Comparing the performance of deep learning and machine learning models for recognizing human pysical activity.

Kavya Naveen Kumar C00223350

MSc in Data Science

Institute of technology Carlow

August, 2018

**Abstract**

The urge to make everyday experiences more comfortable and sophisticated with the technology is increasing with the growth in ubiquitous computing. Human activity recognition developed with the same objective. In the last several decades, human activity recognition caught significant research attention from a wide range of human-computer interaction and pattern recognition researchers because of its outstanding applications such as face recognition, image recognition, smart home healthcare and many others. For example, activity recognition systems can be utilized in a smart home health care system to improve patient's recovery processes. Different sensors can be used in multiple ways for human activity recognition in smartly regulated conditions. Amongst which, usage of wearable sensors for physical human activity recognition contributes essential knowledge of one's level of functional capacity and lifestyle. Human activity recognition facilitates a broad scope of prevalent computing applications by identifying the actions performed by its user.

This analysis is intended to perform the Human activity using nine diﬀerent supervised machine learning algorithms and a deep learning multi-layer perceptron model with the implementation of dimensionality reduction technique with the use of dimensionality reduction techniques with an objective to obtain better predictive accuracy. Adapting the Principal component analysis technique for feature reduction resulted in 100 features. K Nearest Neighbor, Decision Tree, Random forest, Support vector machine, Naive Bayes, Multinomial Logistic Regression, linear discriminant analysis, Artificial Neural Network algorithm and deep learning multilayer perceptron model were used to achieve the classification task. The study resulted in deep learning multilayer perceptron model attaining the best prediction accuracy of 98.47 %. The Artificial neural networks and linear discriminative analysis models also produced exceptional predictive accuracies. However, the other algorithms failed to prove eﬃcient.

**1 Introduction**

**1.1 Background**

Activity recognition is the task of identifying the several physical activity made by multiple users from a set of observations recorded during the user activity in the context of the definitive environment. Recent times have seen the principles of Human Activity Recognition (HAR) providing to various challenging applications built on the increase in ubiquitous, wearable and practical computing. Human Activity recognition has become an important technology that is changing the view of people's daily routine offering to a wide range of applications as assistive technology, health and fitness tracking, elder care and automated surveillance to name a few. Additionally, the research in activity recognition has been so rapid and exceptional that it is starting to cater to applications that go beyond the activity recognition. Nevertheless, the main concerns of the present Human activity recognition systems are the average classification accuracy, and a higher computation overhead is needed to improve the prediction accuracy, and also as the field is rich in practical applications, the difficulties arising for activity recognition are numerous.

The usual workflow of a human activity recognition task deals with data acquisition from a wearable sensor or an external device as cameras. This data is then processed to obtain a cleaner and modified data suitable for additional processing. The data is explored to understand its nature and the type of processing that can be applied in the next stages. The design of the later stages could vary extensively on the application and its area. However, usually, the data is managed to make it more relevant by extracting more valuable features from it. Finally, a predictive model is created to recognize the actions performed by the user and is evaluated for its performance. This detected activity can be repurposed for various applications as detecting a change in the user activity, assistance in regard to the detected activity etc.

**1.2 Research Problem**

In the possibly simple approach of performing HAR, many concerns and challenges are encountered such as selecting the right tools and methods for gathering, storing and manipulating the data. Picking the right algorithm to perform predictions is of utmost importance as there is a necessity to capture the inter-class variability and the intraclass similarity. Typical resource constraints like processing power, availability of time and suﬃcient storage are diﬃcult to handle. There is also a necessity to have a trade-oﬀ between system latency, accuracy, and processing power.

Picking the right sensor or combination of sensors, selecting the attributes and metrics to be measured, placing the sensor at the right location are all crucial in their way. Furthermore, all these must be done by considering the user privacy and usability in context. The process must not be obtrusive for the user and must adapt to the user's behavior and their environment as an entire process can be highly sensitive to the participation and the interaction with the user.

The research question that is planned to be addressed in the current study can be stated as follows:

Can deep learning model give the better classification accuracy of recognition of human physical activity using inertial sensor data compared to supervised machine learning algorithms?

Algorithms: Decision Tree, Random forest, K Nearest Neighbor, Support vector machine, Naive Bayes, Multinomial Logistic Regression, linear discriminant analysis, Artificial Neural Network algorithm.

**1.3 Research Objectives**

A possible solution in overcoming these challenges could be by reviewing and analyzing the existing research and picking the sensor and a location with best-proven capabilities. This eliminates the risk of data inaccuracy and justifies the eﬀort put in engineering the captured data. Also, considering the computing capabilities at hand, performing dimensionality reduction is mandatory. Performing feature reduction is a better option for minimizing the captured activity data, as more amount of user activity can help in create and train a better model. The principal component analysis is the most popular and eﬀective feature reduction technique discovered through the literature review. Principal component analysis performs feature reduction but by generating new features that are a linear combination of the existing features. However, with a smaller number of features, the PCA captures significantly more information about the target feature.

All these data manipulation tasks are of no use if no scientific approach can exploit these. However, picking the right algorithm to perform the task is complicated. A possible approach could be picking multiple algorithms, each having a diﬀerent mechanism of action to tackle the task. Evaluating and comparing each of these models can help in selecting the right approach for performing activity detection. The literature review identified diﬀerent mechanisms in which supervised machine learning and deep learning methods can be performed. Algorithms from each of these methods can be utilized to have a robust and significant solution.

So the primary objective of the research is to determine the algorithm that can perform Human Activity Recognition with classification accuracy higher than the current state of the art technique. To achieve this goal, many tasks must be achieved. A list of these tasks is stated below –

1. Extensive study of the existing literature on experiments and research performed under Human Activity Recognition to identify research gaps.

2. Design a solution to perform Human Activity Recognition by detecting the interclass diﬀerences and the intra-class similarity within the activities.

3. Implement the solution reinforced in the design and induce models to perform Human Activity Recognition to obtain the targeted accuracy.

4. Evaluate the performance of the induced models.

5. Place the findings in the field of study.

**1.4 Research Methodology**

As current study aims to compete with an existing HAR system, this work will primarily perform secondary research using the existing data from the state of the art experiment and no other data will be generated for the scope of this project. Additionally, secondary research of reviewing existing literature on the topic of HAR and its applications will be performed in literature review.

The core experiment to be performed will be based on the quantitative objective of picking the best classifier measured through a predictive classification accuracy value. The research involves the solution to utilize mathematical modeling in creating the modified datasets and in performing evaluation using statistical techniques. So the study can be stated as of empirical research form.

The current study involves observing and analyzing the existing literature to form a hypothesis that can be proved through the experiment. The results of the hypothesis tests performed in the experiment can be transformed into theories that can be generalized to specific contexts.

**1.5 Scope & Limitations**

Various studies in HAR have discovered increased predictive accuracy results when multiple datasets are correctly integrated and used for training the model (Mannini et al. (2013); Sucerquia et al. (2017)). However, this study will utilize only a single dataset from an existing state of the art technique.

The benchmark experiment utilized for the study provides only the derived components of the actual data acquired from the sensors. The study could have greatly benefited if the raw signals from the sensors were provided, but only the derived signals were accessible.

There are plenty of methods to perform dimensionality reduction, but one of the most convenient techniques from the literature review were selected due to the time and computing constraints of the experiment. Similarly, multiple machine learning models and plenty of their parameters were discovered from the literature that can be utilized to gain better insights from the data. However, the study had to limit the number of algorithms and parameters to manipulate, so nine algorithm are employed with limited parameter tuning.

**1.6 Organization of Dissertation**

The rest of the document will be structured as follows –

• Section 2 presents an overview of the current literature on Human activity recognition. It outlines the diﬀerent sensor technologies exist, and the eﬀect of its placement on the appropriate location of the body has on the accuracy of the study. The chapter also examines the diﬀerent supervised machine learning and deep learning techniques that are used to perform HAR. The theory, applications and example studies using the algorithms are discussed here. Various evaluation metrics are also studied to identify the method that is most suitable for the experiment.

• Section 3 outlines the basic design of the research. The structure of the Section is based on the CRISP-DM methodology, so initially, the business understanding phase is designed followed by data understanding phase. Data preparation to be performed is reviewed next, followed by the modeling stage. The chosen evaluation metric and the evaluation strategy are designed in the end. The section concludes by stating the strengths and limitations of the developed design.

• Section 4 details the practical implementation of the experiment and performs a critical evaluation of the experiment and the results obtained. Each of the utilized technique and their corresponding results obtained are illustrated here.

Comparison of each algorithm is performed to understand the outcome. The hypothesis of the study is also evaluated here. This chapter concludes by stating the strengths and limitations of the implementation. This section is also structured according to the CRISP-DM methodology.

• Section 5 concludes the study by reiterating the research question and the problem definition. It also states the workflow of the entire process and additionally states the scope and limitations of the study. The chapter section by specifying fields of possible future work.

**2 Literature review**

This section gives a complete review of the relevant literature about human activity recognition and the state-of-the-art techniques involved in its detection. The appropriate sensor technologies used in similar studies are also examined along with their practical application areas. Additionally, the ideal location of the sensor on the human body is also discussed. So this section provides the groundwork for selecting the technology and the specific location of aﬃxing the sensor on the body of the user before performing the actions. Furthermore, this section also discusses the technologies used for the detecting of the physical activity; firstly, the theory of detection using machine learning and deep learning is discussed along with its applications; it is then followed by discussing the family of machine learning techniques and deep learning techniques that can be applied. The section will conclude by highlighting the state of the art technologies regarding gathering the data and provides justification to employ a set of techniques to perform better in the detection task.

**2.1 Human Activity Recognition**

Human activity recognition has gained much importance in the past decade due to its numerous applications in human-centric applications as in the field of healthcare (Tentoriand Favela,2008; Carús-Candásetal.,2014), ubiquitous computing, ambient-assisting living(Stančićetal.,2017), surveillance and security(Choudhury et al.,2008;Turagaetal.,2008; Poppe,2010). In recent times, smartwatches, Smartphones, ad-hoc wearable devices, and fitness trackers are frequently utilized to observe human activities. These acquired data by the hosted sensors are processed by machine-learning-based algorithms to identify human activities. An essential goal of the HAR in the current scenario is to identify the actions of the user to assist them with their tasks with the help of computing systems Abowd et al. (1998). Computer vision research has been contributing a lot to this aspect of the study. Human activity recognition here mainly refers to physical human activity, as objected to cognitive, mental activities and workload which are part of a diﬀerent, wider research field (Rizzo et al., 2016; Longo, 2011, 2012, 2015, 2016; Moustafa et al., 2017). The initial research on HAR involved detecting gestures and activities from still images and videos in restricted environments and under-constrained settings Turaga et al. (2008); Mitra & Acharya (2007). A significant number of domains have been discovered to benefit due to HAR as in the case of Activities of Daily Living (ADL's) by Katz et al. (1970), which was one of the initial researchers performed as an application of activity recognition, which further boosted the research by Bao & Intille (2004); Ravi et al. (2005); Logan et al. (2007); Tapia et al. (2004). The traditional medical procedures were challenged by introducing the HAR to support patients' daily activity monitoring especially for patients with chronic impairments such as Parkinson's disease, Alzheimer (Corchado et al.,2008) or visual impairments or other medical diagnosis or even for rehabilitation (Starner et al., 1997; J. Chen et al., 2006; Oliver & Flores-Mangas, 2007; Bachlin et al., 2009; Tessendorf et al., 2011) . Another area of HAR application is the accurate classification of physical activity,e.g., for exercise monitoring (exercise adoption, individual incentivizing or prevention of obesity).Bullingetal. (2014). HAR also provided excellent results for other areas of minor severity as the sports and entertainment category (Kunze et al., 2006; Minnen et al., 2006; Ladha et al., 2013), the operations and industrial sector (Maurtua et al., 2007; Stiefmeier et al., 2008). HAR was further explored to cater simple human activities as transportation routines (Krumm & Horvitz, 2006), brushing teeth (Lester et al., 2006) and medicine intake (Wan, 1999; De Oliveira et al., 2010). The newest and current practices of human activity recognition were for gaming consoles as Microsoft Kinect where body gestures and actions are identified to provide, an enhanced gaming experience (Shotton et al., 2013) by using an RGB-D camera, was proposed by (Salvatore Gaglio, 2015). This approach uses Kinect for the calculation of some relevant joints of the human body; K-means clustering, and hidden Markov models machine learning techniques, were merged to detect the positions included while performing an activity, to predict them. This experiment used a public CAD-60 dataset and on a new Kinect Activity Recognition dataset.

**2.2 Sensor Technology for Gathering Data**

Activity recognition can be performed for a single user or multiple users. Multiple user recognition can be performed to identify and track an individual user’s actions. This can be performed for surveillance and monitoring purposes using video camera footage. However, people are concerned about their privacy when the cameras are in use; also installation comes with the non-negligible casts that are required to operate the system (Lukas Koping et al. (2018)). Also, for a single user activity recognition process, there could be multiple ways to gather data. With the progress in sensor technology, a popular method of HAR is known to be using Inertial Sensors. Most inertial sensors have become portable to be connected to the human body. Battery levels and flexible design which are designed for more extended recording and monitoring purposes simultaneously with computing and dynamically consistent interaction, make them manageable and more suitable to use (Florentino-Liano et al., 2012; Bulling et al., 2014). The first application of sensors for activity recognition was in the context of smart homes by 'Neural Network House' along with other applications that help create versatile systems for better user experience of their smart home (Mozer, 1998; Leonhardt & Magee, 1998; Golding & Lesh, 1999; Ward et al., 1997). The inertial sensors with integrated gyroscopes and accelerometers have been utilized for various purposes as medical diagnosis and treatment (Powell et al., 2007), Tele-Rehabilitation (Winters et al., 2003), Fall detection (Wu & Xue, 2008) and human movement monitoring (Sabatini et al., 2005).

Gyroscope, Magnetometer, accelerometer, Bluetooth, microphones, proximity, Wi-Fi and light sensor and cellular radio sensors are the multiple ranges of sensors provided by smartphones (Henry Friday Nweke et al. (2018)). The incorporation of diﬀerent sensors at various positions has been employed previously to provide differing results. One of the most popular sensors used quite frequently for similar studies involving repetitive actions is an accelerometer. An accelerometer is an electromechanical device which is used to measure static and dynamic acceleration forces. For instance, the angle of tilt or inclination of the device can be calculated by measuring the gravity acceleration. The accelerometer was used in multiple studies as a motion sensor, yielding excellent results for their area of application (Bao & Intille, 2004; Mi-hee et al., 2009; Khan et al., 2008).

Sensors as image and audio based have also been utilized for applications as image tagging and activity detection using noise levels for instance if the noise is less, the user could have been asleep (Qin et al., 2014; Bieber et al., 2011). Global positioning system (GPS) sensors are quite widely used as well. The GPS sensor was employed to detect the user activity through location-based signals across single and multiple users and was also used to track any unusual activities by the users (Patterson et al., 2003; Ashbrook & Starner, 2003; Liao et al., 2007). Medical applications which involve detecting the patients' critical medical information as heart or respiration rate or other vital metrics utilize a series of biosensors. Sung et al. (2004) detect body temperature using an arm and chest accelerometers and polar heart rate receiver sensors to detect emergencies due to weather conditions as hypothermia for soldiers surviving in severe weather conditions. Biosensors are also used to create smart clothing which can be used to detect body postures (Harms et al., 2008). Wren & Tapia (2006) utilize infrared sensors which can detect temperature, which was further utilized to diﬀerentiate diﬀerent levels of activities producing low and high levels of heat and thus enabling to be diﬀerentiated.

There are also combinations of sensors utilized together for better detection of the signals. The accelerometer was combined with the psychological sensor to detect signals as skin temperature and energy expenditure (S.-I.Yang&Cho,2008). Apart from these various types of sensors, the simplest and one of the eﬃcient is the accelerometer in combination with a gyroscope, which is a device used to detect angular velocity (Lazzarini, 2007). Due to development of mobile phone technology, the smartphones have built-in accelerometers, and gyroscopes which make the study of activity detection furthermore simplified (Brezmes et al., 2009; Oh et al., 2010). Anguita et al. (2013); Kwapisz et al. (2011) utilize the sensors in the mobile phone to perform the task of human activity recognition.

**2.2.1 Placement of the Wearable Sensor**

Cleland et al. (2013) examine the importance of sensor and the optimal placement of it on the body of the user. Their study explains that the acceleration signal values steadily increase in magnitude as the placement of the sensor moves from head to feet. So it is evident that the location has a direct impact on the results obtained for the HAR process. As shown in figure 1, the sensor can be placed at multiple locations of the body (Attal et al., 2015).

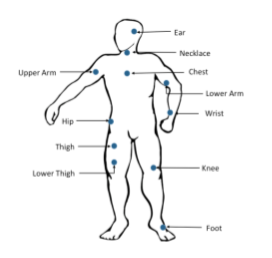


Figure 2.1: Illustration of Wearable Sensor Placement

Multiple studies have utilized the sensor placed at various body parts of the user and have received various ranges of accuracies. Parkka et al. (2006) have studied HAR by placing the sensors on the wrist and chest and performed various activities for 2 hours. The best results out of the three classifiers yielded were of 83% accuracy. Investigating various other studies exhibited the best results from the study by Yeoh et al. (2008) which experimented using three sensors, one mounted at the waist and two more attached to the center of the thighs. Their experiment resulted in an overall accuracy of 100% by detecting four crucial tasks exceptionally well. However, observing the studies that utilized only a single sensor at one location of the body, a sensor at the waist or on the lower back have been yielding good results. Mathie et al. (2004); Gupta & Dallas (2014) have performed HAR task using a single sensor placed at the Waist and have received a classification accuracy of 98%, and Bonomi et al. (2009) have received an accuracy of 93% while detecting for similar activities. Hence, placement of a sensor on the waist can be seen as an ideal position as it is proven and additionally it is also closer to the center of mass of the body (C.-C. Yang & Hsu, 2010). Ruben San-Segundo et al. (2018) have performing robust human activity recognition using smartwatches and smartphones. This experiment proved that compared to a smartphone, continuously wearing smartwatches are more comfortable, they record whole body movement with higher variability. However, smartphones outperformed with the 98.1% accuracy.

**2.3 Modeling Approaches for Human activity recognition**

**2.3.1 Machine Learning**

**2.3.1.1 Theory**

In the past two decades have seen a lift in Machine learning becoming a critical component in Information Technology. With the constant increase in data availability, it is evident that smarter data analysis will be an integral part of the technological processes at every phase of life.

Machine learning as defined by Kelleher et al. (2015) is "an automated process that extracts patterns from the data."

The main idea behind machine learning is to enable the machine to learn the system automatically, with no human interventions based on the designed algorithms. These algorithms originated in many fields such as mathematics, theoretical computer science. It is important to note that the algorithms devised for this task must serve the purpose while being eﬃcient, which comprises both time and space eﬃciency. For the context of learning, the amount of data required by the algorithm is of primary importance and must be utilized to the maximum extent. The algorithms must also be made flexible to enable generalization to various applications. However, the primary motive of machine learning is to harness the predictive capabilities of the machine, and hence the predictive accuracies must be as high as possible with minimal error rate.

**2.3.1.2 Applications**

There is a vast number of applications of machine learning and its usages are ever increasing. Some of the most popular applications are (Alpaydin, 2014)–

• Creation of a good search engine requires the engine to process the search query, identify pages having the information and sort them according to a prescribed algorithm. Machine learning has increasingly been utilized to automate the process of web page ranking and determining the best results for a given query.

• Recommendation engines are another area which has seen rapid development as they help attract the users with newer products or services. E-commerce companies as Amazon and eBay utilize this system to analyze past purchases and viewing options to predict and enable future viewing and purchases. Netflix, which is a video rental store, also utilizes this mechanism. This application can also be termed as Collaborative filtering.

• Another application that is not entirely defined is the text translation problem. This problem is quite tricky as the machine needs to model the grammar and the language of the document. There is also extensive research in this area of application.

• Face recognition is an example of machine learning revolutionizing the current existing applications. The machine breaks down the face of the user into smaller parts as pixels to detect and classify the faces. This system is currently being utilized in various security applications and also social networking websites.

• Other applications that are utilizing machine learning are speech recognition, handwriting recognition, named entity recognition, failure and fault detection in several industrial equipment’s, gaming consoles.

**2.3.1.3 Machine Learning Approaches**

The study of Machine learning can be divided into multiple sectors by the types of work to be done, data dealt with or working procedure. Kelleher et al. (2015) have devised one such method of segregating the algorithms by their ability to learn from the data.

**Decision tree** – The idea behind this approach of machine learning is to utilize the intuitiveness of the data in deriving a model to perform machine learning. Information gain and Gini Impurity are the measures on which the learning process is performed by determining the features that best describe the entire data.

Decision trees are machine learning algorithms that follow information based learning approach. A decision tree algorithm approximates the target function by creating a solution that can be represented by a tree leading to the target feature. Decision trees are widely utilized as they can be represented in the form of if-else statements for better readability and understanding. The algorithm also requires minimal data preparation or feature engineering which can help save time and eﬀort. Hence, decision trees have been highly utilized in performing activity recognition. Fan et al. (2013) performed activity recognition using decision tree algorithms by constructing behavior and position vectors of users performing five diﬀerent activities. The study reported high classification accuracy and less time consumption.

**Random forest**- Random forest is one of the variants of the decision tree. It is developed by aggregating trees, it can be used for both classification and regression problems, but it is most popularly used for classification. It avoids overfitting, and it can deal with a large number of variables and makes feature selection based on the importance of each variable. It is a user-friendly methodology and has two parameters: which are number (ntrees) of trees, i.e., builds 500 trees by default and the variables are randomly sampled as candidates at each split (mtry).

**K-nearest neighbor algorithm-** The concept of similarity based learning is to observe the previously existing data to predict the future or unknown data. This technique utilizes the measure of similarity which indicates how similar or related multiple data points are to each other. This algorithm is described by the techniques used to process the pairwise data, their relationships and the assumptions in their relationships. The most frequent and natural algorithm of similarity based learning is the K-nearest neighbor (K-NN) algorithm. This algorithm assumes that if two data points, ai and aj are similar to each other, the corresponding target or outcome classes, xi, and xj are also similar (Hu et al., 2015).

Kaghyan & Sarukhanyan (2013) have studied the accelerometer of an android mobile phone and applied the K- Nearest neighbor algorithm to predict the activity of a single user and have received satisfactory results. Paul & George (2015) created a model called the Clustered KNN which is an improved KNN algorithm to detect four activities performed by four users and achieved measurable results by utilizing limited memory and a limited training data.

**Support Vector machine -** A Support Vector Machine is a discriminative classifier defined by a separating hyperplane. Support vector machine algorithms can be applied for both regression and classification tasks, but it is frequently used for classification purposes. For the given supervised learning designated training data, the algorithm outputs an optimal hyperplane which classifies unknown data. This hyperplane is a line dividing a plane into two parts wherein each class lay on either side in two-dimensional space.

**Naive Bayes algorithm –** The theory of probability based learning is to utilize the estimates of the likelihood of a data point in determining the target value of the point. The Bayesian algorithm assumes an underlying probabilistic model and captures the uncertainty of the model by calculating the probabilities of the target. The Bayesian algorithms are highly scalable and are hence widely used over plenty of applications.

Sarkar et al. (2010) have studied and compared the Naive Bayes algorithm to the Hidden Markov Model and the Conditional Random field model. They have demonstrated that previous studies have shown the Markov model outdoing the Naive Bayes algorithm. However, it stated that parameter estimation plays a massive role in a classiﬁer and performed two types of smoothing techniques to adjust the maximum likelihood of the classifier. Their experiment oﬀered significant improvement in the classification accuracies by the Bayes algorithm. Ravi et al. (2005) have performed a similar activity recognition system which aims to classify a set of eight diﬀerent activities, at four diﬀerent environment settings. Their study determined Naive Bayes to outperform all the other single classifiers in two of their settings.

**Multinomial logistic regression –** Multinomial logistic regression algorithm is one of the error based learning methods. The concept of error based learning is to initialize a parameterized model with a set of random parameters to identify the parameters which correspond to the minimized error value for the specified training instances. The model and its parameters are adjusted based on the error value to achieve high accuracy values. Multiple error metrics can be utilized under Error based learning, one of the most popular metrics is the Sum of squared errors.

Logistic regression, artificial neural net and Support vector machine are few of the most famous error based machine learning algorithms. The logistic regression is the odds of the target feature taking a specific value modeled by a combination of values taken by the feature vectors. A binary logistic regression model estimates the likelihood that the target is presently given a set of feature variables. Y. Chen et al. (2016) performed activity recognition using multinomial logistic regression with Bayesian regularization and have received a high accuracy of 93% over other algorithms discussed in their study. Artificial neural nets are widely used for HAR. Multiple studies have observed exceptional results in detecting user activities using ANN. Oniga & Suto (2014) stated that with the right topology and parameters, the ANN could detect the most complex activities. Multiclass SVM is another favorite technique as it can model data in diﬀerent non-linear distributions owing to its kernel function and can also be prevented from being sensitive to outliers and hence was used in multiple studies to achieve reasonable accuracy rates (Anguita et al., 2012; D. Lara & Labrador, 2013; Ravi et al., 2005)

Also, the survey by bin Abdullah et al. (2012) demonstrated that the popularity and eﬃciency of various classification algorithms that are commonly used in detecting ADL using embedded sensors from a smartphone device. The most commonly used algorithms that have been deemed eﬃcient as identified by this study were Hidden Markov Models, Multiclass Logistic Regression, Decision Tree, Naive Bayes, Support Vector Machine, Artificial Neural Network, K - Nearest Neighbor, Gaussian mixture models and multiple other ensembles and modified algorithms.

**Artificial neural networks-** Artificial Neural networks are composed of simple units called neurons; each unit does simple calculations like addition or multiplication. It takes the inputs from many other neurons and agglomerates the data that comes in and sends it downstream to other neurons. They are inspired by some of the computations that go on in the human brain. The primary objective is to build up a system to perform various computational tasks quicker than the traditional systems, and the working of the human brain inspires the idea of Artificial Neural Networks by making the right connections with the use of silicon and wires (www.tutorialspoint.com, 2018)

The human brain contains around 1011 neurons, and each of them is connected with other neurons by 1013 synapses (Gurzynski, P, Dlugosz, R, Talaska,T, Swietlicka, A,2013). A neuron consists of 4 parts: namely dendrite, cell body, axon, and synapses as shown in fig.2.2. Dendrite receives the stimuli from the external environment or another neuron, and all the incoming signals are summed up in the cell body to make an input signal. When the sum reaches a threshold value, the neuron shoots a signal, and it travels down the axon to the other neurons. Synapses are the connections between one neuron to another neuron.

The strength called the synaptic weights, of the connections, depends on the amount of signal transmitted. Synaptic weight is nothing but the inter-neuron connection strengths which stores the knowledge of the neural network. The output produced at each node is called its activation value. Networks acquire knowledge through learning which takes place by altering weight values associated with each link (www.tutorialspoint.com, 2018).

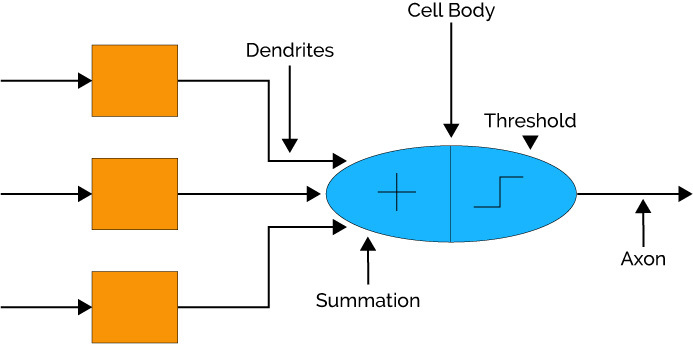
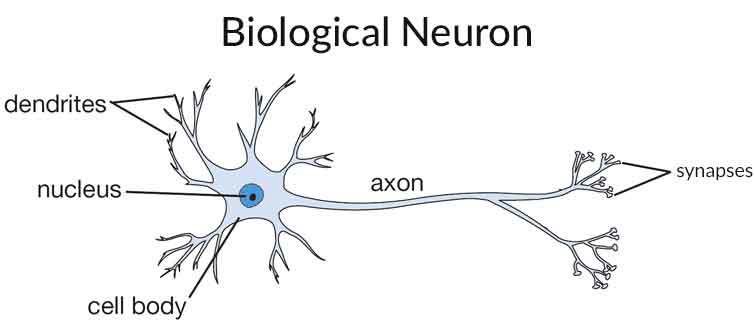


Fig 2.2.Human Neuron Fig 2.3. Artificial Neural Network

Fig 2.2. Illustrates the analogy of Artificial Neural Network v/s biological neural network in fig.2.3.

Working of the artificial neural network:

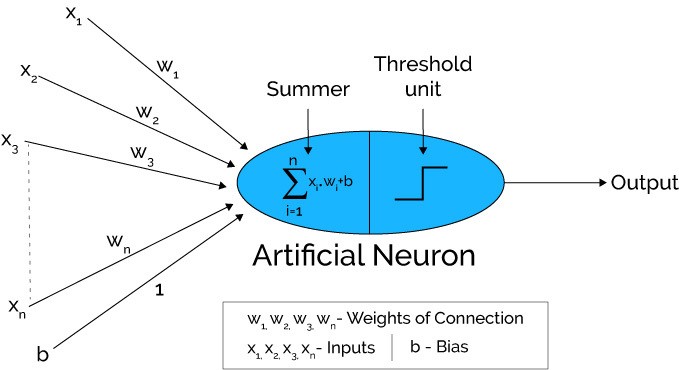


Fig 2.4: Working of artificial neural network

In the above figure it can be noticed that the artificial neural network can also be compared to the weighted directed graph where the Artificial neuron is a node and x1, x2, x3 and so on represents the directed edges with weights w1,w2,w3 and so on and these edges are interconnected between the input and output neurons. A pattern/image is given as an input to ANN from outside in the form of vector. The n(x) notation is assigned for ‘n’ number of inputs. Every input link is given its proper weights to get the desired output. The input with weights is added up in the summer as shown in figure 5.

If the weighted sum is zero, then a ‘b’ (bias) is added to ensure a non-zero output is obtained. The obtained value ranges from 1 to infinity. A threshold value is set up to get the desired value, and an activation function is used to get the required sum. The most commonly used activation function is a binary activation function (www.tutorialspoint.com, 2018).

Architecture of Artificial neural network.

Most of the Neural Networks consists of 3 layers: an input layer, hidden layer and output layer arranged in series as shown in fig 6. The Input layer receives the input from the input data/image on which network will learn, recognize about the further process. Then the hidden layers receive the signals from the input layer and get modified during the learning phase. Finally, the output layer responds to the learning phase and gives the desired output after following many iterations. Neural Networks are fully connected, and since the hidden layer is the middle layer, it is wholly linked to every neuron in its previous input layer and the next output layer.

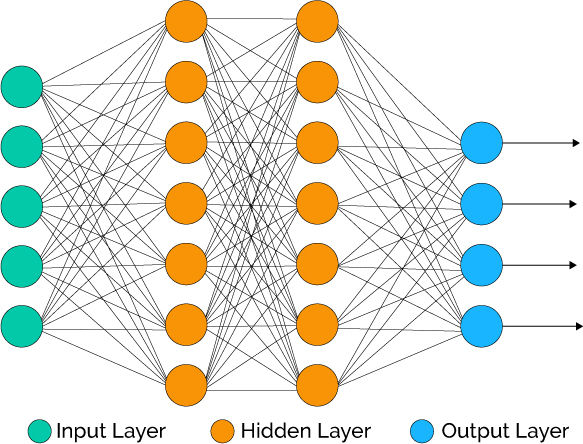


Fig 2.5. Architecture of ANN

There are many neural network Architectures, but the most widely used architecture is Feed-forward architecture or Multi-Layer Perceptron. These networks use more than one hidden layer of neurons.

The most popularly used learning algorithms used in ANN is the Backpropagation algorithm. In this algorithm, initially random weights are assigned to each link of the network, and if the desired output is not obtained there, an error is calculated. The input signals are propagated backward from the output layer to the input layer via the hidden layer iteratively until the network learns to optimize its weights and biases and gives the correct output.

#### Linear discriminative analysis- Linear discriminative analysis model is widely used for classification tasks. It creates the boundaries between the groups of different classes. Then projects data points on a line so that the groups get separated, and each group has a close distance to a centroid. It calculates the deviation from the whole group to determine separability and calculates the mean vectors of the data in all dimensions and then calculates scatter from representatives of the same class using the whole group scatter as a normalizer (Brownlee, J. (2016)).

.

#### Deep learning multilayer perceptron- Deep learning structures algorithms in layers to create an “artificial neural network” that can learn and make intelligent decisions on its own. Deep learning is a subfield of machine learning. While both fall under the broad category of artificial intelligence, deep learning is what powers the most human-like artificial intelligence (Zendesk. (2017)).

A multi-layer perceptron has the same structure as a single layer perceptron with one or more hidden layers. It consists of an input layer which receives the signal and the output layer that predicts the input, in between there are the hidden layers which are a computational engine if the multilayer perceptron. They train on a dependable and independent variable which are given as input and output respectively to the model and learns the correlation between the dependable and independent variables. Models training also includes adjustment of the weights and biases, parameters, to reduce the error. An error can be measured by the root mean squared error (RMSE) method and Backpropagation is used to make the weigh, bias adjustments.

**2.4 Model Evaluation Metrics**

**2.4.1 Performance Measures**

Performance measures encompass the final steps in a machine learning project. They emphasize and evaluate the correctness, eﬃciency, and usefulness of the design and the modeling process and additionally also provide reliability metrics on the entire procedure. There are multiple ways to evaluate the critical criteria based on the project. The appropriate criteria and its measure must be chosen depending on the research question and also the ability and feasibility of the researcher and the study.

**2.4.1.1 Confusion Matrix**

Confusion matrix also termed as a contingency table, provides a comprehensive overview by summarizing the classification results. It showcases the individual results for each of the classes by tabulating the predicted and actual classes.

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Actual | |
| Positive | Negative |
| Predicted | Positive | TP | FN |
| Negative | FP | TN |

Table.2.1 Confusion Matrix

TP – Indicates the true positives which are the number of positive outcomes correctly predicted as positive by the model

TN – Indicates the true negatives which are the number of negative outcomes correctly predicted as negative by the model

FP – Indicates the false positives which are the number of negative outcomes incorrectly predicted as positive by the model

FN – Indicates the false negatives which are the number of positive outcomes incorrectly predicted as negative by the model

Each of these measures has a significance of their own depending on the research question. Multiple standardized metrics can be defined from the confusion matrix –

• Accuracy – It is the total number of correct predictions proposed by the model which includes the positive and negative predictions.

Accuracy = number of correct predictions/ total number of prediction

= TP + TN / TP + TN + FP + FN

**2.4.2 Significance Tests**

The above-discussed metrics represent only the performance measures that can be used for each classifier. However, it is not suﬃcient to compare algorithms alone; the study must provide statistical evidence of the result obtained from the evaluation. Statistical tests come into play in this regard. Three tests were quite popular in evaluating HAR and are analyzed further.

The McNemar's test primarily is used to compare the diﬀerence between proportions in two matched samples. However, Everitt (1992) utilized this test to compare a set of classifiers trained and test on the same datasets. This test does not measure the entire variation in the training data as a partial amount of data is used as a holdout set. The next test is the k- fold cross-validation test, it is performed by creating k equal and disjoint sets of the dataset, the experiment is then run k times with each trial having a specific test set with all other sets merged to create the training set. This method permits each test set to be independent of each other which can be an excellent technique to evaluate with but also permits overlap of training dataset which might not result in a good variation. The last and interesting statistical test is the repeated random sub-sampling (RRSS) or the Monte Carlo cross-validation. It performs a Monte Carlo repetitions of a randomly sampled data, and the final results are obtained by aggregating the results for each repetition. Colaprico et al. (2015) utilize this validation technique to detect tumors related to breast cancer with the help of biomarkers.

Dietterich (1998) studied each of these statistical tests for computational power and cost for running the learning algorithms. The study concluded that the cross-validation tests were the most powerful followed by the McNemar's test. Altun & Barshan (2010) studied HAR through data collected from miniature sensors. The study involved determining the best classification algorithm out of the chosen six. The experiment was evaluated using multiple techniques, out of which RRSS, K fold cross validation proved to be most eﬀective.

**3 Design & Methodology**

This section details the plan and the design methodology for the current study. Several data mining methodologies were studied to identify a robust and well-structured approach for the data mining project. The Cross Industry Standard Process for Data Mining (CRISP-DM), a well-proven method is identified to conduct the current study, which at its core is a data mining project (Piatetsky, 2014). This methodology is an ideal sequence of events, and the current chapter will deal with each of these phases as a separate section. However, the steps can always be traced back to previous stages to repeat or manipulate a step to better suit the following stage.

The first stage is the Business Understanding phase; this is where the business perspective of the project is understood. The desired outputs of this phase are the primary objectives of the study. The next phase is the Data understanding phase. This is the stage where the actual data utilized in the project is acquired. Each element of the data is inspected and described. The data is also further explored to produce any initial findings and their impact on the subsequent project stages. The quality of the data is also accessed at this stage. The third phase is the Data preparation stage. The unnecessary or repetitive data can be eliminated. Further cleaning of the data is also done along with the development of new derived data elements of the existing data. All the manipulated data elements are finally integrated to create a final dataset to be utilized in the later stages. The next step is the modeling stage. The modeling techniques analyzed from the Literature review section are induced. This stage also involves experimentation with various parameter settings aligning with the assumptions of the modeling technique. The next phase is one of the crucial stages, the evaluation phase. At this stage, each model is evaluated with the primary focus on the evaluation criteria and in contrast to the business objective. The models are then assessed of their merits and demerits.

**3.1 Business Understanding**

Human activity recognition, as seen from the previous chapter, has significant business value. The primary motive of this study is to enable patients, senior citizens or infants with immediate medical attention during a case of physical accident or emergency. This can be ensured by detecting their current physical activity with the assistance of a wearable device connected to them. As it is a case of providing medical support, recognizing the right activity of the user is of utmost importance as it acts as a trigger to the chosen emergency action as notifying the guardian or the hospital. From an analytical perspective, to recognize the activities right, we must target onto high accuracy levels of prediction. To achieve the targeted accuracy levels, multiple machine learning and deep learning models, as discussed in Literature review will be employed. The study assumes that with the increase in accuracy of the activity prediction, medical assistance can be improved proportionately.

A significant constraint to note at this level could be the computing capabilities. The machine used for this project is a 64 - bit Intel i5 processor with 8GB of RAM. The coding will be done using R Language in R Studio of version 1.1.442.

Business Objective –

As the current study deals with enabling immediate assistance to render emergency services, they must be reported with high accuracy. This situation requires ensuring high importance given to false positives and false negatives since an undetected change in activity could turn fatal to the user, or a false alarm could cause unnecessary panicking resulting in loss of valuable time, money and other resources. So the objective of the experiment is to implement machine learning models with high classification accuracy.

Business Success Criteria –

The solution must not only result in high classification accuracy but also provide evidence in proving that the experiment and results are significant and would always yield similar results when attempted to replicate the solution.

Given the above objectives, assumptions and constraints, below hypothesis is utilized to address the research question –

Deep learning multi-layer perceptron neural network gives better accuracy than all the supervised machine learning models which are modeled on human activity recognition dataset.

**3.2 Data Understanding**

The dataset used in the current study is generated at the International Workshop of Ambient Assisted Living (IWAAL) held in Spain in 2012. Anguita et al. (2012) designed an experiment by recording a set of six physical activities performed by a group of 30 volunteers. The tri-axial linear acceleration and the tri-axial angular velocity from the built-in accelerometer and gyroscope of a smartphone device are captured. These sensor signals were processed to remove noise and were sampled for every 2.56 seconds with a 50% overlap of the fixed width sliding window. The resulting signals had a combination of gravity and body motion components and hence were passed into a low pass filter to obtain separated components with the gravitational force components cut oﬀ at the lower end of the filter. Time and frequency domain vector of features were obtained from each of the windows created. No cognitive activity, a perception of a load was gathered whatsoever (Longo, 2017).

|  |  |
| --- | --- |
| **Raw Signal** | **Deﬁnitio** |
| tBodyAcc-XYZ | Body acceleration in time |
| tGravityAcc-XYZ | Gravity acceleration in time |
| tBodyAccJerk-XYZ | Jerk in body acceleration in time |
| tBodyGyro-XYZ | Body gyroscope measure in time |
| tBodyGyroJerk-XYZ | Jerk in body gyroscope measure in time |
| tBodyAccMag | Magnitude of body acceleration in time |
| tGravityAccMag | Magnitude of gravity acceleration in time |
| tBodyAccJerkMag | Magnitude of jerk in body acceleration in time |
| tBodyGyroMag | Magnitude of body gyroscope measure in time |
| tBodyGyroJerkMag | Magnitude of jerk in body gyroscope measure in time |
| fBodyAcc-XYZ | Body acceleration in frequency |
| fBodyAccJerk-XYZ | Jerk in body acceleration in frequency |
| fBodyGyro-XYZ | Body gyroscope measure in frequency |
| fBodyAccMag | Magnitude of body acceleration in frequency |
| fBodyAccJerkMag | Magnitude of jerk in body acceleration in frequency |
| fBodyGyroMag | Magnitude of body gyroscope measure in frequency |
| fBodyGyroJerkMag | Magnitude of jerk in body gyroscope measure in frequency |

Table 3.1: Description of raw signals from HAR experiment

Note: The 'XYZ' denotes the three axis directions X, Y, Z for each of the tri-axial signals; 't' indicates time domain variables and 'f' denotes frequency domain variables.

The features extracted from the experiment have been derived from an elaborate process and are tabulated in table 1. The initially obtained raw signals, tAcc-XYZ, and tGyroXYZ were the tri-axial signals obtained from the accelerometer and the gyroscope, and the total acceleration was further split into tBodyAcc-XYZ and tGravityAcc-XYZ. The Jerk signals, which are the rate of change in acceleration over time are derived from the raw signals and are denoted as tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ. Euclidean norm is performed to calculate the magnitude of each of the signals further resulting in components as tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag. Additionally, a Fast Fourier Transform was applied to produce the features as fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyroXYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. Table 2 lists the set of explanatory variables calculated for each of the above raw features.

|  |  |
| --- | --- |
| **Descriptive** | **Deﬁnition** |
| mean() | Average value |
| std() | Standard deviation |
| mad() | Median absolute deviation |
| max() | Maximum value |
| min() | Minimum value |
| sma() | Signal magnitude area |
| energy() | Energy value |
| iqr() | Interquartile range |
| entropy() | Signal entropy value |
| arCoeﬀ() | Autoregression coeﬃcient |
| correlation() | Correlation coeﬃcient |
| maxInds() | Index of the largest magnitude frequency component |
| meanFreq() | Weighted average of the frequency component |
| skewness() | Skewness of the frequency domain signal |
| kurtosis() | Kurtosis of the frequency domain signal bands |
| Energy() | Energy of the frequency within the FFT of each window |
| angle() | Angle between the vectors |

Table 3.2: Description of derived variables from raw signals

Note: The angle is calculated only for gravityMean, tBodyAccMean, tBodyAccJerkMean, tBodyGyroMean, tBodyGyroJerkMean vectors.

Additionally, an identifier variable, 'Subject,' describing the user who carried out the particular activity is provided. It is a 30 factor categorical variable with labels from 1 to 30 each representing a volunteer experimenting. Finally, the feature describing the physical activity performed by the users during which the signals are collected is recorded as 'Activity.' It is a 6-factor specific feature. The activities performed and recorded are tabulated in 3.

|  |
| --- |
| Activity ID |
| Laying |
| Standing |
| Sitting |
| Walking |
| Walking\_Upstairs |
| Walking\_Downstairs |

Table 3.3: Activity list: Classes of a target feature

**3.2.1 Data Exploration**

The first step in exploring the data is to view the dimensions of the dataset and to verify with the data source to ensure the data is intact and have the right number of dimensions. After validating the data integrity, each of these features has to be understood to proceed to the next stages of the experiment. The simplest way to perform this is to evaluate the structure of the entire dataset. It is important to review with the structure of data as investigated from the business understanding stage to validate the structure of every feature obtained after the data import process. If any discrepancies have been found, the features can be translated to the required format in the next stage.

Reviewing the summary statistics is the simplest way to analyze and obtain an overview of the features. The summary statistics would have diﬀerent elements of information for diﬀerent types of features. As for specific features, the summary statistics illustrate the cardinality of the feature, and for numerical features, they detail the mean value of the feature along with median and other fixed quantile values. The categorical features, dependent and the independent variables, combined must be further evaluated for categorical count or percentage within their categories. This enables the study to identify the imbalance in the cardinality between the feature under inspection and the target feature.

Ideally, the next step is to visualize the data and to understand its aspects which cannot be observed otherwise as distribution and normality. Outliers for each feature can also be visualized to gain better insights. The relationship between multiple attributes could be visualized and understood better. Simple aggregations can also help to understand the data at a macro level. Analyzing the sub-populations and their properties would help in getting an overview of any specific feature. Additionally, data quality can also be examined at this stage by understating the outliers, missing values, noisy data and also the inconsistent data. Each of these can have a diﬀerent test performed to understand the issues and to devise a plan to resolve them in the next stages. Redundancy can also be identified at this stage.

In conclusion, the current problem is identified as a Classification task. From the data understanding phase, the dataset has been identified with a total of 563 features, with 561 independent numerical features, one categorical feature describing the target, Activity label and another categorical feature describing the user. Summary statistics with a combination of visualizations and other explorations would be a part of this section forming the data exploration report. Similarly, the quality of the data can also be judged and reported.

**3.3 Data Preparation**

Data preparation is one of the most crucial stages in a data mining project. This process determines the quality of the data used to extract insights which could, in turn, aﬀect the quality of the insights. It is the process of manipulating the data and prepares it to be suitable for the next stages of the project. Data emerging from this stage must not contain any incomplete, noisy or inconsistent data. The primary tasks in data preparation are as follows –

• Data cleaning- The primary task following data cleaning is to evaluate the missing data in the entire data set. A data point can be said missing data if there is either no data value stored in it or values as N/A, not applicable, or 999 are stored instead of a blank field. Having missing values is a common occurrence and can have a significant eﬀect on the feature and insights are drawn from it. Once the missing data is identified, it is crucial to know the mechanism behind it. Outlier detection and analysis are also the part of data cleaning.

The final task of the data cleaning stage is to eliminate data redundancy. Having a large amount of redundant data may distract the model from concentrating the important observations. Furthermore, redundant data may also slow down the entire process. Redundancy may be detected by performing a correlation between the features which provides the relation between one features over the other. The most popular correlation metric is the correlation coeﬃcient. A correlation coeﬃcient closer to positive or negative one can be treated as a strong relation, in which case, one of the features can be dropped and be represented by the other feature.

• Data Integration Data integration is the process of combining data from multiple sources. It is an important process and can aid in increasing the speed of the experiment and accuracy of the ﬁnal prediction results. While performing integration, it is crucial to identify features that need to be matched and combined. The structure of such features has to be understood well along with their functional rules, referential constraints and dependencies which must match for both the integrated features and their datasets, if they fail to have similarities, the data and its conditions must be transformed to make required changes. Redundancy tests performed in the data cleaning stage must be evaluated to perform the better integration. By the end of this stage, we must be able to detect a conflict between several values due to inequality in scaling, encoding or representation and thus provide an appropriate resolution.

• Data Transformation- The data are transformed into another form by applying a mathematical function, so the resulting value is more suitable for the next stages of the data mining project. Data transformation utilizes a function that maps all the values of the feature to a new set of transformed values. There are diﬀerent ways of transforming the data. An ideal method shall be devised based on the experiment and the data.

– Aggregation is a technique when summarization is applied to the data. This transformation also enables in creating a data cube, which is a 3-D range of values which are aggregated from diﬀerent aggregation and abstraction patterns. Aggregation can help when each observation can be better represented as a single aggregated value as opposed to multiple values partially explaining the same feature.

– Normalization is the technique of ensuring that the data falls within a fixed range of values. It is performed since the measurement of a unit can have a significant eﬀect on the final result. So to avoid dependency on the measuring unit, each value in a feature is scaled using the maximum - minimum or mean - standard deviation of all the values in the feature. Normalization tries to provide each feature the same amount of importance and avoids the algorithm to be influenced by higher values of a feature with lesser importance.

– Feature construction is an important part of the transformation of adding new features which can help in understanding the data better by providing a better meaning and improved accuracy. A new feature must be constructed such that it helps in improving the accuracy, computational eﬃciency, and can be generalized to diﬀerent algorithms. However, there is always a risk of overfitting to the problem at hand while constructing new features which must be monitored. (Sondhi, 2009)

• Data Reduction Data reduction is the technique of reducing the overall representation of the data while having minimal loss of domain knowledge and information content. Data reduction helps in decreasing the computational complexity of the algorithms owing to the complex data and its structures which in turn result in quicker analysis. These include methods such as sampling, feature selection, and dimensionality reduction.

– Sampling is the technique of selecting a subset of observations to represent and be analyzed instead of the entire data. It can be used when the data held is massive, and computation can be expensive and time-consuming. The notion of sampling is to obtain a representative sample of the entire data. Types of sampling are random sampling, stratified sampling, cluster sampling and systematic sampling which can be used to derive samples from an imbalanced dataset.

– Curse of dimensionality as stated by Bellman (2013) is the 'problem caused by the exponential increase in volume associated with extra dimensions added in the Euclidean space.' Dimensionality reduction and Feature selection are important techniques that help avoid the curse of dimensionality. Attribute subset selection is one of the methods of dimensionality reduction which detects and stores highly relevant features. The principal component analysis is a similar dimensionality reduction technique, which searches for a set of n-dimensional vectors that can best represent the data over the original feature set. The new set of vectors created is ranked based on the amount of information and variance it represents, so important information still intact.

• Data Split- The final step in data preparation is to create training and test datasets. Each algorithm will be trained on the training dataset and will be evaluated against the testing dataset. The typical data split performed 70 to 30 split between the training and test datasets. Also, it is important not to expose the test data set during the training process. Typically a stratified random sampling is an ideal choice to ensure that the composition of the data between the datasets matches to that of the original data source. (Liberty et al., 2016)

Once data preparation is completed after utilizing either one or more of the above techniques as per the original data conditions, quality and the experiment to be performed, the data must be Valid, Accurate, Complete, Consistent, and uniform and ready to be modeled.

**3.4 Modeling**

In this stage, the final machine learning algorithms will be utilized to induce predictive models and gain insights from the prepared data to solve the research question. Each machine learning model has to be finely tuned for various variables and parameters to have the perfect model synced with the data. The literature review has identified multiple classification algorithms typically used in machine learning and HAR domain.

**Decision tree**- Under the information based family of machine learning, decision trees will be modeled. When modeling the decision trees, every parameter can be modified to create a classification model. The first parameter to be provided will be the target and the selected independent variables. Another critical parameter is the splitting criteria which can either take the Gini impurity or the information gain value to split the data into partitions. As the current study deals with a classification problem, the method must be specified as appropriate. There are several controlling parameters which may be used to enhance or restrain the tree growth; the best approach would be to perform a trial and error procedure to choose the suitable parameters. The resultant decision tree can be inspected in several ways as a graph of the decision tree plot, summary, cross-validation results.

**Random Forest**- Random forest is a user-friendly methodology built by using ‘randomForest’ function of ‘randomForest’ package and has two parameters: which are number (ntrees) of trees, i.e., builts 500 trees by default and the variables are randomly sampled as candidates at each split (mtry).

**Support Vector Machine** - To build the support vector model in the R interface, ‘e1071’ package is used, and ‘svm’ function is designed to fit the model. New data are predicted as usual after fitting the model, and both the matrix and the formula interface are implemented. As required for R’s statistical functions, the engine tries to be smart about the mode to be chosen, using the dependent variable’s type (y): if y is a factor, the engine switches to classification mode, unless, it behaves as a regression machine; if y is omitted, the engine assumes a novelty detection task.

**K-nearest neighbor algorithm** - For the K-nearest neighbor algorithm, the independent variables and target feature must be specified to help the algorithm diﬀerentiate. The critical aspect of the KNN algorithm is the 'K' value, which is the number of neighbors to consider. The experiment must be carried out with a 'tune length' set to various values of 'K' to find the one that fits in best. The default distance used in KNN is the Euclidean distance. The final value of the target is decided by the majority vote of the K nearest neighbors regarding the Euclidean distance.

**Naive Bayes algorithm** – The Naive Bayes algorithm implementation takes in the independent and the dependent features as the first parameters. The algorithm can work with the data with default parameters alone; however, multiple other parameters may be provided to customize it better. Laplacian correction can be used to smooth the categorical data which can be provided with a numerical value, the distribution type can be altered by editing the boolean value of 'use kernel,' and the total bandwidth can also be adjusted.

**Multinomial logistic regression** – Multinomial logistic regression can be implemented by using a penalized logistic regression technique. The critical parameter is the MaxNwts which must be set high enough for the algorithm to function. The summary of the logistic regression gives out the coeﬃcient and the standard error for each of the independent features. Additionally, it also highlights significant values out of all the independent features.

**Artificial neural networks-** Artificial neural networks are complex structures that can be easily mapped to the number of different datasets. The size parameter plays a significant role in specifying the number of units in a hidden layer, and it will take a single or a series of numbers to run the model. The best way to choose the number of hidden units as suited by the data is to run multiple experiments and identifying the right number. The decay parameter is a regularization parameter which penalizes the model and helps avoid over-fitting. Although neural nets can be highly customized, they can be computationally expensive and also cannot be interpreted as well as other machine learning algorithms.

**Linear Discriminative analysis-** Linear Discriminant Analysis is a known machine learning algorithm for classification. This model is interpretable, and the prediction process is quite simple which makes it stand out from other machine learning techniques. When the dependent and independent variables are inputted to the LDA algorithm, it converts it into a matrix. It creates the linear boundaries to divide the data into their respective categories. The model also predicts the category of the unknown data by learning. The linear boundaries are the result of considering that the predictor variables for each category have the same multivariate Gaussian distribution. The linear discriminative algorithm can be modeled using 'lda' function which is a part of 'MASS' package.

**Deep learning-** For the implementation of deep learning multilayer perceptron model, ‘keras’ package is installed. As the interest in deep learning has been stimulating quickly in the past few years, and numerous deep learning frameworks have developed over time. Out of all the currently available frameworks, ‘Keras’ has outperformed for its productivity, flexibility and user-friendly API. At the same time, ‘TensorFlow’ has risen as a next-generation machine learning platform that is both incredibly flexible and well-suited to production deployment. It has a very user-friendly Application platform interface which makes it accessible to model deep learning models instantly.

It also possesses built-in support for recurrent networks, convolutional networks, and any combination of both. ‘Keras’ package supports arbitrary network architectures like multi-input or multi-output models, layer sharing, model sharing. It is also capable of working on top of multiple back-ends including TensorFlow, CNTK, or Theano.

**3.5 Summary of Design**

This chapter has produced a detailed design of the entire experiment to be conducted. Initially, the objective of the study is understood well along with the restrictions and assumptions. The data is then examined to provide an initial data quality, data description, and data exploration reports. Data is then prepared by performing many of the data quality improvement operations. Data transformation techniques are then applied to create a final dataset. The classification accuracies of various machine learning and deep learning algorithms are computed. The model with the highest accuracy is identified. One potential problem of the design is the computational complexity due to the vast size while feature engineering and modeling the algorithms, hence generating each model could be a time-consuming approach.

**4 Implementation, Results, Analysis, Evaluation & Discussion**

This section describes the execution procedure, the results of the experiment, a detailed analysis of the experiment and the results obtained from the design implementation. This section also aims to implement the experiment in the direction of being able to answer the research questions. The results are obtained from the predictive capabilities of each of the models induced based on algorithms using data preparation methods like Principal Component Analysis. Each of these results will be discussed individually before commenting on the general implementation and design strategy. The section will then conclude by stating the strengths and limitations of the experiment.

**4.1 Data Understanding**

To understand the data and its elements, the data must be initially integrated and put in an ideal format to inspect it; this process of data integration is performed and specified in the data preparation stage. The first element of investigation performed on the integrated data is calculating the dimensions using the function 'dim.' There were 10299 rows, which is the train and test set combined and 563 columns consisting of 561 features, the subject, and the activity variable. The names of each of the columns are evaluated to ensure the correct order of the data.

The structure of the dataset was attempted to be analyzed, but as the data size was high, it was not entirely understandable. The six-point summary of the data, which includes the minimum, maximum, mean, median and the 1st and third quartiles are generated for each of the features. These did not add value either. The attributes of the data: names of the columns, rows and the class of the dataset are viewed. To have a quick view of the data, the first two rows of the data are observed using the head function. To understand the composition of the subject and the Activity levels, their frequencies were tabulated, the figures 4.1 & 4.2 are the histograms of the Activity and the subject variables. It is observed that both the fields were free of any class noise or data inconsistency issues.

Unique rows were analyzed to see for any presence of an attribute or class noise, but there were no signs of such errors. The maximum and minimum values of the 561 time and frequency domain variable were seen to be +1 and -1 respectively which stated that the values were normalized between this range while creating the dataset. There were no missing values found under the missing value analysis test.

The users and the activity levels were tabulated to inspect their distribution using simple aggregation methods. It is observed that their composition was quite uniform, for each other and the entire data.

The Data Understanding phase proved to be quite insightful in establishing the context of the data preparation to be performed or to estimate the complexity regarding space and time for the creation of the models as well as predicting the final results. The initial analysis of the data depicted that the Activity feature which is the target variable for the experiment is well balanced as seen in figure 4.1. Furthermore, the subject, which is the identifier variable is also well balanced within itself as seen in figure 4.2 as well as when contrasted with the target feature, seen in 4.3. This eases the case of creating train and test datasets without requiring any stratified sampling to ensure the composition of each of the data splits which will be analogous to the original data composition.

It is also seen that all the independent features fall in a specific range of +1 to –1. It can be understood that the data was range normalized to fit this particular range. It permits the algorithms to provide equal emphasis on every feature and allow better-standardized data into the next stages and not allowing various ranges of the data to aﬀect the significance of any feature.

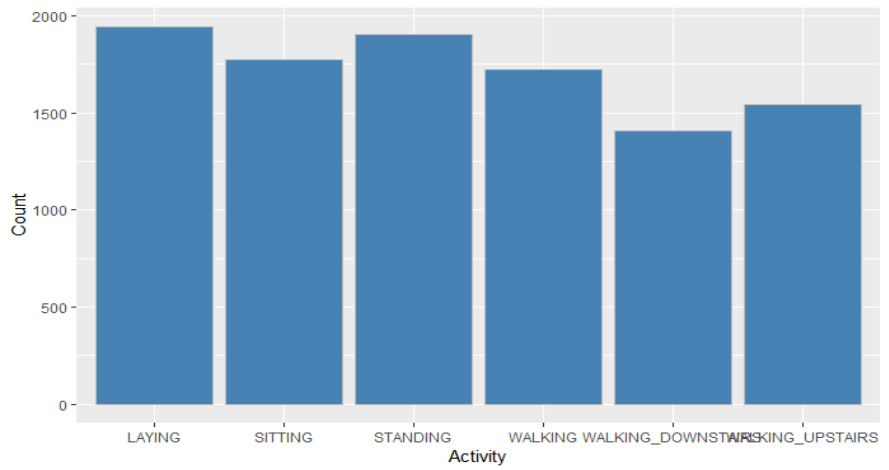


Fig 4.1: Histogram of the target variable – ’Activity’

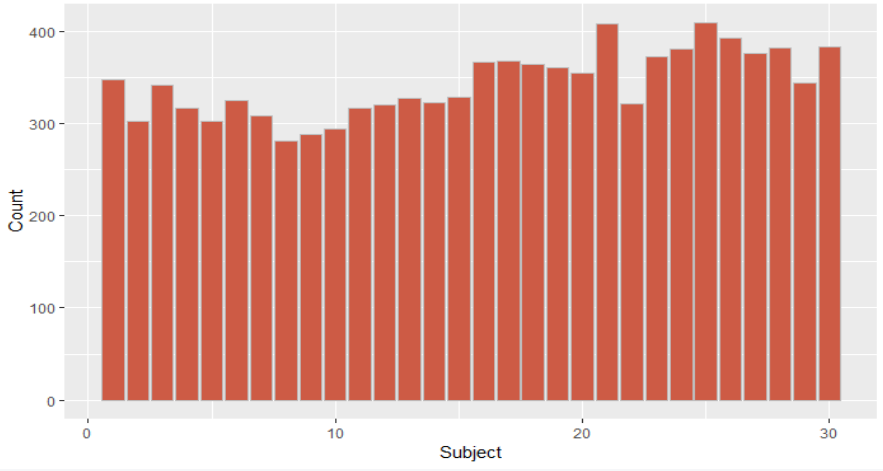


Figure 4.2: Histogram of identiﬁer variable – ’Subject’

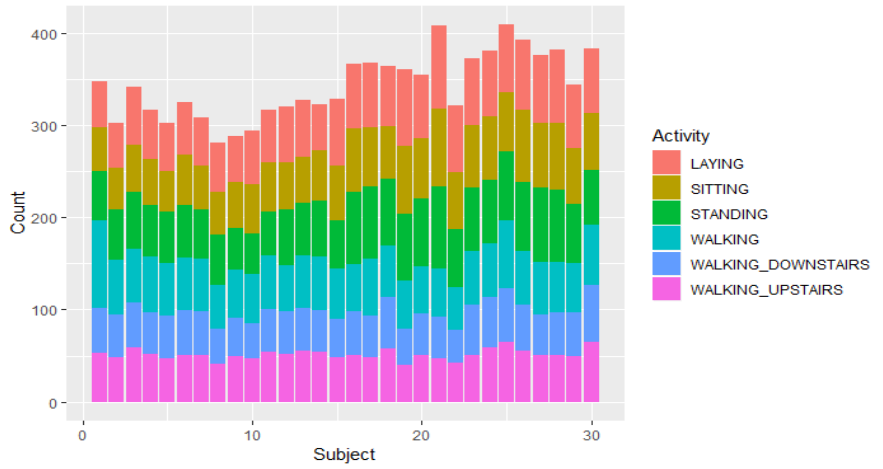


Figure 4.3: Histogram of records per user grouped by target

**4.2 Data Preparation**

• Data Integration- The first step under preparing the data is the data integration which is performed before the data understanding stage as the original data is provided as a csv file consisting for two separate files for training the testing data. This process of completely ignoring users from being a part of the dataset can be misleading. Hence, both the datasets are merged to create the complete "data" dataset.

• Data Manipulation The next step under the data preparation is converting the data into their right primitive data types. The Activity and Subject variable are converting into categorical variables using the function "as.factor." All the other independent features are already numeric. Outlier analysis and distribution tests were not performed as per the design solution as they can be hard both for computation and analysis, also as the features were previously normalized to a fixed range, there could be no further issues.

• Dimensionality Reduction High dimensionality as seen previously can cause severe diﬃculties as it could be increasingly hard to visualize and understand the data, it can also be computationally complicated and expensive. An ideal solution to this problem is to use dimensionality reduction techniques to bring down the size of the data. This study utilized the Principal Component Analysis technique to decrease the size of the feature set.

This technique replaces all the set of features with a linear combination of them called as principal components. To generate the principal components of the feature set, the 'prcomp' function is utilized. The input to the function is the full list of features without the subject and the activity columns and a boolean variable of scaling given as true, to ensure that the features are scaled to have unit variance. The resultant object has five elements as standard deviation, rotation, center, scale and a matrix with the new components which are ordered by decreasing importance and relevance levels concerning the variability being captured. The individual component variance measured is calculated by squaring the standard deviation values. Using the first 100 features, about 94.6% of the total variance is captured which is enough data captured for the proportion of data reduced as shown in the figure. 4.4 below.

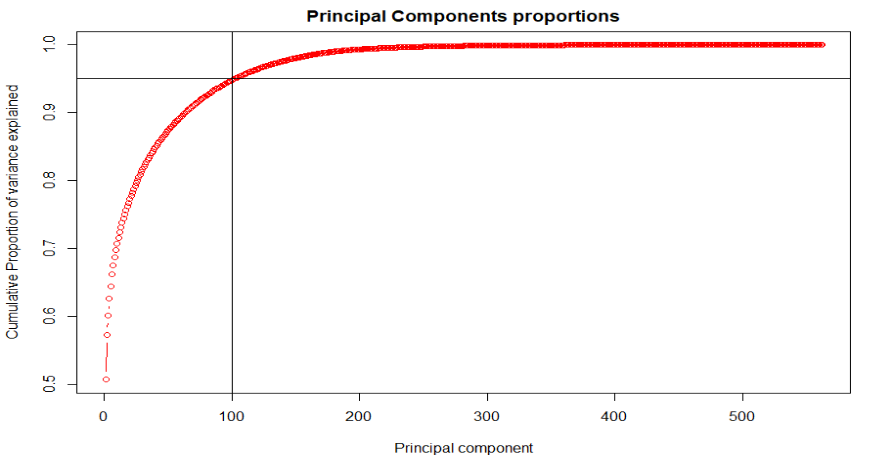


Figure 4.4: Proportion of Variance captured by top 100 components of PCA

**4.3 Modeling**

The model validation techniques must implement the models. To perform the cross-validation, the data should be split into training and testing dataset with the split ratio of 7:3. The Activity variable is given as an input to the function, the percentage of values to be generated is set as 0.7, which is the percentage of the training set against the test set.

Train datasets are created by extracting the records corresponding to the split. All the other rows which correspond to the remaining 30% of the total records are placed into the test datasets. The dataset created with the 100 principal components will be used to prepare train and test datasets. Once the train and test datasets corresponding to each of the manipulated datasets are created, modeling can be performed.

**Support vector machine model:**

The first model created is the support vector machine with 'svm' function with default settings.

**Decision tree model:**

A decision tree with the splits being performed with the help of 'rpart' function which is a part of 'rpart' package, using the Gini impurity value as it is faster and works better for continuous attributes as in this scenario. Gini impurity also works in accordance to minimize the misclassification rate, which is quite suitable for the study.

**Random Forest model:**

Random forest involves three steps: At first, a tree is drawn with 'ntree' bootstrap samples, then for each bootstrap sample, grows an 'unpruned' tree by choosing best split based on random sample of 'mtry' prediction at each node. Finally, new data is predicted using majority votes for classification.

Random forest uses 'randomForest' package which is a part of 'randomForest' function.

randomForest(formula = activity ~ ., data = train.data)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 9

OOB(out of bag) estimate of error rate: 5.62%

Here, by default 500 decision trees are built, and nine variables are utilized at each split (square root of no. of features which is sq. root of 100 PCA variables). Moreover, also, OOB (out of bag) estimate, which is a method of measuring the prediction error of random forests is 5.62%.

**K-Nearest neighbors' model:**

For K-Nearest neighbors' algorithms, the tuned length been set to '7', which seems an ideal solution to compensate between the increasing value of accuracy and tuning and execution time of the model. To iterate the training procedure over K, the model used cross-validation technique.

**Naive Bayes model:**

For the Naive Bayes, the default settings with the use of 'naive\_bayes' function which is a part of 'naive\_bayes' package. It was the quickest execution out of all the models.

**Multinomial logistic regression model:**

The penalized multinomial logistic regression is also run on default settings using 'multinom' function which is a part of 'nnet' package, but the maximum weights have to be set high enough to enable the model to be executed.

**Artificial Neural Network model:**

The neural net is initially modeled with a combination of units of hidden layer and decay value. It is identified that by using 30 units in the hidden layers with a decay value of 0.1 the algorithm converges considerably faster and the model does not over-fit either.

**Linear discriminative analysis model:**

'lda' function which is a part of 'MASS' library is used for linear discriminative analysis. 'Activity' field is given as the dependent variable and the independent variables as predictors, after the training the model test data is inputted into the predict function which is a generic function.

**Deep learning multilayer perceptron model:**

Deep learning based on multilayer perceptron neural networks is one of the underlying algorithms among different deep learning methods. At First, input data is converted into the matrix using 'as.matrix' function. Then the dimension name is changed to 'null.'

All the independent variable are normalized using 'normalize' function which is a part of 'keras' package which uses the tensorflow package in the backend. The sampling of the data with replacement is carried out, and the data is split into 70:30 ratio for training and testing data. 'Activity' variable is set as the training and testing target for this model. One hot encoding is performed which is the process of converting the target variable into the categorical variable which is, in this case, 'activity' variable as shown in the figure below.

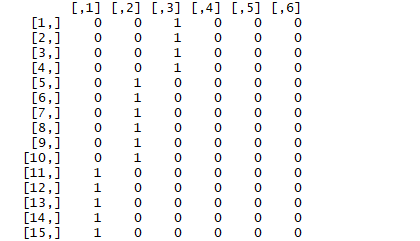


Figure. 4.5: One hot encoding on ‘Activity’

The next step is creating a sequential model using 'keras' package. A pipe function '%>%' is used to add the layers to the model and 'dense\_layer()' is used since the network is densely connected. Starting with 400 neurons in the first hidden layer and with 'relu' (Rectified layer unit) activation all the 562 variables are given as input. In the second layer, six neurons are used since there are six different activities to be classified. Here 'softmax' activation is used which helps to keep the range of the between 0 and one which can be used as probabilities.

There are 22520 parameters from the first layer and 2406 parameters from second layers and 227606 parameters in total as shown in fig. 4.6 below.

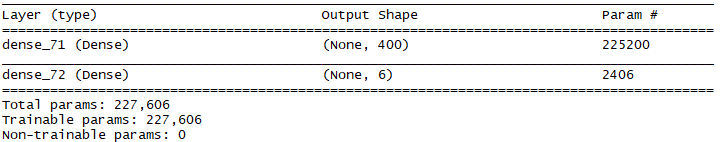


Figure 4.6: Total number of parameters used in deep learning model.

Then the model is compiled using:

loss = 'categorical\_crossentropy' for multiple category in the output variable.

optimiser = 'adam'

matrics = 'accuracy'

Fitting the model: multilayer perceptron neural network for multi-class softmax classification using 'fit' function.

history <- model %>%

fit(training,

trainLabels,

epoch = 200,

batch\_size = 32, # of samples used per gradient (default is 32)

validation\_split = 0.2) # 20 % of data for validation split.

epoch is 200 which means the model will run 200 times (200 iterations).

We can plot the results using the plot function as shown below.

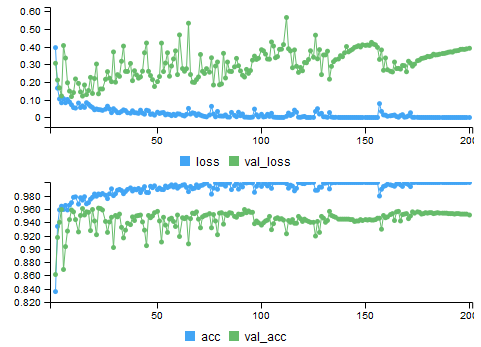


Figure 4.7: Accuracy plot for training and test data of deep learning model.

The first plot is loss plot, the blue line is for loss based on the trained data, using 20 % of validation data. Ideally, when there is a decline in training loss, there has to be a decline in validation loss in a good model, but the opposite indicated the overfitting in the model. Then comes the evaluation phase using test data and test labels. After executing this code the loss and accuracy of the model are acquired. Lower the loss the better the model and higher the accuracy the better. Prediction is made using confusion matrix: a prediction is made on test data using 'predict\_classes' function.

**4.4 Evaluation**

The evaluation is performed by predicting the target values for the testing datasets using their corresponding models. The prediction is implemented using the 'predict' function, from the 'caret' package, by supplying the model and the test dataset to it.

Removing the original target feature from the test dataset is essential. Once the predictive model results are generated, a confusion matrix is plotted using the function 'confusionMatrix' with data input as the predicted values and reference as the original target feature labels. The overall component of the confusion matrix generated has the overall statistics as accuracy levels, sensitivity, specificity, kappa value.

The accuracies of all the nine models are listed in below table:

|  |  |  |
| --- | --- | --- |
| No. | Model | Accuracy (%) |
| 1 | Decision tree | 73.49 |
| 2 | Naïve Bayes | 80.22 |
| 3 | Random Forest | 88.63 |
| 5 | k-nearest neighbor | 90.51 |
| 6 | multilayer logistic regression | 90.97 |
| 4 | Support vector machine | 93.85 |
| 7 | Linear discriminative analysis | 96.23 |
| 8 | Artificial neural network | 96.79 |
| 9 | Deep learning multilayer perceptron | 98.59 |

Table 4.1. Total accuracies of all the models.

The image 5.1 shows the composition of the target features in the entire dataset and also in each of the created train and test data samples.

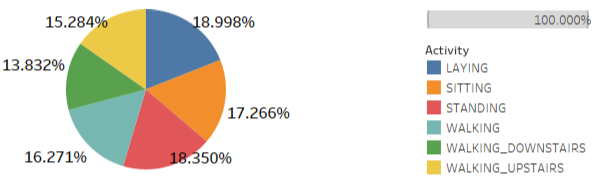


Figure 4.8: Target Feature Distribution

**Support vector machine evaluation:**

Support vector machine algorithm: Support vector machine algorithm is one of the best performing models when compared to the decision tree, random forest, and naive Bayes classifiers. Examining the nature of the activity, the two clusters of activities, stationary and mobile, the SVM algorithm ensured high inter cluster classification accuracy among stationary activities in comparison with mobile activities. However, the intracluster classification accuracy was not very satisfying, especially for the stationary activities. The confusion matrix of support vector machine model is tabulated in table 4.2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SVM | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 534 | 0 | 0 | 0 | 3 | 0 |
| Sitting | 4 | 419 | 63 | 0 | 4 | 1 |
| Standing | 0 | 29 | 500 | 1 | 2 | 0 |
| Walking | 0 | 0 | 0 | 477 | 19 | 0 |
| Walking\_downstairs | 0 | 0 | 0 | 3 | 409 | 8 |
| Walking\_upstairs | 0 | 0 | 0 | 21 | 23 | 427 |

Table 4.2: Confusion matrix for support vector machine.

**Decision tree evaluation:**

Models are created using the appropriate modeling technique and settings to yield various predictive results. The first model implemented was the decision tree, which is one the most robust algorithms and yet can be profoundly influenced by its structure and thus yield drastically diﬀerent results. Table 4.3 exhibits the predictive results for the decision tree. It can be observed that the decision tree has suﬀered in diﬀerentiating the activities from each other. Primarily, the sitting activity is repeatedly misclassified as standing and laying. One of the major flaws of the decision tree that could have impacted the performance of the algorithm over the given data could be the fact that the decision tree works better with categorical variables. The reason is quite intuitive that with categorical independent variables, it is easier for the algorithm to create rules to perform partitions in the data to develop a tree with leaf nodes stating the appropriate target feature. However, in the current case, as the independent features are all numeric, it can be quite challenging for the algorithm to find the right value that can be used to divide the tree to create partitions. Additionally, the decision trees are sought out due to its ability to provide highly interpretable results but in this case, with such vast amounts of data features, interpreting the rules, and the tree partitions could add insufficient value.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Decision tree | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 450 | 84 | 2 | 0 | 0 | 1 |
| Sitting | 1 | 234 | 254 | 0 | 0 | 2 |
| Standing | 0 | 71 | 461 | 0 | 0 | 0 |
| Walking | 0 | 0 | 0 | 459 | 29 | 8 |
| Walking\_downstairs | 0 | 0 | 0 | 135 | 235 | 50 |
| Walking\_upstairs | 0 | 0 | 0 | 120 | 24 | 327 |

Table 4.3: Confusion Matrix for Decision Tree

**Random forest evaluation:**

Random forest performs the classification task far better than the decision and results in increased accuracy around 15% which is quite an improvement. Table 4.4 exhibits the predictive results for the random forest algorithm. Primarily, the laying activity is classified with 99% accuracy. Secondly, the standing activity is repeatedly misclassified as sitting. There is also quite a lot of misclassification among walking, walking \_upstairs and walking\_downstairs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Random forest | References | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 514 | 23 | 0 | 0 | 0 | 0 |
| Sitting | 2 | 368 | 120 | 0 | 0 | 1 |
| Standing | 0 | 29 | 503 | 0 | 0 | 0 |
| Walking | 0 | 0 | 0 | 476 | 19 | 1 |
| Walking\_downstairs | 0 | 0 | 0 | 67 | 310 | 43 |
| Walking\_upstairs | 0 | 0 | 0 | 22 | 10 | 439 |

Table 4.4: Confusion Matrix for Random forest

**K nearest neighbors evaluation:**

The next model generated is the K nearest neighbors algorithm. It was one of the models that have performed remarkably in producing results with better accuracies. The table 4.5 depicts the confusion matrix for the K-nearest neighbor algorithm. It is seen that the algorithm has decidedly fewer misclassifications on the whole. Considering the nature of the activity, the two clusters of activities, stationary and mobile, the KNN algorithm ensured high inter cluster classification accuracy.

However, the intracluster classification accuracy was not very satisfying, especially for the stationary activities. The sitting and standing activities were highly misclassified. As the current experiment involves detecting the user activity over the span of 2.5 seconds, it is evident that the actions preceding and following it influences the activity. So the KNN algorithm which takes into

account the specified number of neighbors for performing classification justifies its satisfactory performance. However, it must be noted that the modeling and evaluation of the algorithm were computationally very demanding and had taken the highest amount of time.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| KNN | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 565 | 3 | 0 | 0 | 1 | 1 |
| Sitting | 11 | 459 | 46 | 2 | 14 | 13 |
| Standing | 3 | 70 | 525 | 10 | 34 | 14 |
| Walking | 0 | 0 | 0 | 500 | 25 | 12 |
| Walking\_downstairs | 0 | 0 | 0 | 1 | 324 | 2 |
| Walking\_upstairs | 4 | 1 | 0 | 3 | 23 | 421 |

Table 4.5: Confusion Matrix For K Nearest Neighbors

**Naive Bayes evaluation:**

The Naive Bayes model is a simple generative model that performed an indirect computation of the required probability through the Bayes function. Table 4.6 shows the confusion matrix of the Naive Bayes algorithm. The classification accuracy is not noteworthy compared to the results from other models and is also inadequate. The natural assumption of parameter independence can imply the poor performance of the Naive Bayes.

In the current experiment, each of the features is a derived attribute from the collected initially metrics and hence neglecting the interactions between the features can result in loss of plenty of information and prove damaging. However, it must also be noted that the algorithm was the fastest to converge compared to all the other algorithms.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Naïve bayes | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 504 | 12 | 0 | 0 | 21 | 0 |
| Sitting | 5 | 303 | 158 | 1 | 24 | 0 |
| Standing | 3 | 31 | 470 | 10 | 14 | 4 |
| Walking | 0 | 0 | 0 | 434 | 50 | 12 |
| Walking\_downstairs | 0 | 0 | 0 | 101 | 241 | 78 |
| Walking\_upstairs | 0 | 0 | 0 | 21 | 38 | 412 |

Table 4.6: Confusion Matrix for Naive Bayes

**Multinomial logistic regression evaluation:**

The multinomial logistic regression model was aimed to act as a baseline and was expected to provide mediocre results. However, the table 4.7 can showcase the eﬃciency of the algorithm for the given data. The logistic regression was one of the best performing models and provided exceptional classification accuracy. The ability of the algorithm to handle any nonlinear eﬀects and reduce the influence of noise could have been the factors corresponding to the increased predictive accuracies. It can be observed that there is low inter cluster misclassification by the model but some amount of intracluster misclassification for the stationary activities. The model also took plenty of time to converge but was more eﬃcient compared to the time complexity of the KNN algorithm. Additionally, the virtue of the algorithm of not expecting the data to adhere to normality and linearity assumptions has proved favorable to the experiment.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Logistic regression | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 526 | 0 | 11 | 0 | 0 | 0 |
| Sitting | 0 | 406 | 82 | 0 | 0 | 3 |
| Standing | 0 | 29 | 502 | 1 | 0 | 0 |
| Walking | 0 | 0 | 4 | 485 | 4 | 3 |
| Walking\_downstairs | 0 | 0 | 8 | 17 | 370 | 25 |
| Walking\_upstairs | 0 | 8 | 0 | 65 | 6 | 392 |

Table 4.7: Confusion Matrix for multinomial logistic regression

**Linear Discriminative analysis evaluation:**

Linear Discriminative analysis is the second best classifier in this study. In this model, interclass classification among mobile activities is higher when compared to all the models mentioned above. Laying activity has 100% classification accuracy. However, there is a slight misclassification among standing and sitting activity which affects the overall classification accuracy of the model. Table 4.8 shows the confusion matrix of the Linear Discriminative analysis algorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LDA | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 537 | 0 | 0 | 0 | 0 | 0 |
| Sitting | 0 | 434 | 22 | 0 | 0 | 0 |
| Standing | 0 | 56 | 510 | 0 | 0 | 0 |
| Walking | 0 | 0 | 0 | 490 | 1 | 11 |
| Walking\_downstairs | 0 | 0 | 0 | 0 | 405 | 0 |
| Walking\_upstairs | 0 | 1 | 0 | 6 | 14 | 460 |

Table 4.8: Confusion Matrix for multinomial linear discriminative analysis

**Artificial neural network evaluation:**

The final individual algorithm was the artificial neural net. As seen in section 2, the ANN algorithm was highly credited in the HAR task. The ability of the ANN to implicitly detect complex nonlinear relationships and also to acknowledge and utilize the interactions amongst the independent variables are two of the most significant advantages of the algorithm that have permitted in capturing the essence of the data resulting in the high accuracy values. Table 4.9 presents the confusion matrix of the ANN model. It can be seen that the mobile activities have been classified exceptionally well and so was the lying down activity with almost 100% classification accuracy. However, standing and sitting activities were slightly misclassified, and the reason is unknown as the algorithm is a black box and cannot be used to interpret. Additionally, the ability of the algorithm to generalize well due to its associative memory ensures that the classification results for all the diﬀerent samples of data are quite similar to each other.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ANN | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 580 | 1 | 0 | 0 | 0 | 0 |
| Sitting | 3 | 499 | 37 | 0 | 0 | 0 |
| Standing | 0 | 31 | 533 | 0 | 0 | 0 |
| Walking | 0 | 0 | 0 | 512 | 5 | 8 |
| Walking\_downstairs | 0 | 0 | 0 | 2 | 413 | 4 |
| Walking\_upstairs | 0 | 2 | 1 | 2 | 3 | 451 |

Table 4.9: Confusion Matrix for artificial neural network analysis

**Deep learning multi-layer perceptron evaluation:**

Deep learning is the best classifier model in this study obtaining 100% classification accuracy among laying and walking\_downstairs activity as shown in the table below. It results in an almost negligible amount of misclassification rate except for standing and sitting activity. The classification accuracy can be improved by adding the multiple layers to the model. However, too many layers can also result in misclassification. Table 4.10 shows the confusion matrix of the deep learning multi-layer perceptronalgorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Multilayer perceptron | Reference | | | | | |
| Prediction | Laying | Sitting | Standing | Walking | Walking\_downstairs | Walking\_upstairs |
| Laying | 586 | 1 | 0 | 0 | 0 | 0 |
| Sitting | 0 | 538 | 17 | 1 | 0 | 0 |
| Standing | 2 | 16 | 537 | 0 | 0 | 0 |
| Walking | 0 | 0 | 2 | 501 | 1 | 0 |
| Walking\_downstairs | 0 | 0 | 0 | 3 | 404 | 0 |
| Walking\_upstairs | 0 | 0 | 0 | 0 | 3 | 442 |

Table 4.10: Confusion Matrix for multinomial multilayer perceptron

**4.5 Hypothesis Evaluation**

The hypothesis of the current experiment is restated below –

H1: Deep learning multi-layer perceptron neural network gives better accuracy than all the supervised machine learning models which are modeled on human activity recognition dataset.

Earlier in this section, the results for each of the machine learning models created were discussed. To analyze the results of the first hypothesis, the results all the supervised machine learning and deep learning models are tabulated in table 4.1. From this table, it is evident that the current experimental results produced by deep learning multilayer perceptron is higher than all the other supervised machine learning algorithms.

With this null hypothesis can be rejected hence proving that the deep learning methods can perform better than all the well-known machine learning algorithms for human activity recognition dataset.

**4.6 Strengths & Limitations**

One of the main strengths of the study is the low standard deviation in the accuracy values for artificial neural network, linear discriminative analysis and K-nearest neighbor models. These three models had accuracies very close to best performer model and the variation in the results is quite small. This evidence suggests that the models have been developed with high robustness. It also signifies that these results can be replicated when necessary.

Additionally, all the models have shown a more significant understanding of inter-cluster diﬀerences by eﬀectively diﬀerentiating the stationary and mobile activities. This suggests that the models were able to capture the underlying concepts of the data to model it.

Limitation determined in the analysis is that most of the models were not able to capture the intracluster diﬀerences as all the models failed in diﬀerentiating sitting and standing activities which fall under stationary activities.

**5 Conclusion**

This section reviews the current study. It reemphasizes the research question and all the diﬀerent stages involved in clarifying it. The objectives of the research and all the essential phases are quickly walked through. Additionally, the contributions of the research are also stated. The section concludes by highlighting the areas of further research.

**5.1 Research Overview**

Primarily, the research intended to recognize the human physical activity of a user using the data generated by a wearable sensor device. The initial study was focused on understanding the current state of the art techniques in performing human activity recognition.

After acquiring relevant data for the research, the next central area of focus in the research was to perform suitable feature engineering to alter the data to be compliant to apply machine learning algorithms in the restricted time and memory conditions. The principal component analysis method was used to create latent features by capturing the underlying variance of the target feature. This technique proved eﬀective for the experiment as it minimizes the total number of feature to about 30% of the original features.

The goal of the research was to model the reduced data using machine learning algorithms and deep learning models to obtain high predictive accuracies.

**5.2 Problem Definition**

Based on the literature review, a gap in the current body of knowledge was exposed. Multiple studies on HAR have utilized the basic algorithms from the various families of machine learning and resulted in high accuracy values. However, the set of 9 algorithms were never used against feature reduction techniques to perform classification of activities on the dataset under inspection. The research work sought to empirically determine the better classification algorithm out of the nine most used ones applying the feature reduction techniques. The research question investigated in the study is:

Can deep learning model give the better classification accuracy of recognition of human physical activity with inertial sensor data compared to supervised machine learning algorithms?

\* Algorithms: Decision Tree, Random forest, K Nearest Neighbor, Support vector machine, Naive Bayes, Multinomial Logistic Regression, linear discriminant analysis, Artificial Neural Network algorithm.

**5.3 Workflow**

The following depicts the stages followed as an aim to answer the research question –

1. Performed extensive study on the existing literature of Human Activity Recognition. Gaps have been identified in the research domain.

2. A solution was designed to address the gaps in the Human Activity Recognition research. The primary motive of the design was to maximize the accuracy of the detection process.

3. The solution was implemented primarily based on the design. Additionally, occasional tweaking was performed where the solution commanded.

4. Future areas of research are identified to extend the field of study.

Multiple recommendations on the study have also been made by eﬃciently exploiting the various aspects of the knowledge acquired from literature review, the solution built consists of feature reduction technique identified from the literature review to be utilized. On the reduced dataset, the machine learning algorithms like the decision tree, Random forest, K nearest neighbor, Naive Bayes, logistic regression linear discriminative analysis, artificial neural network and the deep learning multilayer perceptron model is induced to generate the models for evaluation.

The evaluation metric chosen from the literature review is the Classification accuracy as it provides a comprehensive insight into the performance. The deep learning multilayer perceptron model performed the best in overall accuracy levels of 98.49%. The second best algorithm is the artificial neural network with an average of 96.79% then linear discriminative analysis model with 96.23% followed by support vector machine algorithm with 93.85%. The deep learning multilayer perceptron model has succeeded to perform better than all the other machine learning models with 98.59%. The rest of the models like logistic regression, naive Bayes, random forest and KNN have performed decently with the accuracies ranging from 80% to 90 %. However, the unsatisfactory performance was by the decision tree with an accuracy of 73.49%.

**5.4 Contributions & Impact**

The primary contributions to the body of knowledge from the current research are as follows

• A comprehensive literature review was performed which can help readers in understanding the domain and the current state of the art techniques in Human Activity Recognition.

• Systematic practical investigation of the quantitative properties of the Human Activity Recognition dataset was performed. This can be used as a baseline for future solutions using this dataset.

• Demonstrated that deep learning multilayer perceptron model performs best in specific cases of Human Activity Recognition.

• Demonstrated that Artificial Neural Network, Linear discriminative analysis, Multinomial Logistic Regression, and K-Nearest Neighbor algorithms can be used to model Human Activity Recognition

• Demonstrated PCA in the context of Human Activity Recognition

**5.5 Future Work & Recommendations**

There are multiple ways in which the study could have been manipulated which subsequently could have yielded in completely diﬀerent insights.The decision tree was modeled using the recursive partitioning method. There are plenty of other methods of decision trees that may be utilized to get better results. The splitting criteria provided in the current study is Gini impurity. Additionally, information gain may also be used. For the KNN algorithm, the tuned length was used a seven as it was seen that the increase in the K value only decreased the performance. However, it is not evident if the final k value of K was the local minima or the global minima of convergence. So a higher value of tune length could have helped resolve the ambiguity. For Naïve bayes classifier, the Laplacian correction and bandwidth adjustment values can be manipulated to observe the changes. For logistic regression, the study utilized a penalized multinomial regression, and there are multiple other models as bagged, boosted, regularized or ordered logistic regression. All the models have a diﬀerent protocol and tuning parameters which may be exploited.

In addition to PCA, exploratory factor analysis can also be tested as a dimensionality reduction technique. For evaluation, it was advised that K fold cross-validation could also be used for better confirmation of results. However, due to time constraints, it was not performed.

Another exciting aspect of investigation would be to divide the data into train and test for users. The models can be trained and evaluated precisely like the current experiment to observe if the models can generalize to unknown users.