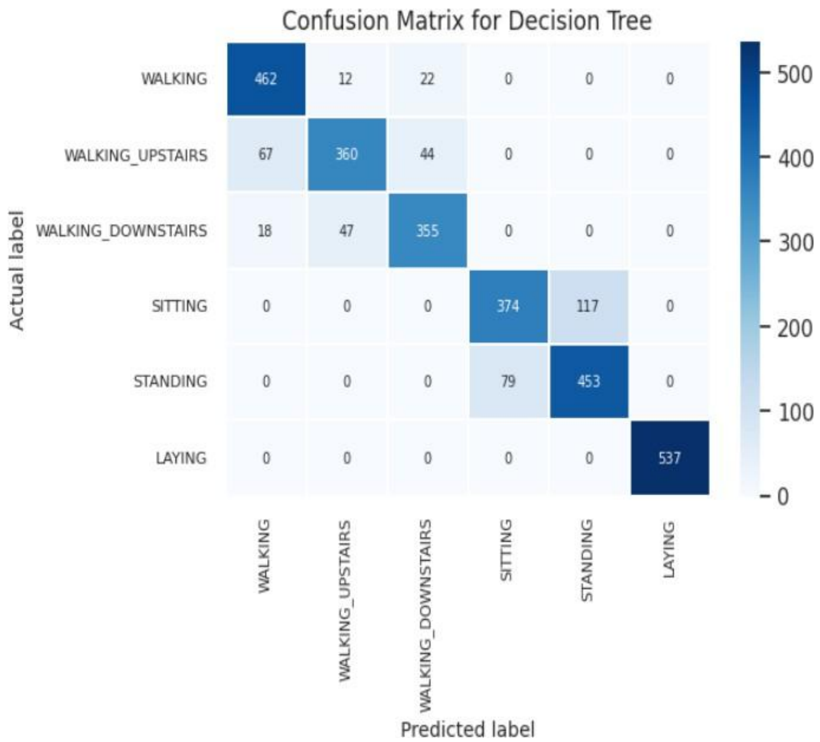

Figure 6.2: Confusion matrix for Decision Tree

The decision tree model in highly accurate in predicting laying activity.In static activity the model misclassified more in sitting activities to standing activities.

WALKING with WALKING_UPSTAIRS (29 times) and WALKING_DOWNSTAIRS (21 times).

WALKING_UPSTAIRS with WALKING (61 times) and WALKING_DOWNSTAIRS (46 times).

WALKING_DOWNSTAIRS with WALKING(17 times) and WALKING_UPSTAIRS (44 times).

SITTING with STANDING (97 times).

STANDING with SITTING (79 times).

Random Forest :

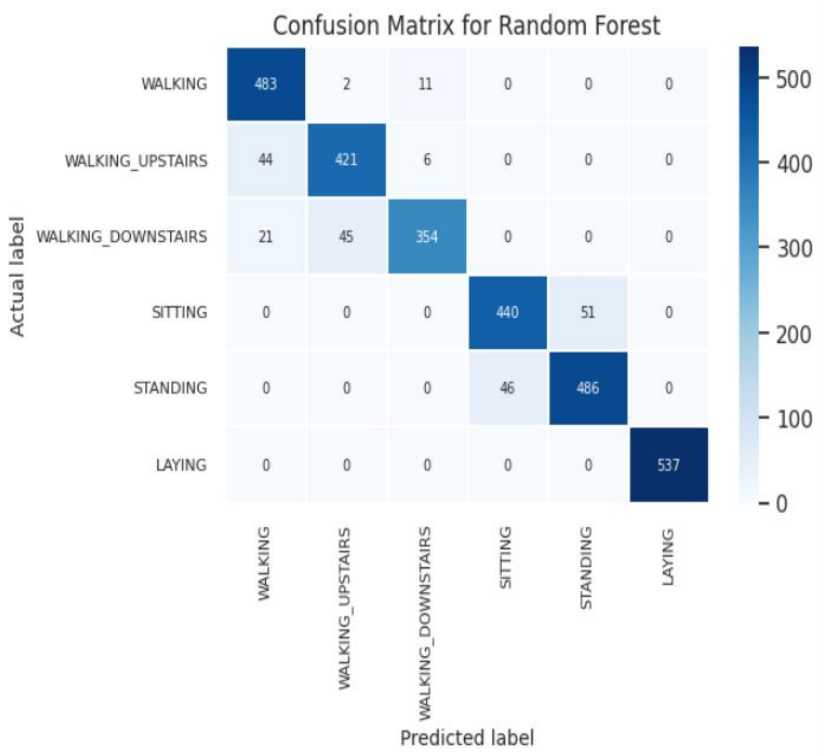

Figure 6.3: Confusion matrix for Random Forest

The random forest model is highly accurate in predicting laying The classifier had some difficulty distinguishing between:

WALKING and WALKING_UPSTAIRS (3 instances) and WALKING_DOWNSTAIRS (10 instances).

WALKING_UPSTAIRS was occasionally misclassified as WALKING (32 times) and WALKING_DOWNSTAIRS (7 times).

WALKING_DOWNSTAIRS was sometimes confused with WALKING (15 times) and WALKING_UPSTAIRS (40 times).

SITTING and STANDING were confused 56 and 45 times, respectively

Support Vector Machine :

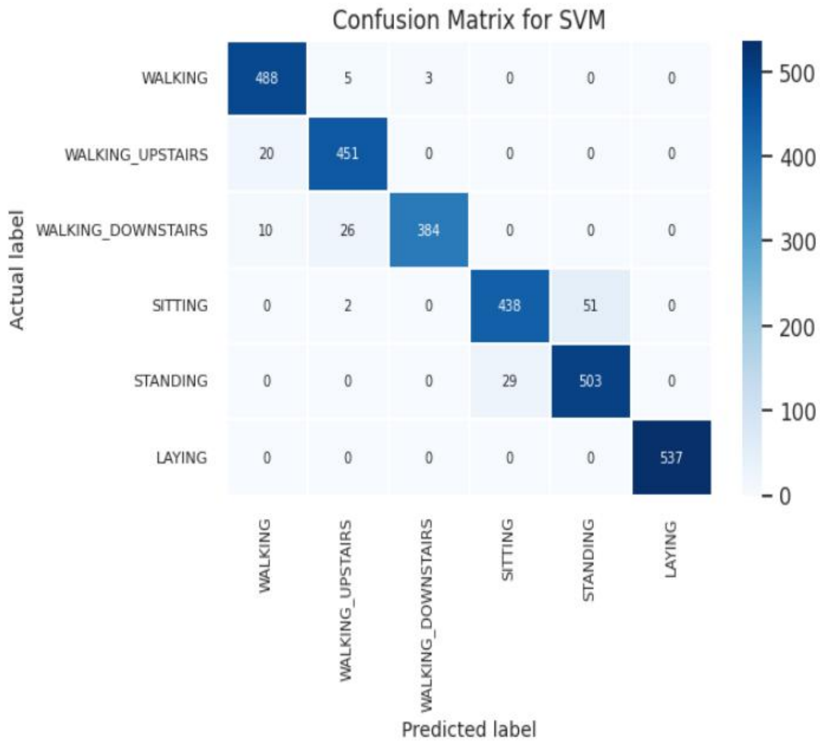

Figure 6.4: Confusion matrix for Support Vector Machine

The classifier has some difficulty distinguishing between:

WALKING and WALKING_UPSTAIRS (7 instances) and WALKING_DOWNSTAIRS (3 instances).

WALKING_UPSTAIRS is misclassified as WALKING (21 times).

WALKING_DOWNSTAIRS is confused with WALKING (8 times) and WALKING_UPSTAIRS (24 times).

SITTING is misclassified as STANDING (46 times).

STANDING is confused with SITTING(31 times).

K - Nearest Neighbour :

The classifier had some difficulty distinguishing between:

WALKING and WALKING_UPSTAIRS (3 instances) and WALKING_DOWNSTAIRS (13 instances).

WALKING_UPSTAIRS was sometimes confused with WALKING (28 instances) and WALKING_DOWNSTAIRS (25 instances).

WALKING_DOWNSTAIRS was misclassified as WALKING (30 instances) and WALKING_UPSTAIRS (40 instances).

SITTING was misclassified as STANDING (78 times).

STANDING was incorrectly classified as SITTING(57 times)

LAYING was almost always correctly classified with only 3 instances of misclassification.

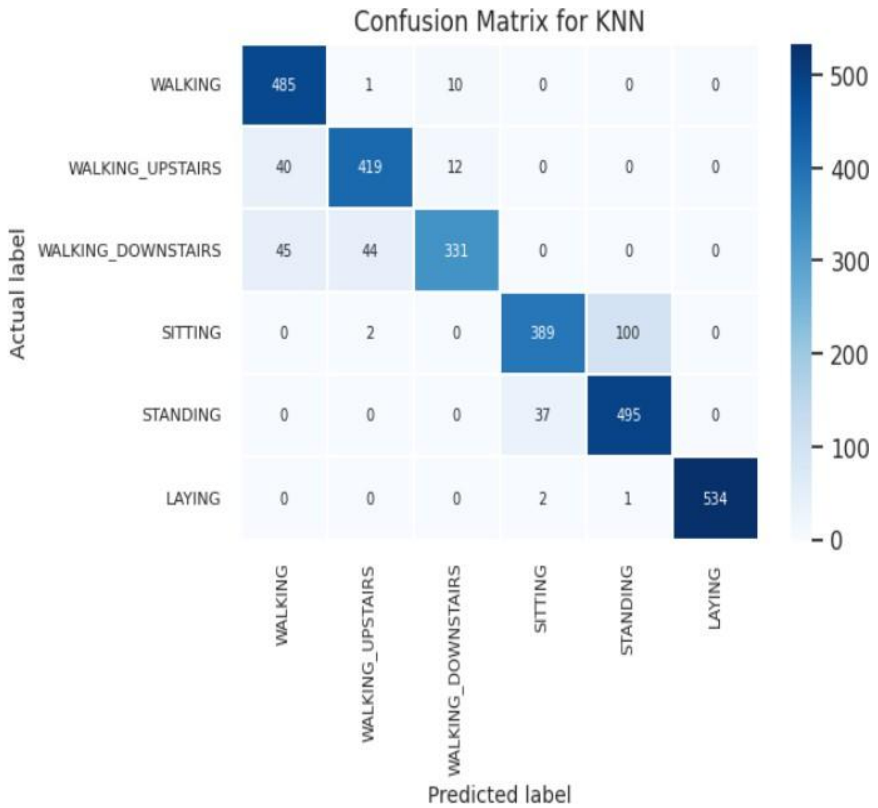

Figure 6.5: Confusion matrix for K-Nearest Neighbour

Naive Bayes :

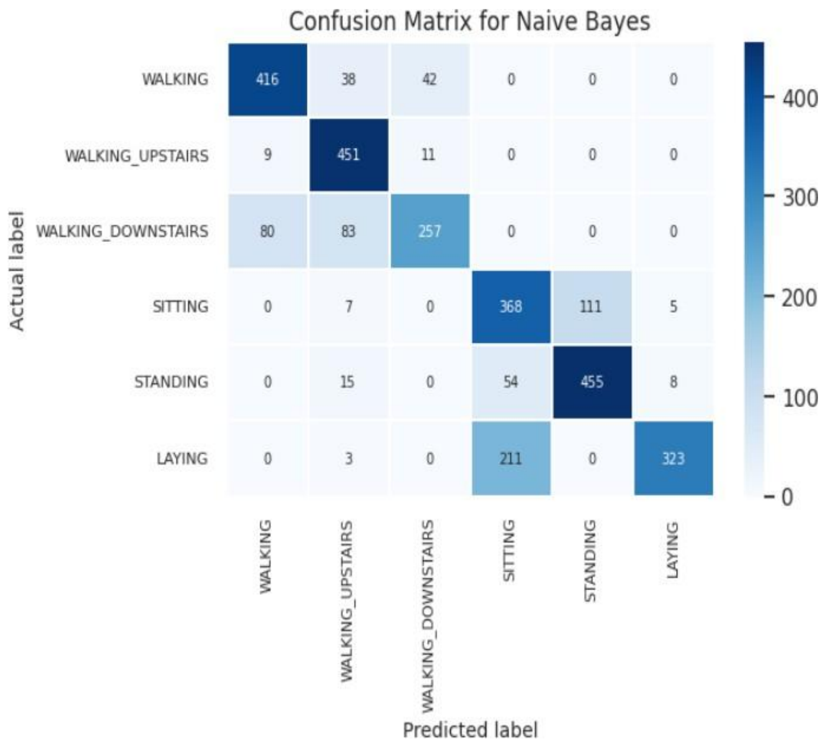

Figure 6.6: Confusion matrix for Naive Bayes

WALKING was often confused with WALKING_UPSTAIRS(40 times) and WALKING_DOWNSTAIRS (41 times).

WALKING_UPSTAIRS was mistaken for WALKING(9 times) and WALKING_DOWNSTAIRS(12 times).

WALKING_DOWNSTAIRS was misclassified as WALKING (77 times) and WALKING_UPSTAIRS (84 times).

SITTING was incorrectly predicted as STANDING(61 times).

STANDING was misclassified as SITTING(162 times) and LAYING (8 times).

LAYING was confused with SITTING 207 times.

There is notable confusion between 'SITTING' and 'STANDING' as well as between 'STANDING' and 'LAYING.The activities involving motion ('WALKING', 'WALK- ING_UPSTAIRS', 'WALKING_DOWNSTAIRS') are sometimes confused with one another, which may indicate similar feature patterns for these activities.

Comparison of Miss-classification rate across best models

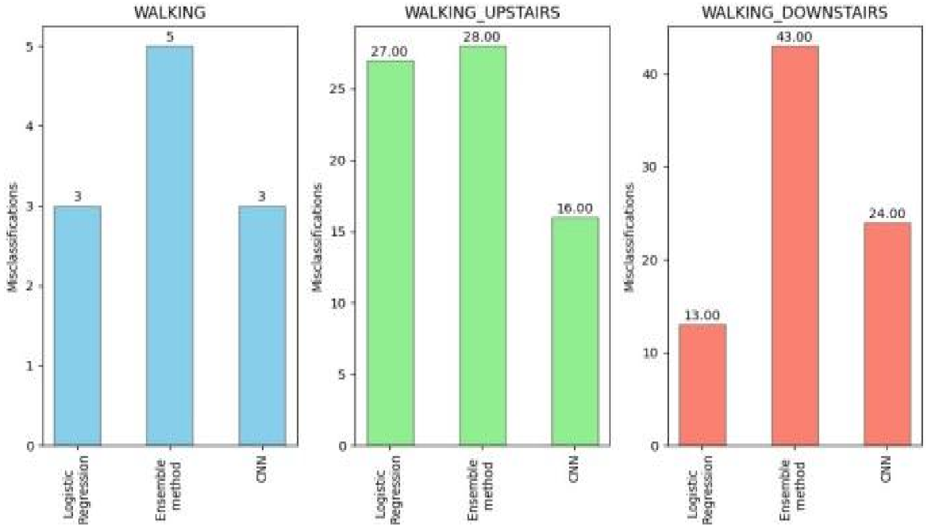

Figure 6.7: misclassification in Dynamic Activities

Out of ML models, DL models and Ensemble groups, the best model was chosen based on highest accuracy: LR under ML, Ensemble, CNN under DL. The misclassification rate is calculated based on how many activities were wrongly classified from the actual one in the test data, the models are chosen based on their accuracy from their base models.

Activities	Logistic Regression (%)	Ensemble Model (%)	CNN (%)
Walking	0.60	1.00	0.60
Walking Down Stairs	3.09	10.48	5.71
Walking Up Stairs	5.6	5.9	3.39
Avg	3.09	5.79	3.23

Table 6.2: The misclassification rate for the Dynamic Activities

From the table, the ML models, LR have performed well and CNN has performed well in the DL part with both giving less misclassification and the overall average for the dynamic activities is 4.03%.

Activities	Logistic Regression (%)	Ensemble Model (%)	CNN (%)
Sitting	11.60	8.99	4.49
Standing	2.4	6.7	10.15
Laying	0	0	0
Avg	4.66	5.23	4.88

Table 6.3: The misclassification rate for the Static Activities

From the table, among the ML models, LR has performed well and CNN has performed well in the DL part with both giving less misclassification and the overall average for static activities is 4.92%.

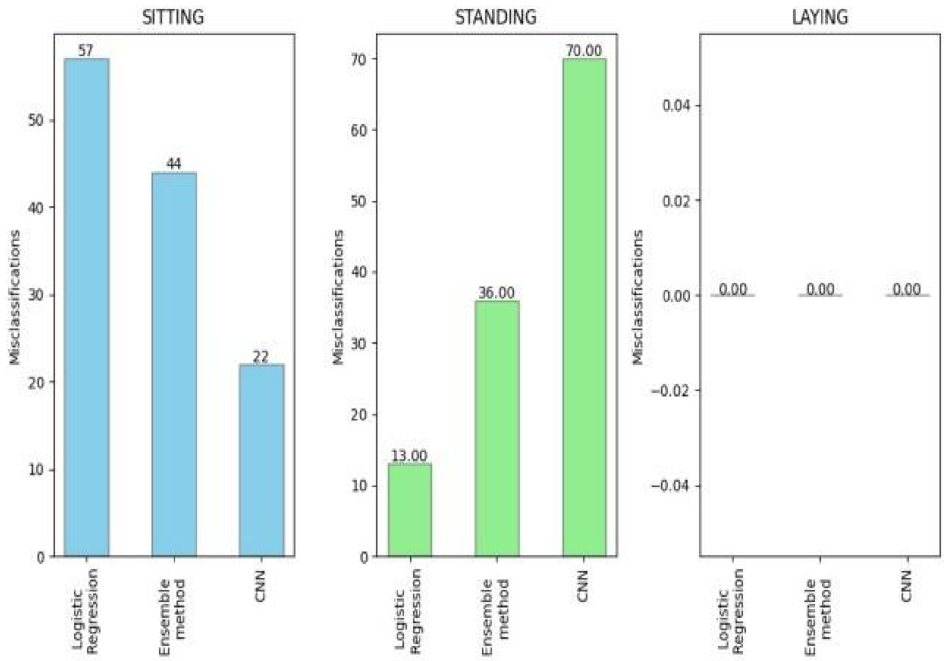

Figure 6.8: Misclassification in Static Activities

The dataset used in this project is already a preprocessed one where the continuous stream of data collected from the accelerometer and gyroscope are segmented into fixed size window. Each window is labeled with the activity that represents most of its duration. This process converts the problem into a supervised learning task. Now in the case of LR it treats each input feature independently and does not account for the sequence or order of data points. This is a significant limitation for HAR data, where the sequence of sensor readings is crucial for recognizing activities.

6.2 Result and Analysis of Hyper Parameter Tuning

6.2.1 Dataset performance on various models

Logistic regression:

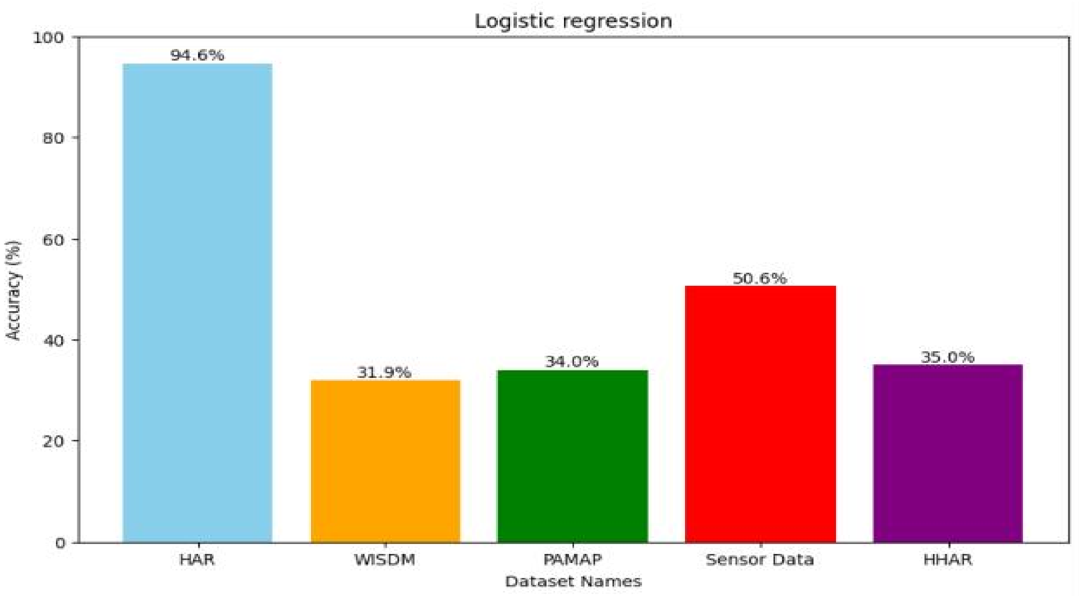

Figure 6.9: Performance of model in Logistic Regression

Logistic Regression is used for the classification of various activities in the above mentioned datasets.A standard Scaler is used to scale the feature to avoid underfitting and overfitting.A logisitic regression model is initialized with 1000 iterations to ensure convergence.Model performance is evaluated using accuracy.

The logistic regression model shows exceptional performance on HAR dataset with an accuracy of 94.6 percent and the performance significantly poor in case of WISDM,PAMAP ,HHAR and raw sensor data. It is a clear evident that logistic regression perfroms better on well segmented pre processed data.Where as WISDM,PAMAP and HHAR yielded poor result because of their complex acitivity patterns.From this its is concluded that Logistic regression might not be suitable for advanced HAR.

Decision Tree:

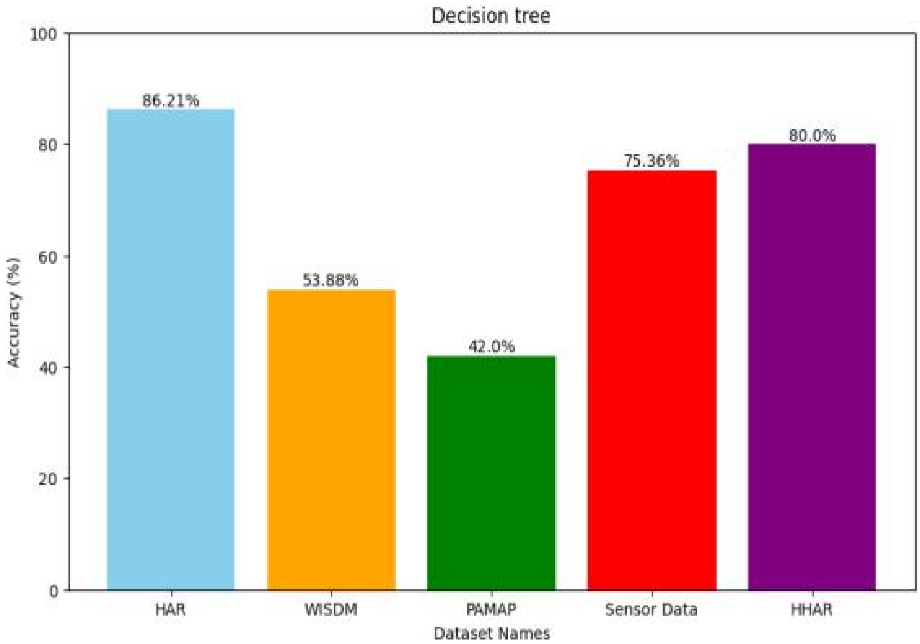

Figure 6.10: Performance of model in Decision Tree

A decision tree classifier is initialized with a set random state of 42 for reproducibility.In case of UCI HAR dataset the classifier is trained on the training dataset.The trained model is used to predict the outcome of the test dataset .In case of WISDM, HHAR, PAMAP ans Raw dataset the data set is splitted in 70:30 for train and test data and The accuracy of the model is calculated.

The decision tree model achieves its highest accuracy on the UCI HAR Dataset at 86.21 percent and HHAR at 80 percent accuracy.These results suggest that Decision tree model can effectively handle structured and semi structured data where the activitesare well designed and less complex.Since the PAMAP and WISDM data sets possibly have complex activity pattern poor performance of model is observed.

Random Forest:

A random Forest with 100 trees is initialized, using a fixed random state for reproducibility.In case of UCI HAR dataset the classifier is trained on the training dataset.The trained model is used to predict the outcome of the test dataset .In case of WISDM, HHAR, PAMAP ans Raw dataset the data set is splitted in 70:30 for train and test data.Predictions are made on the test dataset and the accuracy of the model is computed.

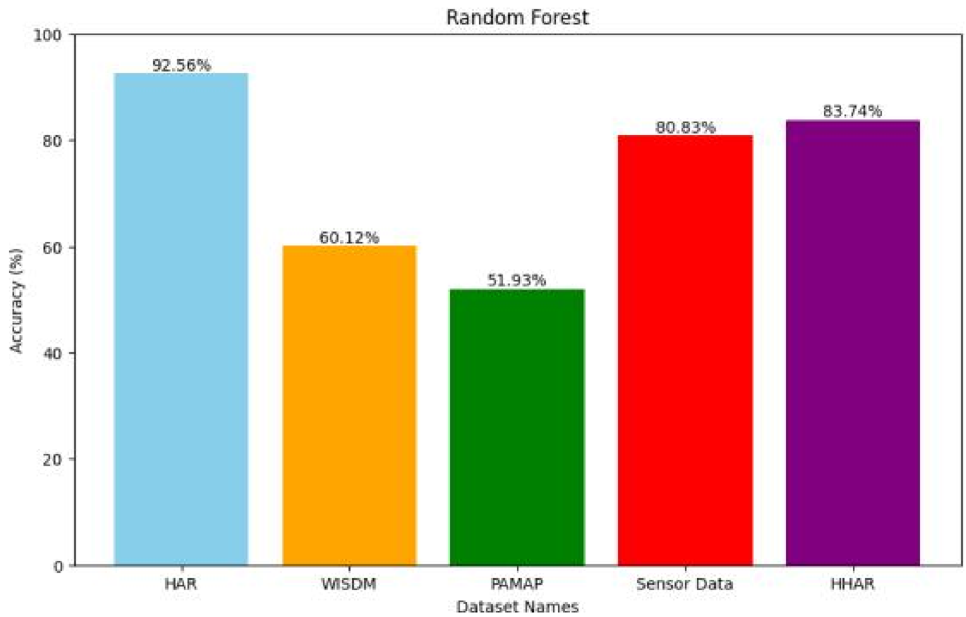

Figure 6.11: Performance of model in Random Forest

The random forest model performed well in UCI-HAR ,hhar and sensor data.These results suggest that the RF model is well at managing structured and clean datasets with less complex activities. WISDM and PAMAP having complex activities but its is observed that RF is comparatively good than LR And DT.

1D CNN:

1D convolution layer with 64 filters and a kernel size of 3 is used to extract features from the input data followed by a maxpooling 1D layer for reducing the dimensionality of the data and also helps in preventing from overfitting.The data is then flattened to feed into fully connected neural network.The fully connected neural network is followed by a dense layer with 100 neurons and a RELU activation providing the network with the ability to learn non linear relationships.The final layer is another dense layer with a softmax activation function.The model is compiled with the Adam optimizer and categorical cross entropy loss function.It is trained on the reshaped training data for 10 epochs with a batch size of 32 including validation data for performing evaluation.

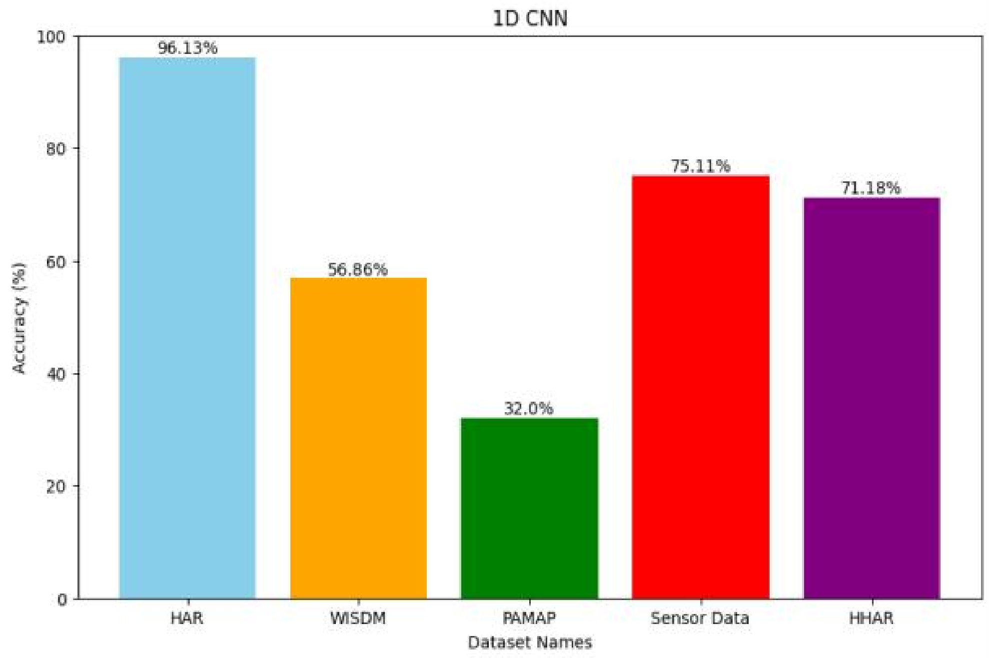

Figure 6.12: Performance of model in 1D CNN

1D CNN model performed well in case of UCI HAR with an accuracy of 96.13 percent. UCI HAR contains well structured and clean dataset ideal for cnn.The performance on raw sensor data and HHAR is also reasonable.The performance on WISDM and PAMAP can be altered by modifying the parameters of the CNN .

The analysis showed that traditional machine learning model RF performed well and 1D CNN model performed well in deep learning. Because of the CNN's ability to learn from large volumes of data and pattern recognition and ability to capture temporal dependencies.1D CNN is considered to be the best model.To work on the further enhancement of the perform of 1D CNN model UCI HAR data set Is considered for the further work.

6.2.2 Working of GA on basic CNN

To enhance the performance of 1D CNN model.Genetic algorithm is appointed to fine tune the parameters. The parameters includes number of filers,kernel size ,max pooling layer, activation functions,learning rate.A set of values of parameters are given to the GA and a tournament based selection approach is used to randomly select the parameters and given as an input to 1D CNN models and after going to certain iteration and performing some cross over and mutation by GA and performing multiple generation the best configurations are selected by GA .

```
Filters: 32, Kernel Size: 3, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 26.33 seconds
Filters: 32, Kernel Size: 3, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 41.98 seconds
Filters: 32, Kernel Size: 3, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 18.60 seconds
Filters: 32, Kernel Size: 5, Pool Size: 2, Accuracy: 0.9114353656768799, Time: 35.61 seconds
Filters: 32, Kernel Size: 5, Pool Size: 3, Accuracy: 0.41194435954093933, Time: 23.37 seconds
Filters: 32, Kernel Size: 5, Pool Size: 5, Accuracy: 0.8754665851593018, Time: 19.26 seconds
Filters: 32, Kernel Size: 7, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 28.03 seconds
Filters: 32, Kernel Size: 7, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 24.65 seconds
Filters: 32, Kernel Size: 7, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 21.51 seconds
Filters: 64, Kernel Size: 3, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 44.14 seconds
Filters: 64, Kernel Size: 3, Pool Size: 3, Accuracy: 0.8985409140586853, Time: 35.86 seconds
Filters: 64, Kernel Size: 3, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 42.10 seconds
Filters: 64, Kernel Size: 5, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 46.99 seconds
Filters: 64, Kernel Size: 5, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 38.45 seconds
Filters: 64, Kernel Size: 5, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 42.18 seconds
Filters: 64, Kernel Size: 7, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 47.04 seconds
Filters: 64, Kernel Size: 7, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 39.60 seconds
Filters: 64, Kernel Size: 7, Pool Size: 5, Accuracy: 0.8554462194442749, Time: 33.08 seconds
Filters: 128, Kernel Size: 3, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 84.39 seconds
Filters: 128, Kernel Size: 3, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 101.05 seconds
Filters: 128, Kernel Size: 3, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 53.04 seconds
Filters: 128, Kernel Size: 5, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 89.88 seconds
Filters: 128, Kernel Size: 5, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 71.64 seconds
Filters: 128, Kernel Size: 5, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 57.49 seconds
Filters: 128, Kernel Size: 7, Pool Size: 2, Accuracy: 0.18221920728683472, Time: 119.89 seconds
Filters: 128, Kernel Size: 7, Pool Size: 3, Accuracy: 0.18221920728683472, Time: 82.14 seconds
Filters: 128, Kernel Size: 7, Pool Size: 5, Accuracy: 0.18221920728683472, Time: 59.36 seconds
Best configuration: Filters: 32, Kernel Size: 5, Pool Size: 2, Accuracy: 0.9114353656768799, Time:
35.61 seconds
```

Figure 6.13: Output of GA on basic CNN

6.2.3 Neural Architecture Search

A neural architectural search is also appointed to methodically search for optimal network architectures and parameters.NAS also utilized a GA with a tournament selection method to evolve not only the hyperparameters but also the architectural elements of the network.GA determines the optimal number of convolutional layers that should run in parallel and the number of layers that work sequential after the merge.To explore the potential of various architectures the GA was executed over three generation each with a population size of 15.Each generation was evaluated based on performance metrics,with the best performing architectures carried forward to subsequent generations for mutations and crossover.This GA approach enhances network configuration and also provides the best parameters.

Each individual configuration varied in terms of the number of convolutional layers,filter sizes,pool sizes and activation functions.These parameters impacts the model ability ti extract and learn features from the input data.

Analysis of Gen-1 NAS

The GA is used to explore various configurations focusing on structure of convolutional layers and parameters and model was evaluated to find out the impact of the parameters on model performance and best model is opted for crossover and mutation in evolving the network

Accuracy plot of Gen1:

There is a wide range of values of accuracy ranging from 17 percent to 98 percent.Individuals 10 and 13 achieved the highest accuracy, suggesting that their configurations were suitable.From the Generation 1 result it is evident that varied filter sizes are being used from 16,32 and 64, kernel size of 3 and 5 are used and RELU and ELU are suitable activation functions.

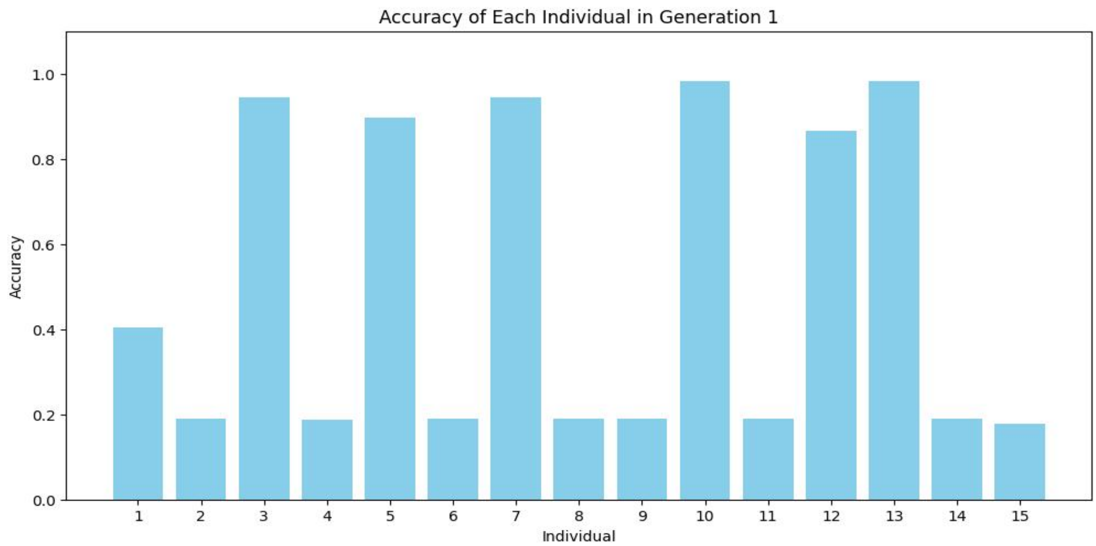

Figure 6.14: Accuracy plot of all 15 individuals

Convolutional layers:

Configurations are ranged from single layer setup to more more complex structures with multiple layers both in parallel branches and sequential flows.More layers are showing better performance and might have ability to capture more complex features.

Individual 10 and 13 achieved highest accuracy with configuration of 2 parallel ,3 sequential layers and 3 parallel and 3 sequential layers with a combination of Elu and Relu activation functions.Individual with low accuracy are using sigmoid activation function so even though they have more layers there performance is low from this we can evident that ELU and RELU are the best choice.

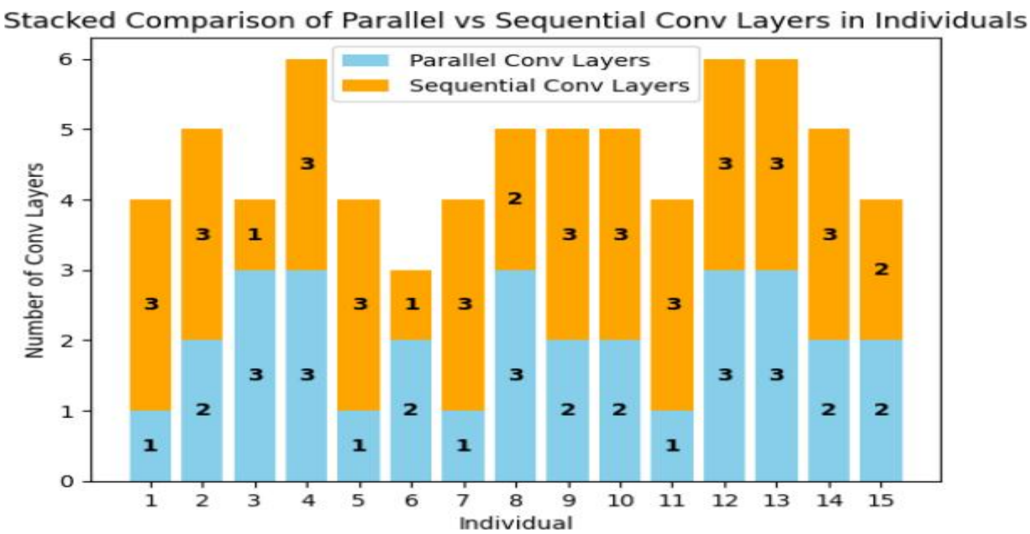

Figure 6.15: Number of parallel and sequential layers

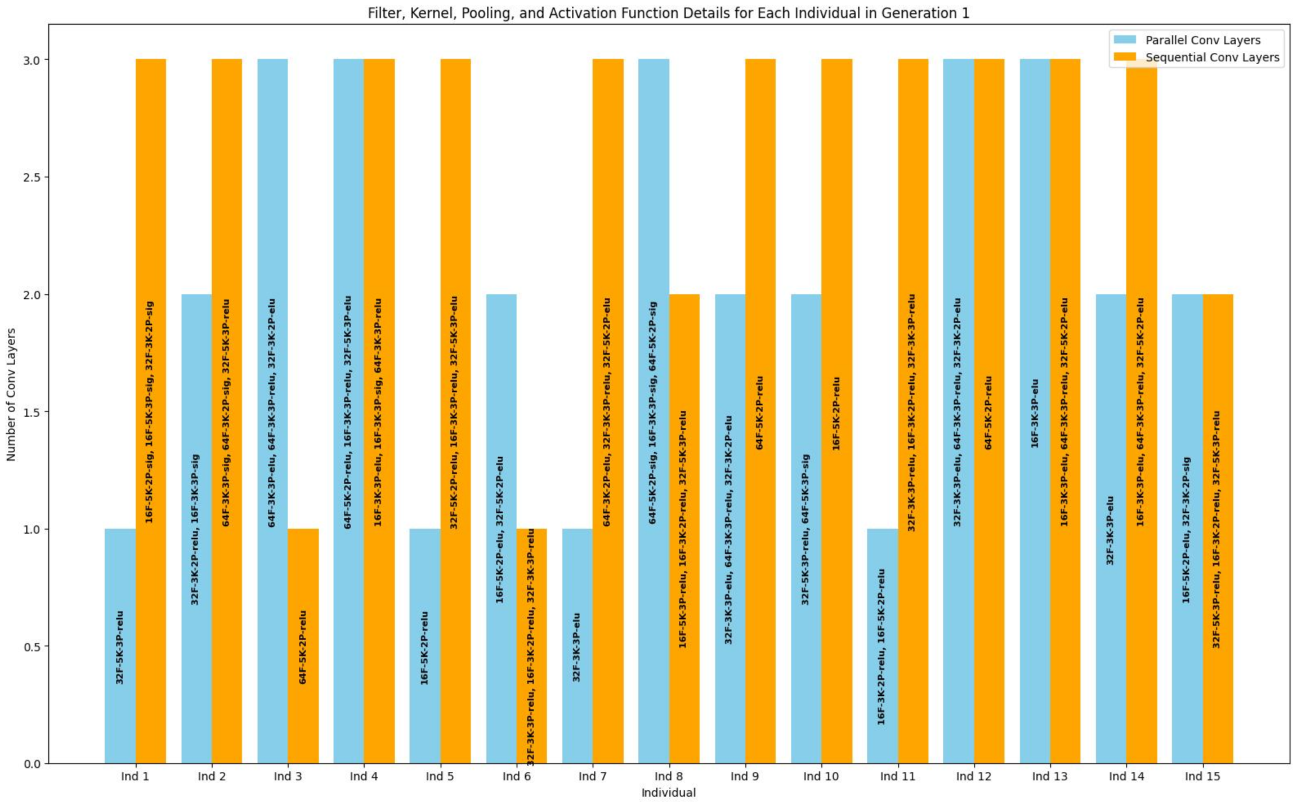

Figure 6.16: List of all the parameters used in each layer

From the obtained output,the best configurations are identified based on their accuracy scores.The highest accuracy is 98.30 percent founded in individual 10 and 13.The genetic algorithm employs two main operations to explore the configuration space crossover and mutation.These mechanism helps in finding optimal solutions over gen- erations by combining successful configuration and introducing variations to explore more possibilities.The crossover involve swapping layers ,changing the sequence of lay- ers,mixing parameters like filter size, activation function ,max pooling ,epochs etc,,.Mutation randomly exchanges the configuration within the population prevents the code from fix- ated on local optimum.

From the obtained population result in generation 1 the model performed 5 crossover and 4 time mutation.

Analysis of Gen-2 NAS

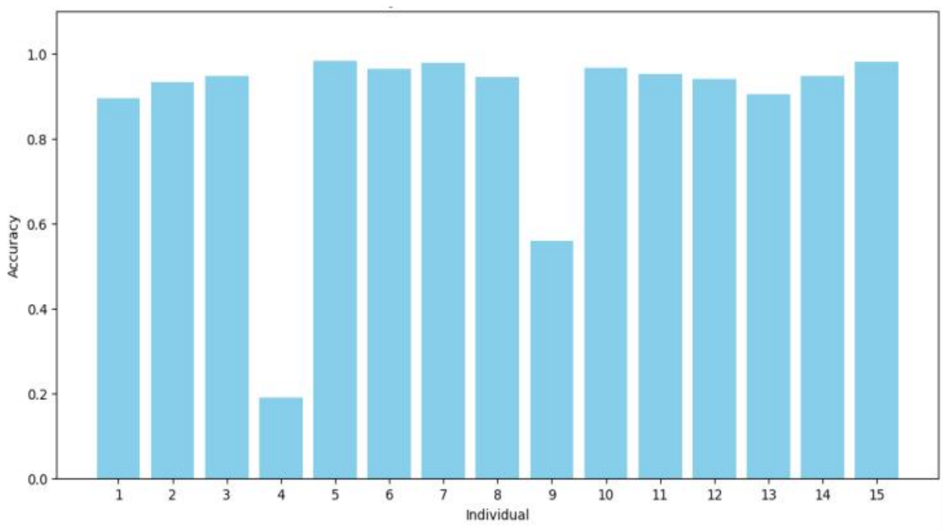

Figure 6.17: Accuracy plot of all 15 individuals

There is a slight variation in the accuracy's ranging from 17 percent to 98 per- cent.Individuals 5 and 15 achieved the highest accuracy and the individuals giving least accuracy are re- duced suggesting that their configurations were suitable.From the Generation 2 result it is evident that filter size of 32 and 64 are used more, kernel size of 5 is often used RELU and ELU are suitable activation functions.

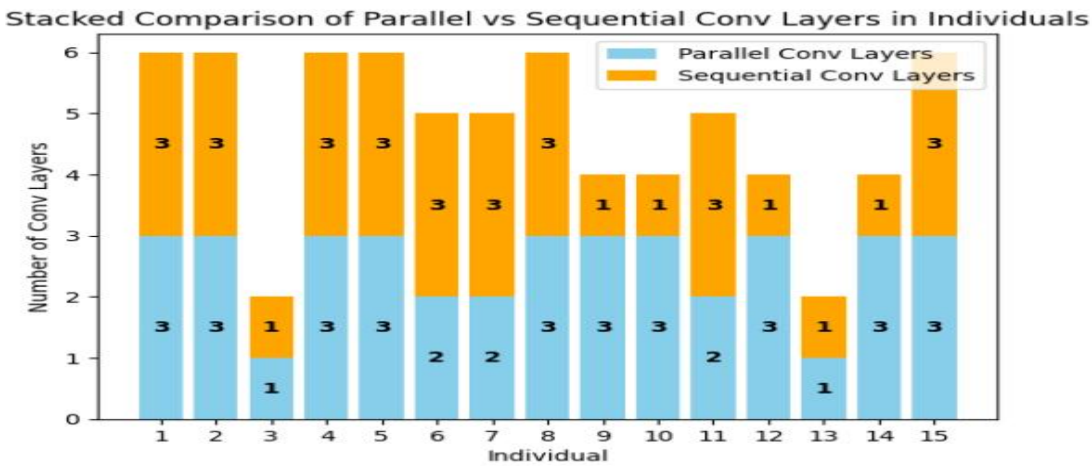

Figure 6.18: Number of parallel and sequential layers

Comparing to Gen 1 the model with 3 parallel and 3 sequential is performing better by giving better accuracy. The model with 1 parallel and 1 sequential is giving less accuracy and the usage of that model is reduced in Gen 2.

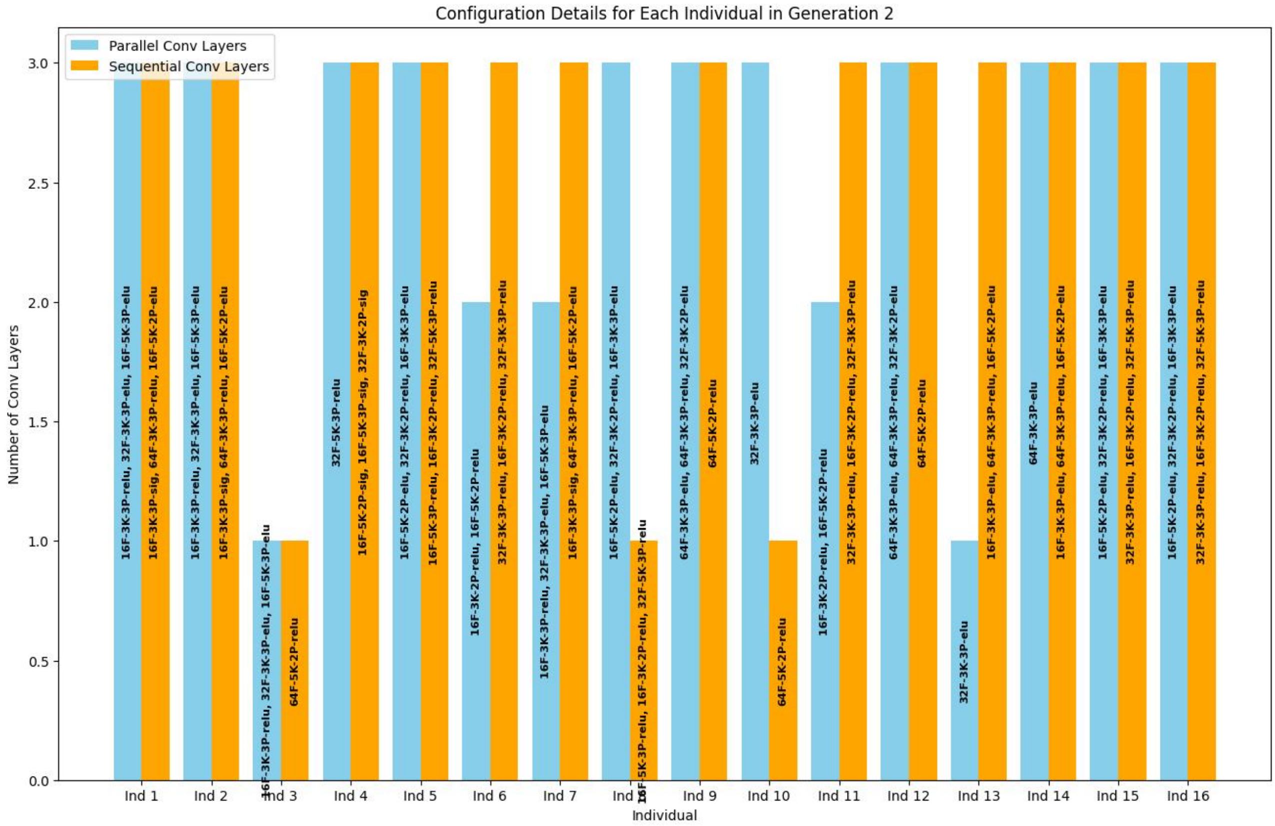

Figure 6.19: List of all the parameters used in each layer

Individual 5 and 15 achieved highest accuracy with configuration of 3 parallel ,3 sequential layers and 3 parallel and 3 sequential layers with a combination of Elu and Relu activation functions.Individual with low accuracy are using 1 parallel and 1 sequential layers so even though they are using ELU and RELU functions it concludes that by having more layers there performance is improved.

Analysis of Gen-3 NAS

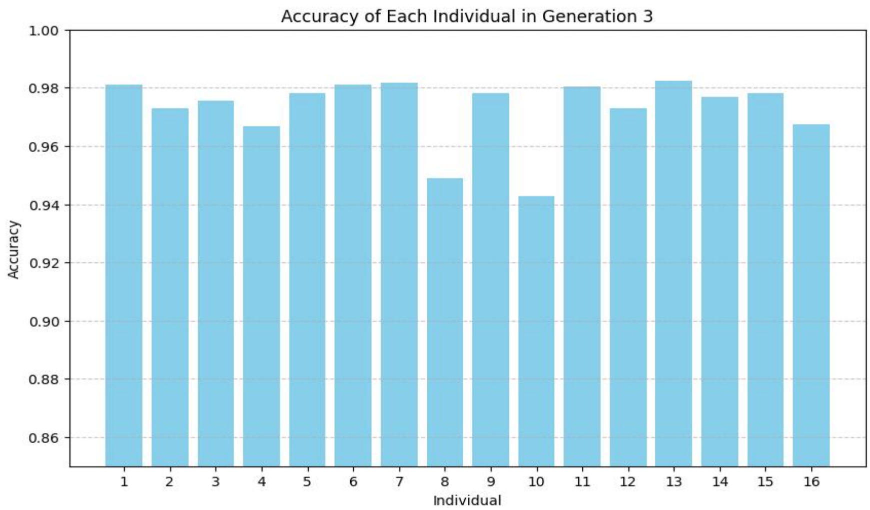

Figure 6.20: Accuracy plot of all 15 individuals

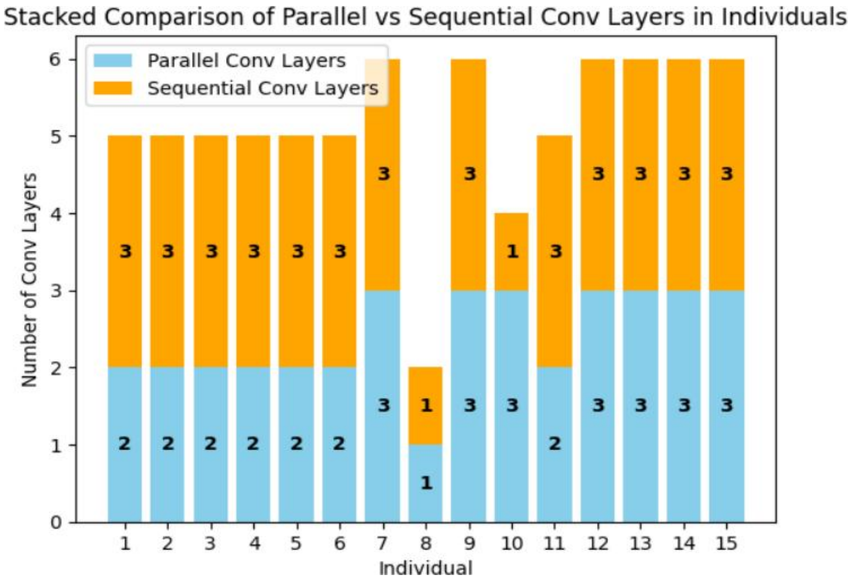

Figure 6.21: Number of parallel and sequential layers

The individuals with 3 parallel and 3 sequential layers are giving the best accuracy while individuals with variable number of layers are giving less accuracy. The individual with 1 parallel and 1 sequential layers are giving the far less accuracy.

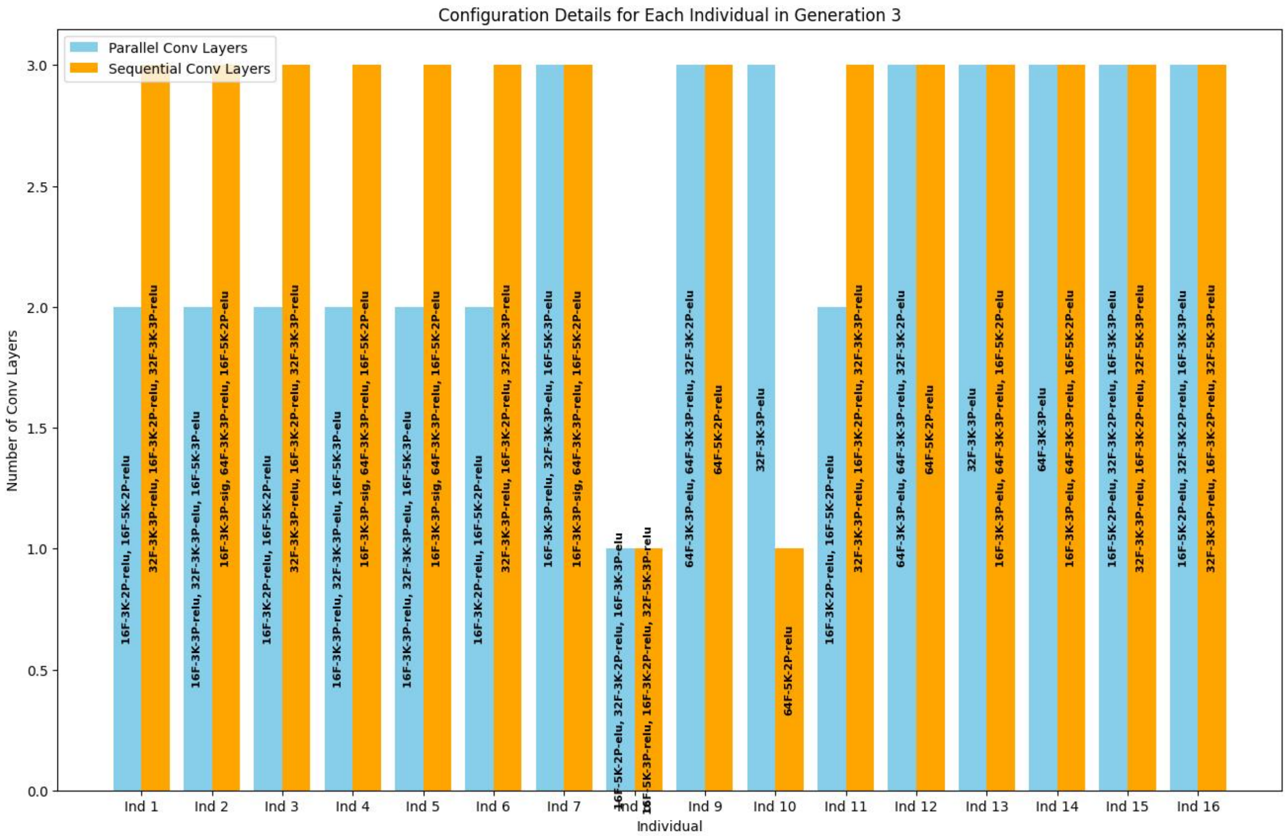

Figure 6.22: List of all the parameters used in each layer

At the end of Gen 3 all the individuals are giving almost similar accuracy since all the individuals are the best, selected from the previous generations. It has been concluded that models with 3 parallel and 3 sequential layers are best for the given dataset. However we can do experiments with different datasets to get the model that gives the best accuracy.

6.2.4 Comparison of Accuracy

Upon finalizing the model , the analysis was extended to compare its performance with other datasets including WISDM ,PAMAP, HHAR and the raw sensor dataset.The comparison was crucial to validate the adaptability and robustness our the final model across different types if sensor data.

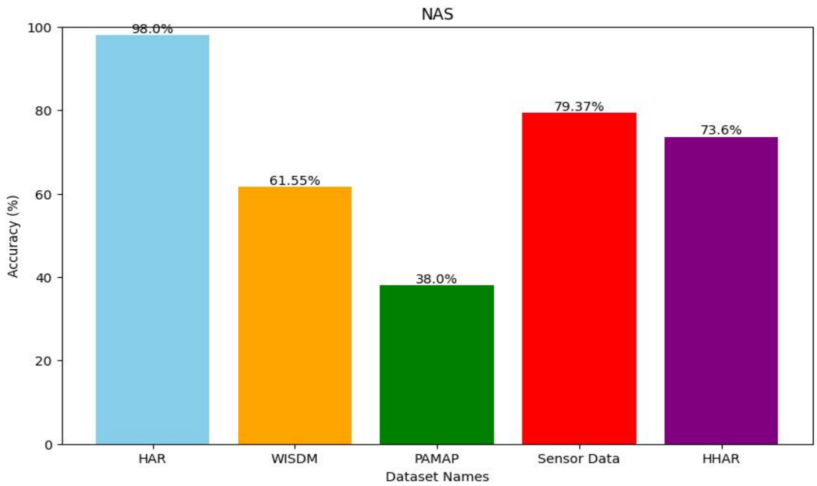

Figure 6.23: Accuracy across various datasets with best configuration

Comparative analysis of the results obtained from all the implemented models for UCI HAR dataset and the collected data.

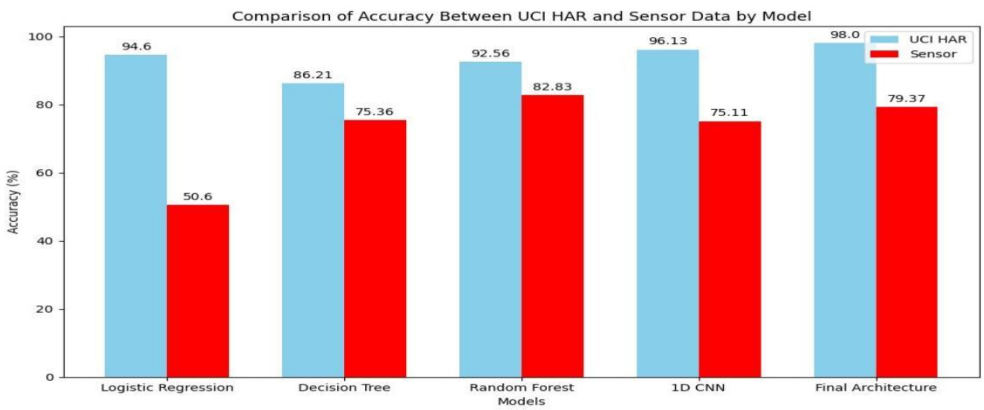

Figure 6.2: Accuracy comparison between UCI HAR and Custom dataset

Chapter 7

CONCLUSION

This project is an attempt to provide the best model with best architecture and best parameters across various datasets using a Deep Learning model and an Evolutionary Algorithm for Hyper parameter tuning in the field of Human Activity Recognition.
After implementation of all ML, ensemble and DL models on chosen dataset, an experimental comparison is done on all models using performance metrics like feature count, accuracy, execution time etc. Among ML models, LR and RF are performing well and CNN among DL models. In fields like HAR for research work, more importance should be given to dynamic activities because if anyone implement this in any of the possible sensor-based systems or Artificial Intelligence (AI) based systems in real time, so that it will be useful for security purposes, motion recognition, fall detection, elderly care, patient monitoring system etc, then the activities which require better tracking are dynamic activities for which the CNN is performing good.

Thus, for further phase, CNN is considered with some parameters and implemented hyper-parameter tuning using GA. Then an architecture for CNN is fixed and implemented along with GA to find the best parameters. Again the tuned parameters were given to CNN and there was a slight improvement in the accuracy. Since, the considered architecture might not be the best one, so to find the best possible architecture, in this project a neural architecture search is tried using GA on the CNN model and got the architecture with parameters. Then the resultant architecture and the parameters are given to the CNN model and seen a huge improvement in the performance of the model. Then, implemented other datasets using this architecture to verify the performance along with other ML and Ensemble models. The following result is obtained: any dataset implemented using this architecture, CNN and RF are showing better performance than other models.

Since, for the justification that CNN is better than other implemented models, a real-time RAW sensor data is collected using smart-phones in which the accelerometer and gyroscope values will be obtained. First, all possible models are implemented and the performance of LR is least among all other models. Also, the results shown that RF and CNN performs better than any other model with the resulting architecture. But RF has a problem of over-fitting (it will be depending upon the given train dataset and gives the results. But when given with any new data, it fails to perform well) thus, it may not be considered as best model. Then, to find the best architecture along with best parameters for raw-data, a neural architecture search is tried on the data and got the best model with best possible architecture and parameters. Then, implemented the model again with the resulting parameters and architecture and the results proved an increase in the performance of the model. Hence, CNN is the best model among all the ML, Ensemble and DL models in the field of HAR.

Chapter 8

FUTURE ENHANCEMENT

The initial experiments are on UCI-HAR dataset which is a preprocessed and well organized one. Architecture search, parameter adaptation and incorporation enabled to have a dataset neutral architecture which will help in HAR (an architecture that gives best possible results). The aim of this project is to come up with a DL model which performs equally good with the most standard and important datasets used for HAR whether it is preprocessed or raw-data. Then the same architecture was tried on manually collected datasets as well and found to be working good. The future work may include trying out even more datasets with the proposed architecture. Tried adding more parameters to get a better architecture and also can add more activities and also try more datasets as well. Once established, then it can be tried in recognizing the activities in some other domain. Also, the application can be implemented in real-time in any of the possible sensor-based systems or Artificial Intelligence (AI) based systems so that it will be useful for security purposes, motion recognition, fall detection, elderly care, patient monitoring system etc. NAS can be further improved with the objectives of minimizing the time, minimizing the no of parameters, but not compromising the performance.

REFERENCES

[1] Kulsoom, Farzana Narejo, Sanam Mehmood, Zahid Chaudhry, Hassan Butt, Ayesha Bashir, Ali. (2022). “A Review of Machine Learning-based Human Activity Recognition for Diverse Applications. Neural Computing and Applications”. 34. 10.1007/s00521-022-07665-9.

[2] Zunash Zaki, Muhammad Arif Shah, Karzan Wakil, Falak Sher. (2020). “Logistic regression based human activity recognition”. Journal of mechanics of continua and mathematical sciences. Vol.-15, No.-4, April (2020) pp 228-246 ISSN (Print) 0973-8975.

[3] M. Janaki, Dr. S.N. Geethalakshmi. (2023). “An Efficient System for Human activity recognition and monitoring for elderly people using Machine Learning”. Scandinavian Journal of Information Systems, 35(1), 1194–1206.

[4] Priyadarshini, I., Sharma, R., Bhatt, D. et al. “Human activity recognition in cyber-physical systems using optimized machine learning techniques”. Cluster Compute 26, 2199–2215 (2023). https://doi.org/10.1007/s10586-022-03662-8.

[5] Saurabh Gupta, “Deep learning based human activity recognition (HAR) using wearable sensor data”, International Journal of Information Management Data Insights, Volume 1, Issue 2, 2021, 100046, ISSN 2667-0968, https://doi.org/10.1016/j.jjimei.2021.100046.

[6] Essa, Ehab Abdelmaksoud, Islam. (2023). Temporal-channel convolution with self-attention network for human activity recognition using wearable sensors. Knowledge-Based Systems. 278. 110867. 10.1016/j.knosys.2023.110867.

[7] Ankalaki, Shilpa. (2022). A customized 1D-CNN approach for sensor-based human activity recognition. International Journal of Advanced Technology and Engineering Exploration. 9. 216-231. 10.19101/IJATEE.2021.874828.

[8] Akter, Morshead, Shafew Ansary, Md. Al-Masrur Khan, and Dong-wan Kim. 2023. "Human Activity Recognition Using Attention-Mechanism-Based Deep Learning Feature Combination" Sensors 23, no. 12: 5715. https://doi.org/10.3390/s23125715.

[9] Zhang, Shibo, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. 2022. "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances" Sensors 22, no. 4: 1476. https://doi.org/10.3390/s22041476.

[10] Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, Miguel R. Luaces, “Deep learning models for real-life human activity recognition from smartphone sensor data”, Internet of Things, Volume 24, 2023, 100925, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2023.100925.

[11] O. S. A. Aboosh, A. N. Hassan and D. K. Sheet, "Fake Video Detection Model Using Hybrid Deep Learning Techniques," 2023 6th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2023, pp. 499-504, doi: 10.1109/ICOIACT59844.2023.10455952.

[12] Séamus Lankford, Diarmuid Grimes, “Neural Architecture Search using Particle Swarm and Ant Colony Optimization”, 2024, 2403.03781, arXiv, cs.NE.

[13] Z. a. Almoussawi, K. S. Kumar, T. Veena, M. M. Adnan and M. B. Jabar, "Face Detection and Classification using Convolutional Neural Network with Channel Attention Module," 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, India, 2023, pp. 1-6, doi: 10.1109/ICMNWC60182.2023.10435738.

[14] Reddy, K.A.K., Sai, C.V., Akella, S.V.V.S., Kumar, P. (2023). A Hybrid Architecture for Action Recognition in Videos Using Deep Learning. In: Saraswat, M., Chowdhury, C., Kumar Mandal, C., Gando mi, A.H. (eds) Proceedings of International Conference on Data Science and Applications. Lecture Notes in Networks and Systems, vol 552. Springer, Singapore. https://doi.org/10.1007/978-981-19-6634-7_23.

[15] Shakya, Sarbagya & Zhang, Chaoyang & Zhou, Zhaoxian. (2018).

"Comparative Study of Machine Learning and Deep Learning Architecture for Human Activity Recognition Using Accelerometer Data". International Journal of Machine Learning and Computing. 8. 10.18178/ijmlc.2018.8.6.748.

[16] Muralidharan, K. & Ramesh, Anirudh & Rithvik, G. & a a, Reghunaath & Prem, Saket & Gopinath, Masila Pandiasankar. (2021). "CONVOLUTION APPROACH TO HUMAN ACTIVITY RECOGNITION USING SENSOR DATA AND COMPARISON WITH MACHINE LEARNING ALGORITHMS". International Journal of Cognitive Computing in Engineering. 2. 10.1016/j.ijcce.2021.09.001.

[17] Balaha, Hossam Hassan, Asmaa. (2023). "Comprehensive machine and deep learning analysis of sensor-based human activity recognition. Neural Computing and Applica- tions". 35. 1-39. 10.1007/s00521-023-08374-7.

[18] Voicu, Robert-Andrei Dobre, Ciprian Bajenaru, Lidia Ciobanu, Radu-Ioan. (2019). "Human Physical Activity Recognition Using Smartphone Sensors". Sensors. 19. 458. 10.3390/s19030458.

[19] P. Rojanavasu, P. Janta Wong, A. Jitpattanakul and S. Mekruksavanich, "Improving Inertial Sensor-based Human Activity Recognition using Ensemble Deep Learning," 2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommuni- cations Engineering (ECTI DAMT NCON), Phuket, Thailand, 2023, pp. 488-492, doi: 10.1109/ECTIDAMTNCON57770.2023.10139689.

[20] Pramila R.P., Gehlot A., Chafik B., Sundaram A., Lekha H., "Human Activity Recog- nition with Anomaly Prediction for E-Health Systems using Lightweight AI", (2022) IEEE International Conference on Knowledge Engineering and Communication Sys- tems, ICKES 2022, Cited 1 times. DOI: 10.1109/ICKECS56523.2022.1006.

[21] Sarkar, A., Hossain, S.K.S. Sarkar, R. Human activity recognition from sensor data using spatial attention-aided CNN with genetic algorithm. Neural Compute Applica 35, 5165–5191 (2023). https://doi.org/10.1007/s00521-022-07911-0.

[22] Raziani, Saeid Azimbagirad, Mehran. (2022). Deep CNN Hyperparameter Optimiza- tion Algorithms for Sensor-based Human Activity Recognition. Neuroscience Infor- matics. 2. 100078. 10.1016/j.neuri.2022.100078.

[23] Steven R. Young, Derek C. Rose, Thomas P. Karnowski, Seung-Hwan Lim, Robert M. Patton (2015). "Optimizing deep learning hyper-parameters through an evolutionary algorithm." Oak Ridge National Laboratory PO Box 2008, MS-6085 Oak Ridge, TN 37831 youngsr@ornl.gov.

[24] A. Shrestha and A. Mahmood, "Optimizing Deep Neural Network Architecture with Enhanced Genetic Algorithm," 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 1365-1370, doi: 10.1109/ICMLA.2019.00222.

[25] Guo, B.; Hu, J.; Wu, W.; Peng, Q.; Wu, F. The Tabu - Genetic Algorithm: A Novel Method for Hyper-Parameter Optimization of Learning Algorithms. Electronics 2019, 8, 579. https://doi.org/10.3390/electronics8050579.

[26] Okewu, E., Misra, S., Lius, FS. (2020). Parameter Tuning Using Adaptive Moment Estimation in Deep Learning Neural Networks. In: Gervasi, O., et al. Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science (), vol 12254. Springer, Cham. https://doi.org/10.1007/978-3-030-58817-5_20.

[27] Loussaief, Sehla and Afef Abdelkrim. “Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms.” International Journal of Advanced Computer Science and Applications 9 (2018): n. Pag.

[28] Raji,Ismail Damilola, Habeeb Bello-Salau, Ime Jarlath Umoh, Adeiza James Onumanyi, Mutiu Adesina Adegboye, and Ahmed Tijani Salawudeen. 2022. "Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models" Applied Sciences 12, no. 3: 1186. https://doi.org/10.3390/app12031186.

[29] Xueli Xiao, Ming Yan, Sunitha Basodi, Chun yan Ji, Yi Pan. 2020. “Efficient Hyperpa- rameter Optimization in Deep Learning Using a Variable Length Genetic Algorithm” 2006.12703, arXiv, cs.NE.

[30] Hamdia, K.M., Zhuang, X. Rabczuk, T. An efficient optimization approach for design- ing machine learning models based on genetic algorithm. Neural Compute Applica 33, 1923–1933 (2021). https://doi.org/10.1007/s00521-020-05035-x.

[31] X.J. Luo, Lukumon O. Oyedele, Anuoluwapo O. Ajayi, Olugbenga O. Akinade, Ha- keem A. Owolabi, Ashraf Ahmed, Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings, Renew- able and Sustainable Energy Reviews, Volume 131,2020,109980, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2020.109980.

[32] Ayan, E. Genetic Algorithm-Based Hyperparameter Optimization for Convolutional Neural Networks in the Classification of Crop Pests. Arab J Sci Eng 49, 3079–3093 (2024). https://doi.org/10.1007/s13369-023-07916-4.

[33] Y. Xue, Y. Wang, J. Liang and A. Slowik, "A Self-Adaptive Mutation Neural Architec- ture Search Algorithm Based on Blocks," in IEEE Computational Intelligence Maga- zine, vol. 16, no. 3, pp. 67-78, Aug. 2021, doi: 10.1109/MCI.2021.3084435.

[34] Franchini Giorgia; Ruggiero Valeria; Porta Federica; Zanni Luca. 2023. “Neural ar- chitecture search via standard machine learning methodologies”, Franchini, Giorgia; Ruggiero, Valeria; Porta, Federica; Zanni, Luca. - In: MATHEMATICS IN ENGI- NEERING. - ISSN 2640-3501. - 5:1(2023), pp. 1-21. [10.3934/mine.2023012].

[35] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen and K. C. Tan, "A Survey on Evolutionary Neural Architecture Search," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 2, pp. 550-570, Feb. 2023, doi: 10.1109/TNNLS.2021.3100554.

[36] Santiago Bonilla Diaz, E., Naredo, E., Francisco Mateo Díaz, N., Mota Dias, D., Ale- jandra Bonilla Diaz, M., Harnett, S.and Ryan, C. (2024) ‘Neural architecture search for bearing fault classification’, available: https://doi.org/10.34961/researchrepository- ul.25398391.v1.

[37] Ali, M.J., Moalic, L., Essaid, M., Idoumghar, L. (2024). Robust Neural Architecture Search Using Differential Evolution for Medical Images. In: Smith, S., Correia, J., Cintrano, C. (eds) Applications of Evolutionary Computation. Evo Applications 2024. Lecture Notes in Computer Science, vol 14635. Springer, Cham. https://doi.org/10.1007/978-3-031-56855-8_10.

[38] Kang,Jeon-Seong, JinKyu Kang, Jung-Jun Kim, Kwang-Woo Jeon, Hyun-Joon Chung, and Byung-Hoon Park. 2023. "Neural Architecture Search Survey: A Computer Vision Perspective" Sensors 23, no. 3: 1713. https://doi.org/10.3390/s23031713.

[39] *L* Ma et al., "A Novel Fuzzy Neural Network Architecture Search Framework for Defect Recognition with Uncertainties," in IEEE Transactions on Fuzzy Systems, doi: 10.1109/TFUZZ.2024.3373792.

[40] N. Fayyazifar, "An Accurate CNN Architecture for Atrial Fibrillation Detection Using Neural Architecture Search," 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 2021, pp. 1135-1139, doi: 10.23919/Eu- sipco47968.2020.9287496.