HUMAN ACTIVITY PREDICTION USING LIFELOGGING DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GİZEM SARIARSLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

**HUMAN ACTIVITY PREDICTION USING LIFELOGGING DATA**

submitted by **GIZEM SARIARSLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ________________

Assist. Prof. Dr. Emre Akbaş
Supervisor, **Computer Engineering, METU** ________________

Prof. Dr. Adnan Yazıcı
Co-supervisor, **Computer Engineering, METU** ________________

**Examining Committee Members:**

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ________________

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU ________________

Prof. Dr. Adnan Yazıcı
Computer Engineering, METU ________________

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU ________________

Prof. Dr. Ahmet Coşar
Computer Engineering, University of THK ________________

Date:

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: Gizem Sarıarslan

Signature :

# ABSTRACT

## HUMAN ACTIVITY PREDICTION USING LIFELOGGING DATA



Sarıarslan, Gizem
M.S., Department of Computer Engineering
Supervisor : Assist. Prof. Dr. Emre Akbaş
Co-Supervisor : Prof. Dr. Adnan Yazıcı



September 2019, 83 pages



The lifelogging is an interesting new field with fast developing technology. People record every moment of their lives using wearable cameras, smartwatches and smartphone applications. The lifelogging composes this recorded data and contains multimodal data such as activity, body metrics, images and location. These data change depending on the activity of the person. In this study, the human activity learning is developed using lifelogs. The NII Testbeds and Community for Information Access Research (NTCIR) lifelogging dataset were selected. This dataset includes the lifelogging data from two users. It has multimodal data. The dataset includes activity, biometric, image, location and music data of two users. Biometric data from the dataset was studied for activity learning. Well-known algorithms such as Random Forest, Decision Tree, K-Nearest Neighbor, Naive Bayes, Support Vector Machine and Neural Network have been applied to the biometric data. Some state of art algorithms, such as Catboost, XGBoost, LightGBM, have also been applied on the biometric data for activity learning. In addition, some synthetic features were generated using mathematical operators on biometric data features, were trained and tested with applied algorithms. Furthermore, genetic algorithms have been used for feature selection on the dataset containing semi-synthetic features. Finally, two different late fusion methods were applied using the results from the image data and biometric data. The first is to select the activity with the highest class probability ratio from the results of the image and biometric data. The second is that the activity probabilities were learned separately from the images and biometric data. These scores were concatenated. Then, a new classifier was used on these data to learn the activity more accurately. As a result, our experiments show that classification based late fusion has a better prediction score than the first method. However, the two late fusion methods improved the prediction score from 89.08% to 95.29% for 3 classes and from 60.77% to 88.18% for 16 classes compared to the single learn model.

Keywords: Lifelogging, Machine Learning, Prediction, Activity Learning, Data Fusion

# ÖZ

## GÜNLÜK YAŞAM VERİLERİNİ KULLANARAK İNSAN AKTİVİTELERİNİN TAHMİNLENMESİ

Sarıarslan, Gizem
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Emre Akbaş
Ortak Tez Yöneticisi : Prof. Dr. Adnan Yazıcı

Eylül 2019 , 83 sayfa

Yaşam günlükleri, hızla gelişen teknolojiyle yeni bir ilginç alandır. İnsanlar, giyilebilir kameraları, akıllı saatleri ve akıllı telefon uygulamalarını kullanarak hayatlarının her anını kaydeder. Yaşam günlükleri, kaydedilen bu verileri oluşturur ve aktivite, vücut ölçümleri, görüntüler ve konum gibi çok modelli veriler içerir. Bu veriler, kişinin faaliyetine göre değişmektedir. Bu çalışmada, yaşam günlükleri kullanılarak insan etkinliği öğrenmesi gerçekleştirildi. NII Testbeds ve Bilgi Erişim Araştırması Topluluğu (NTCIR) yaşam kurucu veri seti seçildi. Bu veri kümesi, iki kullanıcıdan gelen yaşam boyu veriyi içerir. Çok modelli verilere sahiptir. Veri kümesi, iki kullanıcının etkinlik, biyometrik, görüntü, konum ve müzik verilerini bulundurmakta. Veri setindeki biyometrik veriler, aktivite öğrenmesi için çalışıldı. Biyometrik verilere Rastgele Orman, Karar Ağacı, K-En Yakın Komşu, Naif Bayes, Karar Destek Makineleri, Sinir Ağları gibi iyi bilinen algoritmalar uygulanmıştır. Catboost, XGBoost, LightGBM gibi bazı son teknoloji algoritmalar da aktivite öğrenmesi için biyometrik verilere uygulanmıştır. Ayrıca, bazı sentetik özellikler biyometrik veri özellikleri üzerindeki matematiksel operatörler kullanılarak üretildi, uygulanan algoritmalar ile eğitildi ve test edildi. Ayrıca, sentetik özellikleri içeren veri setindeki özellik seçiminde genetik algoritmalar kullanılmıştır. Son olarak, görüntü ve biyometrik verilerden elde edilen sonuçlar kullanılarak iki farklı geç füzyon yöntemi uygulanmıştır. Birincisi, görüntü ve biyometrik verilerin sonuçlarından en yüksek sınıf olasılık oranına sahip etkinliği seçmektir. İkincisi, faaliyet olasılıklarının görüntü ve biyometrik verilerden ayrı olarak öğrenilmiş olmasıdır. Bu puanlar birleştirildi. Daha sonra bu veriler üzerinde etkinliği daha doğru öğrenmek için yeni bir sınıflandırıcı kullanıldı. Sonuç olarak, deneylerimiz sınıflandırma temelli geç füzyonun ilk yöntemden daha iyi tahmin puanına sahip olduğunu göstermektedir. Bununla birlikte, iki geç füzyon yöntemleri, tekli öğrenme modeline kıyasla tahmin puanını 3 sınıf için 89.08%'dan 95.29%'a ve 16 sınıf için 60.77%'den 88.18%'e iyileştirmiştir.

Anahtar Kelimeler: Hayat Günlüğü, Veri Madenciliği, Aktivite Tahmini, Veri Birleştirilmesi

*To My Family*

## ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Adnan Yazıcı and co-advisor Asst. Prof. Dr. Emre Akbaş for their constant support and guidance. They always motivate me throughout my study. I gained valuable experience from them. It was a great honor to work with them for this interesting topic.

I would like to special thank my mother, my father and my sister for their endless support during this long period. From my childhood to till now, my family encourages me about my success. They are always with me and motivate me. They are really good guides. They remind me never to give up and what I could do, if I want. They always believe me. Thanks to them, every time I feel their support. I'm a lucky person to have this great family. Also, I would like to special thank Levent Koç for his motivation support.

These works are done using the dataset which is prepared and shared by Dr. Frank Hopfgartner and his team. I would like to thank Dr. Frank Hopfgartner. Also, I would like to thank Kader Belli for sharing her work result which is related to the dataset and is used in my thesis.

I would like to thank Gizem Nur Karagöz for motivational support. Lastly, I would like to thank my colleague, Selim Kutluay Tülek, at Havelsan for his support and understanding.

TABLE OF CONTENTS

## LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

# LIST OF ALGORITHMS

ALGORITHMS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| HR | Heart Rate |
| CHOL | Cholestrol |
| GSR | Galvanic Skin Response |
| BP | Blood Pressure |
| NTCIR | NII Testbeds and Community for Information access Research |
| KNN | K-Nearest Neighbors |
| LightGBM | Light Gradient Boosting Methods |
| XGBOOST | eXtreme Gradient Boosting |
| CatBoost | Category Boosting |
| XSD | XML Schema Definition |
| JAXB | Java Architecture for XML Binding |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation and Problem Definition

From past to present, people have recorded their daily events by painting and writing on to their diaries. With the rapid developing of technology, people have started to record their daily activities and behaviors by new smart devices such as smartphones, wearable cameras, and smartwatches. Each data recorded with these smart devices create personal lifelogs, and these lifelogs keep into personal digital archives [1].

The definition of lifelogging is a unified digital record containing the personal experiences of a user. This record is saved as multimodal data via digital sensors and kept in personal multimedia archive [2]. People create their lifelogging data with sensors on smart devices like smartphones, wearable cameras, and smartwatches. These smart devices record various data belonging to the user. For example, smartwatches and smartphones record the user's biometric, location, activity, and music information minute by minute. In addition, activity photos of the people at every moment are recorded via wearable cameras such as Google glass, necklace camera, body-mounted cameras (see Fig. 1.1 (a), (b), (c) and (d) for details). The combination of the data from these smart devices creates a large scale and multimodal lifelogging dataset.

The lifelogging dataset is of interest to data scientists because it contains the multimodal data and large scale data. Also, it has comprehensive information about humans. Many studies have been conducted with lifelogging data. People's daily activities are analyzed and learned by using the multimodal data on the lifelogging dataset. As shown in Figure 1.2, data from different sources are gathered and activities are determined. In this thesis, the main objective is to learn human activity

(a) (b) (c) (d)

Figure 1.1: Sample wearable cameras: (a) GoPro action camera. (b) Google Glass Camera. (c) Sensecam. (d) Fitbit smartwatch.

by understanding the relationship between activity and biometric data. It is sort of classification problem. We therefore apply well-known machine learning algorithms, as well as some state-of-art boosting algorithms to biometric data for the classifying activities. In addition, we used different approaches to obtain a better and more accurate result, such as generation of synthetic feature, the selection of features and the late fusion.

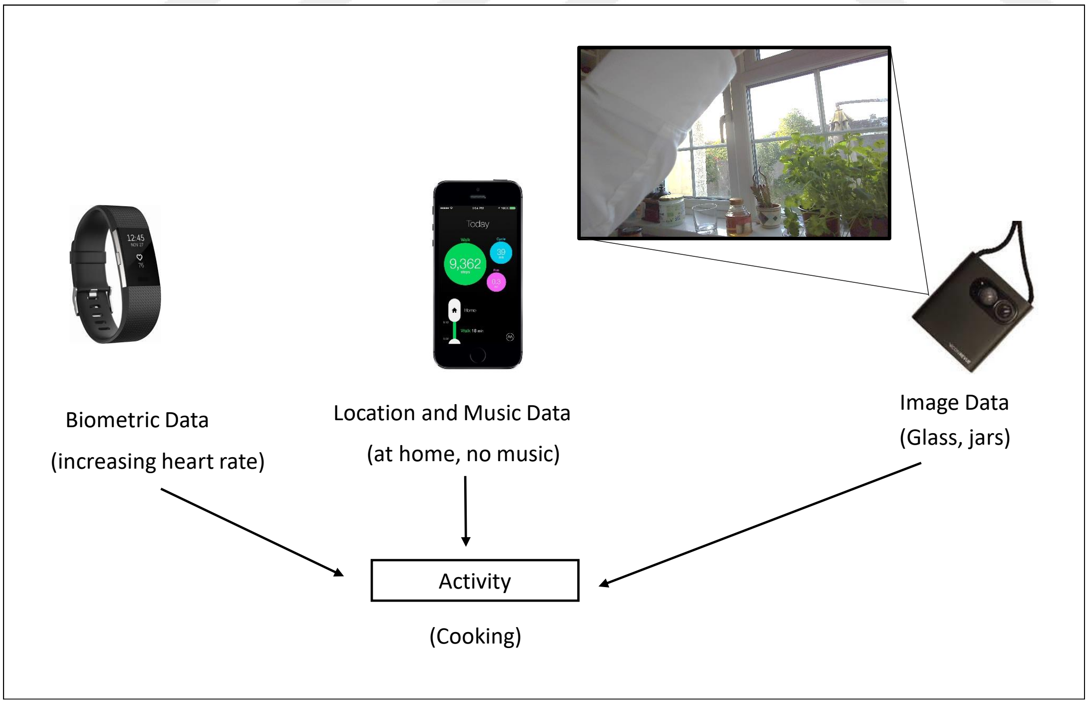



Figure 1.2: Gathering multimodal data source and defining human activity.

## 1.2 Proposed Methods and Model

NII Testbeds and Community for Information access Research (NTCIR) lifelogging dataset is chosen in this study. It is a multimodal dataset and contains minute-based biometric data, image data, location data, music data, and activity data of two users. The data given in the XML format were extracted and migrated all data to the MYSQL database system. Then, these minute-based biometric data are analyzed according to the activities.

In this thesis, the experiments focus on learning the activities using a human's lifelog. There is a relationship between human activity and biometric measurement. In order to classify activities according to biometric data, well-known machine learning algorithms such as Random Forest, Decision Tree, K-Nearest Neighbor have been applied to the biometric data. Some state of art algorithms, such as Catboost, XGBoost, LightGBM, have also been applied to the biometric data for activity learning.

Then, some semi-synthetic features are proposed to improve classification performance. Semi-synthetic features are obtained from existing biometric features using mathematical operations such as mean, sum, multiplication and square. The applied learning algorithms allow to better learn activity better learn the activity using newly created information. On the other hand, you need to learn more data by increasing feature dimensionality, and reducing run time performance. Thus, we propose a genetic algorithm for the selection of futures. It eliminates redundant features and improves the accuracy score for our problem.

In addition, late fusion methods are used for image data and biometric data. It combines complementary data and learns the activity using this fused result. Moreover, it is useful to create insight information which is not learned from a single data source. In our work, two late fusion methods are studied, namely late rule-based and classification-based fusions. The results of late fusion show an improvement in classification performance.

## 1.3 Contributions and Novelties

The contributions of this study are as follows:

- We extract the NTCIR lifelogging data in an XML file, and migrate to the MYSQL database system.
- We generate 18 semi-synthetic features from existing biometric data using mathematical operations such as sum, multiplication, mean and square.
- Semi-synthetic features increase the dimensionality of the dataset. Therefore, feature selection is necessary for reducing dimensionality and learning important features. We use the genetic algorithm for feature selection on biometric and semi-synthetic features. The results show that six semi-synthetic and five original features are selected among thirty features and accuracy results are improved from 89.22% to 90.91%, with even fewer features.
- We use some algorithms, namely CatBoost, LightGBM and XGBoost, as well as well-known machine learning algorithms, namely Random Forest, Naïve Bayes, K-Nearest Neighbors, Decision Tree, Support Vector Machine and Neural Network. These algorithms are experimented on the biometric and semi-synthetic data to learn human activity.The best accuracy result with 89.08% is obtained with Catboost algorithm on the dataset containing 3 activity classes and 12 features, while the Random forest algorithm are gave best accuracy scores on the semi-synthetic dataset and the dataset containing 16 activity classes with 89.22% and 60.77% respectively.
- We use two different late fusion methods on the image data and biometric data results to obtain more accurate results, namely late rule-based and classification-based fusions. The first method is to select the activity which has the highest class probability ratio among the results of the image and biometric data. The second method consists of learning probabilities of activity classes separately from image and biometric data. We concatenate these probability ratios and give these data to a new classifier to learn the activity. The fusion methods are applied on two lifelogging datasets containg 3 activity classes and 16 activity classes. Maximum rule based late fusion improves accuracy from 60.77%

to 86.22% and classification-based fusion give higher accuracy with 88.18% for the dataset containg 16 activity classes. On the other hand, maximum rule based late fusion has higher accuracy for the dataset containg 3 activity classes and the result improves from 89.08% to 95.29%. Classification-based fusion also have positive effect on accuracy core with 93.72%.

### 1.4 The Outline of the Thesis

This thesis is organized as follows.

Chapter 2 gives a summary of the background and research in the literature about lifelogging and activity prediction. Chapter 3 discusses the detail of our activity prediction methods. Dataset, preprocessing data, user evaluation, and proposed works are mentioned in this chapter. Chapter 4 focusses on the fusion method for multi-modal data and detail of applied fusion method are given. Chapter 5 explains the experiment results of the proposed solution. Also, classification performances of the experiments are compared. Finally, Chapter 6 gives a summary of the thesis.

# CHAPTER 2

# BACKGROUNDS AND RELATED WORKS

In this chapter, feature engineering methods, applied algorithms and literature reviews are stated.

## 2.1 Semi-Synthetic Feature Generation

Feature generation is also called as feature construction, feature extraction or feature engineering. The definitions for these terms have some differences. Feature construction is the creating features from raw data [3]. Feature extraction is creating a mapping to convert original features to new features [4]. Feature engineering is creating new features using one or multiple features [5].

In this thesis, feature engineering is used as feature generation method. The goal of feature engineering is accuracy improvement. Some dataset does not have obvious pattern. New meaningful feature are generated by using unobvious feature. Also, there is feature interaction in some dataset. Two or more features are not relevant with each other, but when they come together, they have a high influence on the dataset. The feature engineering improves accuracy by creating high important features.

In feature engineering, the new features are generated by one or more features with some mathematical operation like multiply, average, square. As a result of this process, new semi-synthetic features are created and the result feature set contains more features than the original feature set.

## 2.2 Feature Selection

Feature selection is a method to find the optimal subset features of a given dateset. It is important for the dataset containing high feature size. The dataset needs to be cleaned from irrelevant features. The main aim of feature selection is removing useless and redundant data [6].Selecting a subset of optimal features from whole features provides to reduce cost and running time of problem's solution. In addition, it improves accuracy score.

Genetic algorithm is inspired by Charles Darwin's idea of natural selection. It is a kind of search technique to find an optimal solution and is used for feature selection [7]. It simulates abstract representations (chromosomes, gen type of genes) of candidate solutions (individuals, phenotype), mutation strategies, reproduction methods and fitness functions that decides which individual survives to the next generation. Genes represent each feature and their values are determined based on existing features. If the feature exists, the value of this gene is set 1. In another case, it is set 0. All feature row is represented by chromosomes, and all chromosomes create population.

## 2.3 Classification Algorithms

This section mentions about the background information of applied classification algorithms.

### 2.3.1 Naive Bayes

A Naive Bayes classifier which is a kind of probabilistic machine learning model is used for classification problem. Naive Bayes algorithm works using the Bayes theorem.Bayes theorem uses formula 2.1

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})}. \tag{2.1}$$

Given a class variable is shown with $\theta$ and a dependent feature vector is shown with **D**. This Bayes theorem 2.1 is attempted around this thesis on the lifelogging dataset, added synthetic feature version of lifelogging dataset and version of extended with new activity dataset.

### 2.3.2 K-Nearest Neighbors Model

K-Nearest Neighbors(KNN) is a simple classification algorithm. The algorithm rule is firstly introduced by Evelyn Fix and J.L. Hodges [8]. This algorithm postpones the learning part until a new instance comes. Therefore it is called a lazy learning algorithm. KNN algorithm calculates the distances of each instances according to distance function and finds the nearest neighbors. Sample instances are defined as a class that belongs to the nearest neighbors.

### 2.3.3 Decision Tree Model

Decision Tree Classifier is a non-parametric supervised learning method that is used for classification and regression problems. It recursively divides data into node segments which consist of the root, split, and leaf nodes. Data splitting step is done at each node using certain stopping criteria [9]. This algorithm aims to build a model which predicts the target value by learning decision rules which are inferred from the features of the dataset.

### 2.3.4 Random Forest Model

Random Forest is a kind of machine learning method which is used for classification and regression. It constructs multiple decision trees on sub-dataset and learns from sub-trees. To improve prediction accuracy, it uses the average result. Random Forests consist of multiple decision tree classifiers and work according to the Breiman's "bagging" idea. It generates a new training sample using the original training sample. It repeatedly selects random sets of sample [10]. Representation of Random Forest is given in Figure 2.1. This figure show that dataset divided into many sub-decision trees. Each of sub-trees predict a result. Final prediction result determine with respect to majority voting of classes.

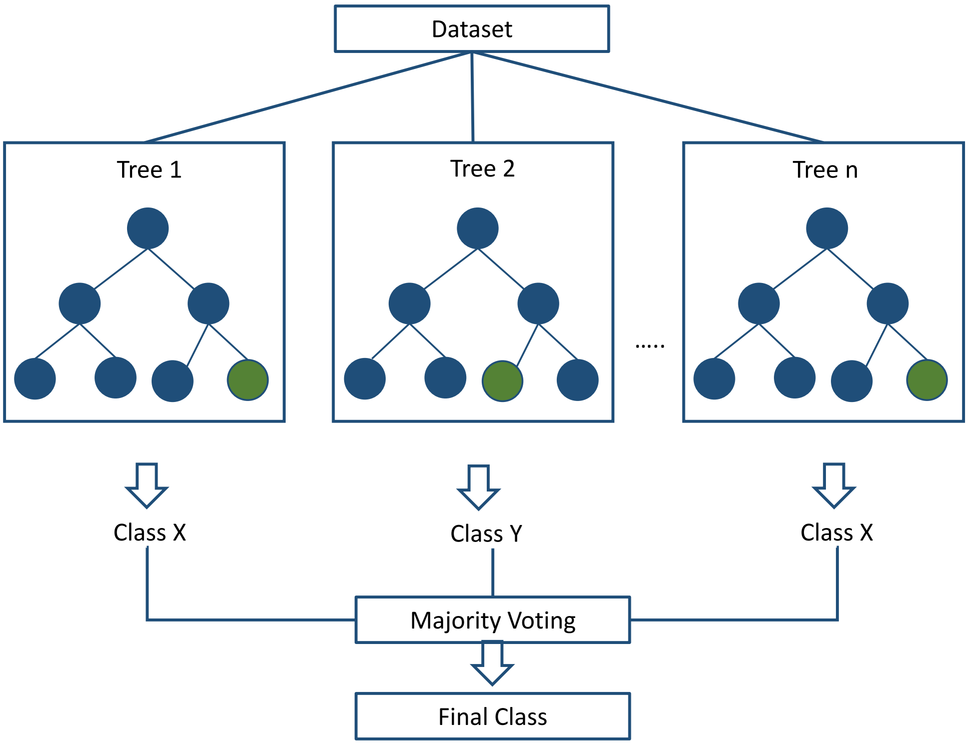



Figure 2.1: Random Forest Diagram

### 2.3.5 Support vector machine

Support vector machine (SVM) is a kind of machine learning algorithm which is a supervised learning algorithm. It was proposed by Vapnik [11]. SVM has hyperplane which separates data from different classes. All of the instances in the dataset are labeled according to existing classes. Then all instances are shown with points in space. SVM is an instance-based algorithm. It builds a model from the training part of the dataset, and unknown instances are classified by using this trained model.SVM algorithm was successfully applied to various engineering problems [12]. For instance, pattern recognition, handwritten digit recognition which is known problem was solved with SVM. The major applications of the SVM used in real life include time series forecasting, handwriting recognition, text categorization, bankruptcy prediction, face identification and recognition, and biological and medical aid. In addition to these, SVM was used for human activity classification [13], [14].

### 2.3.6 Neural Network Model

The neural network is an algorithm that is inspired by the working principle of the biological brain. First concept about an explanation of how the brain works published by Warren McCullock and Walter Pitts [15]. Then this idea developed, and this is used to recognize patterns. As shown in figure 2.2, the neural network occurs three main layers such as input layer, hidden layer, and output layer. Each layer has neurons that process information. The patterns are given to the neural network by the input layer. Then, the hidden layer receives them. In the hidden layers, the neuron calculates a weighted and bias. This process iteratively continue in a backward manner.

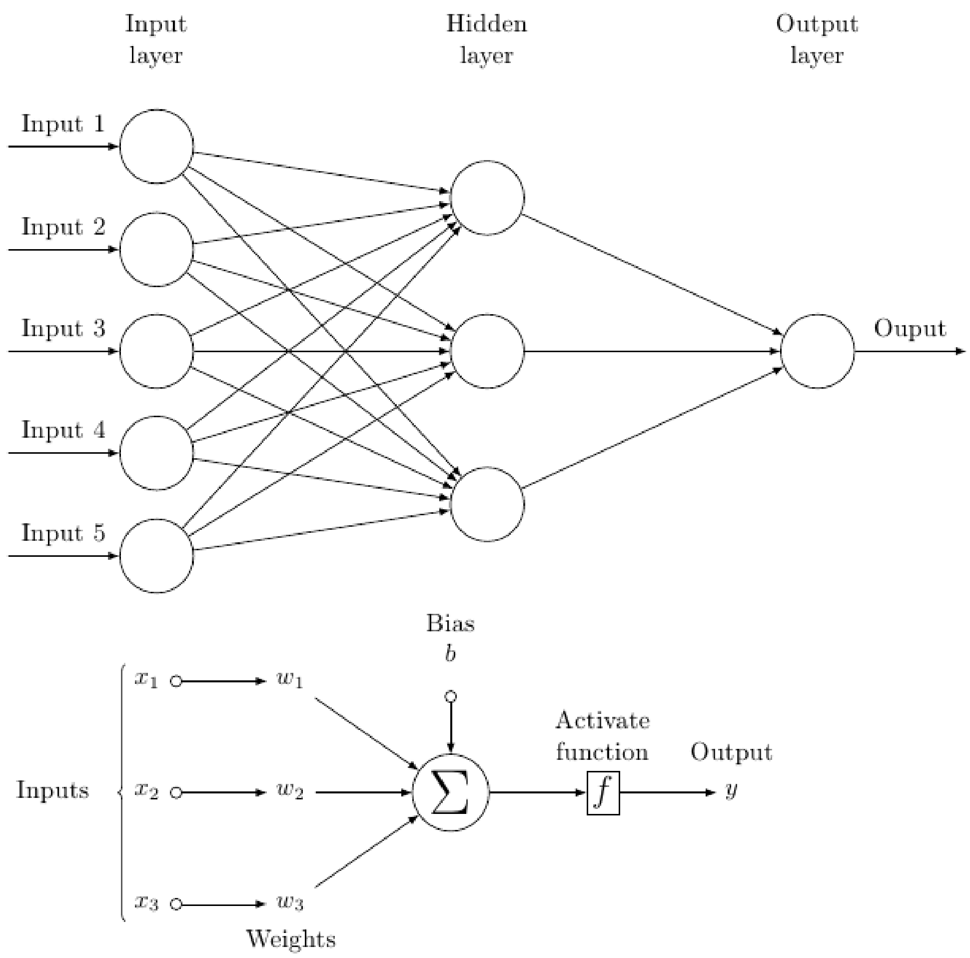



Figure 2.2: Neural Network Diagram

### 2.3.7 XGBoost

XGBoost is a type of popular gradient boosting algorithm which is proposed by Friedman in 2001. This algorithm uses a gradient descent algorithm for minimizing the loss while adding a new model. $C$ is a classification. $X_i$ and $Y_i$ are the training and the class label respectively. The prediction scores is summed up to obtain the final score. Next, the final score evaluated with respecting to Equation 1

$$\hat{y_\iota} = \sum_{k=1}^{C} f_k(x_i)_r f_k \in F. \tag{1}$$

XGBoost algorithm has a lot of advantages. This algorithm works efficiently, and it provides accurate solutions to many data science problems. Also, this algorithm aims to work portable and flexible.

### 2.3.8 CatBoost

CatBoost is a novel gradient boosting algorithm proposed by Yandex Company. It works better with categorical values because the algorithm has own encoder method. That means no need to do an encoding process. Besides, CatBoost algorithm has many improvements when it is compared with other boosting algorithms. CatBoost creates symmetry trees. The first advantage of this symmetry tree is overcoming the overfitting problem. Secondly, it works faster on the GPU. Lastly, It increases reliability [16].

### 2.3.9 LightGBM

LightGBM is a kind of gradient boosting method which uses a tree-based learning algorithm. LightGBM proposed novel techniques for decreasing the complexity of histogram building. One of these novel techniques is Gradient Based One Side Sampling (GOSS), and another one is Exclusive Feature Bundling (EFB). LightGBM has differences than other known gradient boosting method. LightGBM grows trees vertically. However, the other algorithms grow trees horizontally. That means LightGBM grows tree leaf-wise, but other algorithms grow level-wise. This difference provides to work faster than other boosting algorithms. Figure 2.3 shows that leaf-wise growth.

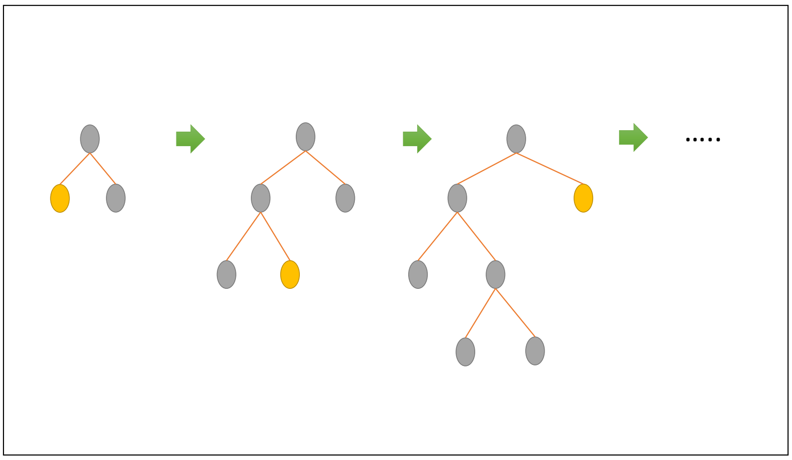



Figure 2.3: Leaf-wise tree growth.

## 2.4 Data Fusion

Data fusion is a process dealing with the correlation, association and combination of data from single and multiple sources. The aim of data fusion is improved information by using multiple sources. Complementary information from different sources can be learned with the data fusion and more inferences are obtained from multimodal data instead of using data from a single source [17]. Therefore, it is preferred to make an prediction on the dataset accurately. On the other hand, it needs attention to use correctly because data fusion increase the data size. Running performance problem can be occurred [18].

Data Fusion is categorized with respecting to different criteria such as applied level, time and data type. In our work, the fusion methods are classified as early fusion and late fusion.

### 2.4.1 Early Fusion

Early fusion is a process for multisensor data. The features from different sensor inputs are extracted. Then, these features are combined in a new feature vector and multimodal data which are created using the new features are classified. Early fusion has an advantage. The data from multisensor inputs are combined and different type of data are obtain. It increases the dimensionality of the feature vectors and give more information about the data model.

### 2.4.2 Late Fusion

Late fusion is a process on the trained data and a useful technique for multimodal data. Each of multisensor data is trained with different classification algorithms and pre-classified independently. Then, the final classification is done based on the fusion of the class probabilities of the trained data. Two kinds of late fusion model can be applied namely, rule based late fusion and classification based late fusion. In rule based late fusion, the class probabilities are fused some methodologies such as averaging, maximum, weighting, voting, sum rule, product rule. In classification based late fusion, the class probabilities are concatenated into new vector. Then, new vector fed into new classifier and the latest decision are determined for the class [19] [20].

Late fusion method is generally used more than early fusion. Late fusion provides flexibility in the classification model. It can be used for all classification models. Each of multisensor data can be trained with different models. Moreover, it does not create high dimensionality of data feature like early fusion and prevent the performance problem.

## 2.5 Related Works About Lifelogging

In this section, used methodologies and works in lifelogging are explained.

History of the lifelogging system begins with Vannear Bush's 1945 "Memex" vision. "Memex" is a device that provides storing, searching, and indexing of personal information [21]. It is a sort of desk and brings all the material information on it. This information is archived, and they become part of lifelogging data. "Memex" has a huge impact on developing lifelogging systems. It provides limited data. Therefore it needs to be developed. Nowadays, people with wearable cameras can record their lifelogs easily. Fast growing technology is enabled to collect various types of lifelogging data such as voice, video, image, biometrics data.

Lifelogging data is collected using wearable sensors to record raw data. These raw data are segmented into activities, are annotated, and can be accessible by search or browsing tools. Information retrieval is an essential part of lifelogging. It can be access to different kinds of data. Through a timeline of semantic life, the activity can be reachable with information retrieval [22]. Li et al. proposed ZhiWo system for information retrieval from lifelogging data. It is a kind of human activity recognition system. It semantically processes sensor data and identifies user activities using the machine learning model. They trained data with Support Vector Machine[22].

Smartphones not only show biometric lifelog data but also show the mood of people. A. Bogomolow et al. worked on the prediction of happiness of human based on smartphone lifelog data. They want to classify happiness as happy, neutral, and unhappy. The authors proposed a method to recognize the happiness of a person using mobile phone data (such as SMS, call history and Bluetooth proximity data) with Random Forest classifier. According to their work, the classification accuracy of daily happiness is 80.81% [23].

Lifelogs are chronological sequences of activity objects. These sequences of data are useful for the recommendation system. G. Kumar et al. described the context of activity occurrence for building a content-based recommendation system. They worked on recommendation systems using similarity between current and past sequences of lifelog data. Lifelogs give information about daily activities which can be learned user's daily routine and help to recognize personalized activity plan. New future activities which will be recommended to user derived from similar timelines activities. They use user specific contextual information for each user such as a sequence of activity objects, device and sensor data [24].

Different kinds of work are done using the lifelogging dataset. One of them is classification. N. Ravi et al. focused on finding the best classifiers for recognizing activities. Among the selected features, he defined the more important ones and detected activities that are difficult to recognize. Firstly, the feature extraction part is done. Then he has worked with some classifiers: Decision Trees, K-nearest neighbors, Support Vector Machine and Naive Bayes. [25]

Physical activities affect human biometric values. Heart rate is an example of this. Works in this area show that physical activities have a huge impact on increasing or decreasing heart rate. Different kind of study is done for recognizing the relationship between them. F. Xiao et al. define a model that has a multi-step prediction for high accuracy. They used the Feed Forward Neural Network (FFNN) method. In order to optimize the structure and weight of the FFNN, an evolutionary algorithm was chosen [26].

The lifelogging data contains multi-sensor data. Fusion methods are applied on these multi-sensor data to obtain meaningful data and new inferences. There are many data fusion methods. H. Gunes et al. described a data fusion methods and applied on decision level. They use function based late fusion using product, sum and weight criteria. They have improvement on their result with late fusion [20]. On the other hand, fusion of multisensor data can cause high dimensionality of the data and performance of the system is decreased. Thus, feature selection can be needed. There are different feature selection methods. E. Ijjina et al. proposed genetic algorithm for feature selection. Proposed algorithms finds the appropriate subset features by using evolutionary theorem [27].

## 2.6 Recent Literature on Lifelogging

The lifelogging has been gaining significant attention from academia in recent years and latest work are stated in this section.

Researchers work on human daily activity recognition using wearable and visual sensing data. In 2016, Xi Liu et al. recorded accelerometer and visual data of lifelogger by smart devices. Then, they extracted important features among all recorded data and identified 20 activities. They aim to recognize human activities with these selected features, and they learned activities using a nonlinear SVM algorithm and a linear algorithm, called Logistic Regression. Their experiments show that SVM gets better performance than Logistic Regression [28].

K. Belli worked on the NTCIR lifelogging dataset for activity classification. She defined new activities using image and annotations, which are given on dataset. Activity numbers are increased from 3-class to 16-class [29]. K.Belli worked on image and annotation information to predict user activities with the NTCIR lifelogging dataset. She proposed a combined model as shown in the Figure 2.4 for predicting activities using image and annotation information. Images were trained with ResNet-50 architecture, and annotation texts were trained with a multilayer perceptron (MLP) classifier. The results of images and text annotations were concatenated. Then batch normalization was applied. Finally, new activities are classified on the NTCIR lifelogging dataset. This dataset has missing values. All cases are situated below:

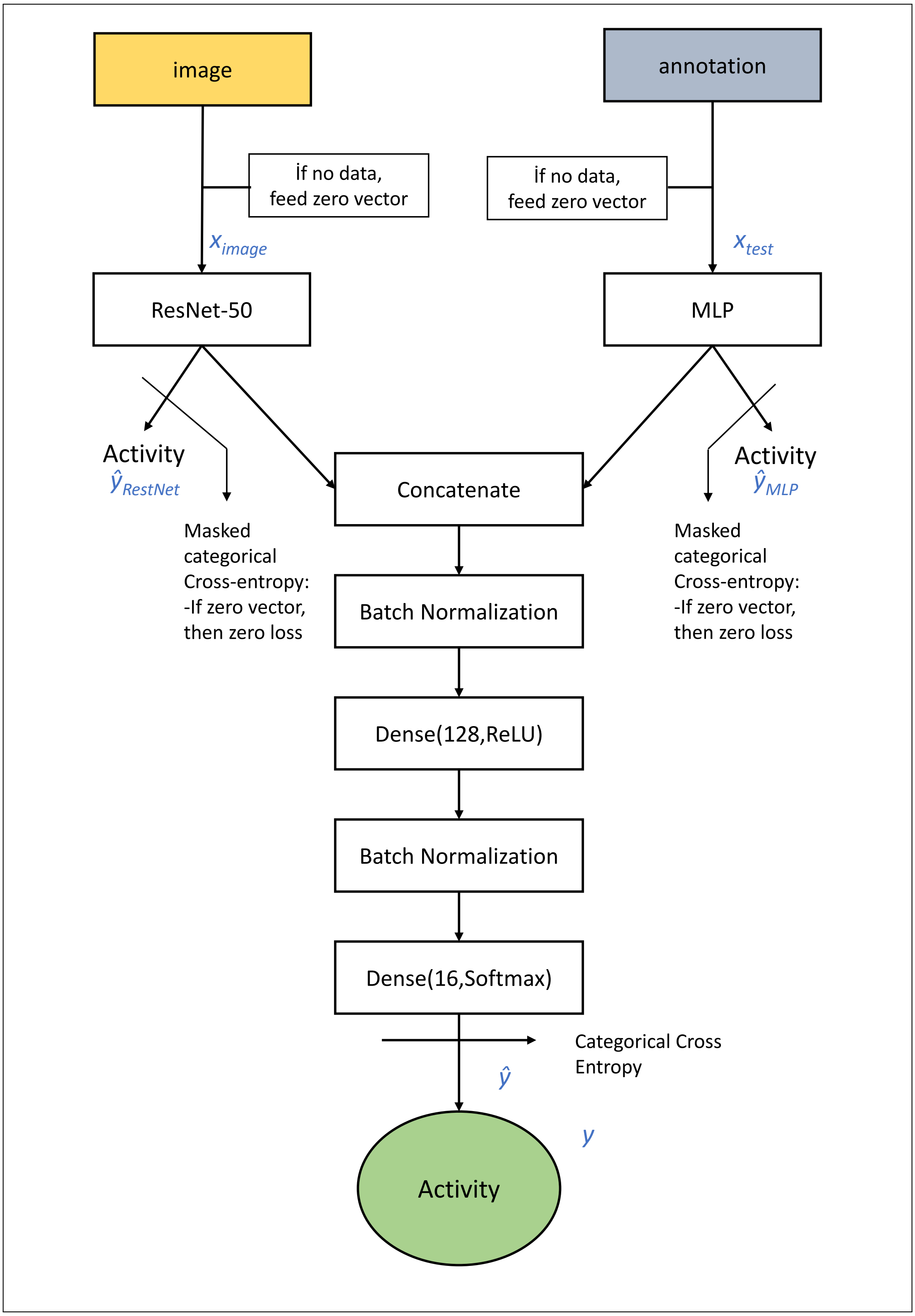



Figure 2.4: Learning model for image and annotation information on NTCIR lifelogging dataset

- There is biometric information on a minute with missing image and annotation.
- There is image information on a minute with missing annotation.
- There is annotation information with missing images.
- There is image and annotation information on a minute

In order to overcome missing images, a zero-image vector was used. This zero-image vector is a kind of vector, and all values of it are zero. Then, this zero-image vector was fed to her model. The same approach was applied to missing annotations. Zero-vector for annotations was fed to the MLP subsection of the model. She tried to prevent the learning model from zero-vector. She introduced a version of categorical cross-entropy as the loss function for ResNet-50 image classifier and MLP text classifier outputs. Equation 2.2 was used as a loss function of ResNet-50 image classifier and Equation 2.3 was used as a loss function of MLP text classifier model.To ignore zero-vector for feeding model, equations 2.4 and 2.5 were used [29]. The proposed methods increased the accuracy result. The result of this combined model are used for late fusion in this thesis.

$$Loss_1(x, y, \hat{y}_{ResNet}) = -\sum_{i=0}^{N}(c(x_i) * \sum_{j=0}^{M}(y_{ij} * log(\hat{y}_{ResNet,ij}))) \tag{2.2}$$

$$Loss_2(t, y, \hat{y}_{MLP}) = -\sum_{i=0}^{N}(c(t_i) * \sum_{j=0}^{M}(y_{ij} * log(\hat{y}_{MLP,ij}))) \tag{2.3}$$

$$c(x_i) = \begin{cases} 1, & \text{if } sum(x_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

$$c(t_i) = \begin{cases} 1, & \text{if } sum(t_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

where each record can be shown as a tuple $r$;

$\boldsymbol{r} = (x, t, y, \hat{y}_{ResNet}, \hat{y}_{MLP}, \hat{y})$

$\boldsymbol{x}$ : Input for image

$\boldsymbol{t}$ : Input for text

$\boldsymbol{y}$ : Ground-truth label

$\boldsymbol{\hat{y}_{ResNet}}$ : Prediction generated by ResNet-50 sub-section of the model

$\boldsymbol{\hat{y}_{MLP}}$ : Prediction generated by MLP sub-section of the model

$\boldsymbol{\hat{y}}$ : Prediction generated by the complete model

$\boldsymbol{M}$ : Number of categories

$\boldsymbol{N}$ : Number of records

# CHAPTER 3

# ACTIVITY PREDICTION FROM LIFELOGGING DATA

## 3.1 Dataset

The data was taken from the NII Testbeds and Community for Information access Research (NTCIR) repository. The dataset was published for the first time by Gurrin et al.[30].It contains multimodal lifelog data. The lifelog data was collected from two lifeloggers (user 1 and user 2) for about 90 days period. Lifeloggers used an OMG Autograper wearable camera and a smartphone application for recording their lifelogs minute-by-minute. These recorded data created their life archives. [30].

Below table 3.1 shows statics information about dataset.

Table 3.1: Statistics Information of Dataset

| **Type** | **Count** |
|---|---|
| Lifeloggers | 2 |
| Days | 90 |
| Image | 88,124 |

Dataset has three types of data: multimedia data, biometric data, and human activity data. Biometrics data and human activity data are stored in XML files. The below Figure 3.1 shows XML data structure. Totally, there are 90 days of lifelog data. 59 days of belong to User1 and 31 days of belong to User2.

```xml
<?xml version="1.0" encoding="utf-8"?>
<NTCIR-lifelog-dataset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///C:/Users/gizem/Desktop/ntcir.xsd">
        <users>
                <user>
                        <days>
                                <day>
                                        <date>2016-08-08</date>
                                        <image-directory>NTCIR-Lifelog-dataset/u1/2016-08-08/</image-directory>
                                        <day-metrics>
                                                <Calories>2684.6</Calories>
                                                <Steps>9916</Steps>
                                                <Sleep-Duration>318</Sleep-Duration>
                                                <Sleep-Score>60</Sleep-Score>
                                                <Number-of-Interruptions>0</Number-of-Interruptions>
                                                <Number-of-Toss-and-Turns>40</Number-of-Toss-and-Turns>
                                                <Interruption-Duration>0</Interruption-Duration>
                                                <Resting-Heart-Rate>66</Resting-Heart-Rate>
                                                <Walk-Calories>381.2</Walk-Calories>
                                                <Walk-Duration>70</Walk-Duration>
                                                <Walk-Steps>7643</Walk-Steps>
                                        </day-metrics>
                                        <day-activities/>
                                        <personal-logs>
                                                <health-logs/>
                                                <food-logs/>
                                                <drink-logs/>
                                        </personal-logs>

                                        <minutes>
                                                <minute>
                                                        <location>
                                                                <name>Home</name>
                                                        </location>
                                                        <bodymetrics/>
                                                </minute>
                                                .
                                                .
                                                .
                                        </minutes>
                                                .
                                                .
                                                .
                                </day>
                                                .
                                                .
                                                .
                        </days>
                </user>
        </users>
</NTCIR-lifelog-dataset>
```


Figure 3.1: XML Formatted Lifelogging Data

Part of XML elements are shown in the Figures 3.2 and 3.3. In these values, some of them are worked on activity prediction.

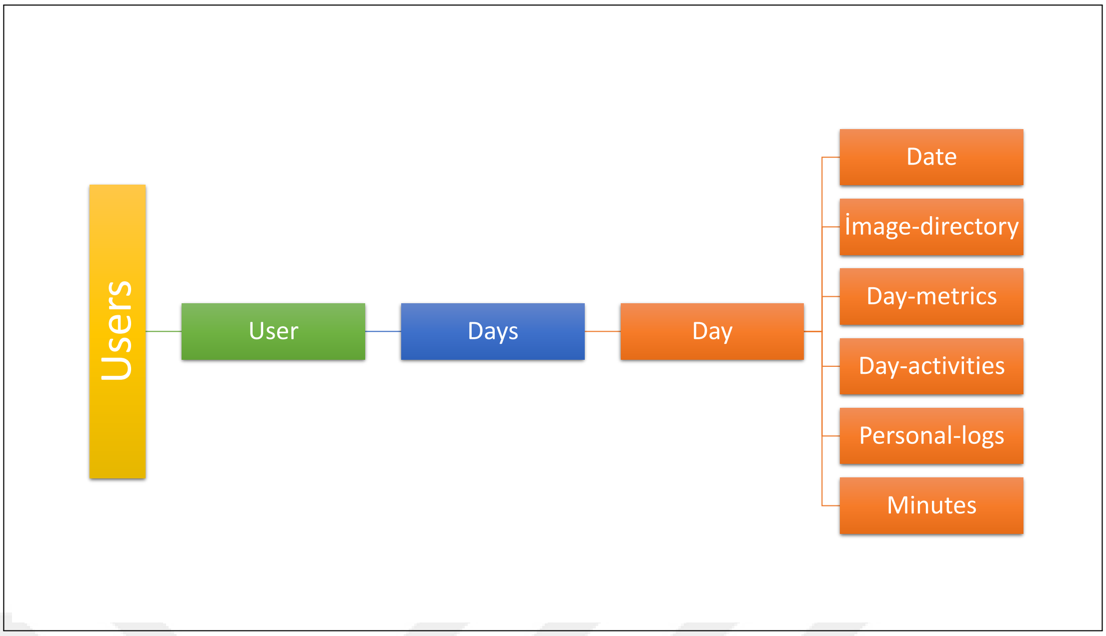



Figure 3.2: Dataset

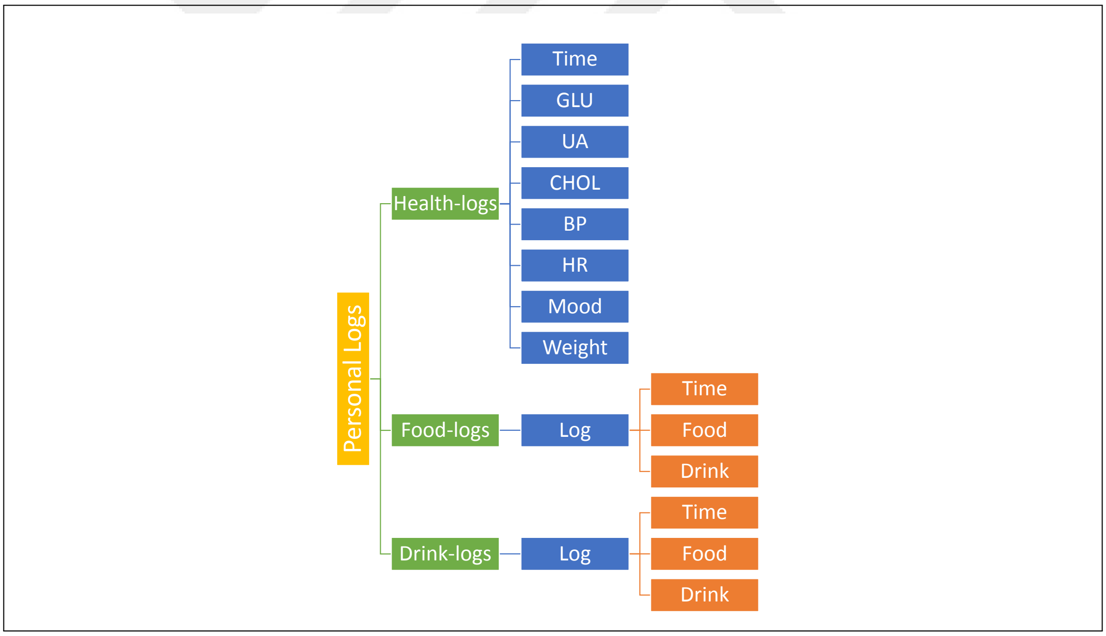



Figure 3.3: Personal Log

In this study, activity prediction is worked on the NTCIR lifelogging dataset using biometric values on each minute. Below table 3.2 shows activity name and count on the dataset.

Table 3.2: Count of activities in NTCIR lifelogging dataset

| **Activity** | **Count** |
|---|---|
| Airplane | 1,059 |
| Transport | 7,119 |
| Walking | 2,854 |

Multimedia data is given as a jpeg formatted photo under the daily folder and image information for minutes is stated in the XML file. In total there are 88,124 images in this dataset. Some of the minutes have missing image information, but some of them have one image information. Besides, others have more than one image information. Multimedia data are not worked on this study. However, activities are defined in other works by K. Belli using these multimedia data [29]. We used these activities to compare prediction accuracy results using existing activities and tagged new activities.

Although the dataset contains 129,600 minutes, each minute does not have a tagged activity name. There are only 11,032 minutes which include activity information. New tagged activities which are defined using multimedia data of the NTCIR lifelogging dataset are used to work with more data. Information of new tagged activities are shown below table 3.3 [29]. The table 3.3 includes 14 new activity names and the count of new activities. There is two more activity are defined as *"eating"* and *"time with children"* but user 1 does not have these activities. Thus, just 14 classes are used.

## 3.2 Data Preparation

Data preparation is a process of cleaning and transforming data. In this work, the NTCIR lifelogging dataset given an XML file and we need to transform XML data to the database data. Using an XML Tool which is an XML editor for modeling, editing, and transforming the XML file, the XML Schema Definition(XSD) file of provided lifelogging data is generated. Created XSD file is used with Java API called Java Architecture for XML Binding (JAXB). JAXB is used for extracting data from

Table 3.3: Count of tagging activities in NTCIR lifelogging dataset

| **Activity** | **Count** |
|---|---:|
| Cooking | 1,410 |
| Creative activities | 95 |
| Eating | 2,511 |
| F2f interacting | 7,920 |
| Houseworking | 3,066 |
| Other activities | 4,393 |
| Physical activities | 6,913 |
| Praying | 110 |
| Reading | 435 |
| Relaxing | 16,735 |
| Shopping | 2,172 |
| Socialising | 5,873 |
| Traveling | 13,504 |
| Using a computer | 24,641 |

an existing XML file into Java classes regarding XSD rules. Data in provided XML file are transformed into JAVA object classes. Then, JAVA object data are stored into MYSQL database tables. Finally, the migration of XML data to MYSQL database is completed as represented in the Figure 3.4.

The dataset includes image data. In the XML file, the minute attribute contains image information at that moment. Some of the minute attributes don't have image information, but some of them have one or more image information. Image information shows the path of the image directory. Using these image data missing activity detection is studied by K. Belli [29]. She trained images and defined to missing minute activities with trained images. We merged biometric data information and identified new minutes activities using her work. In this way, We decreased the missing activity rate. The used dataset is growing up from 3 classes to 16 classes and from 11,032 data to 89,494 data which has activity information.

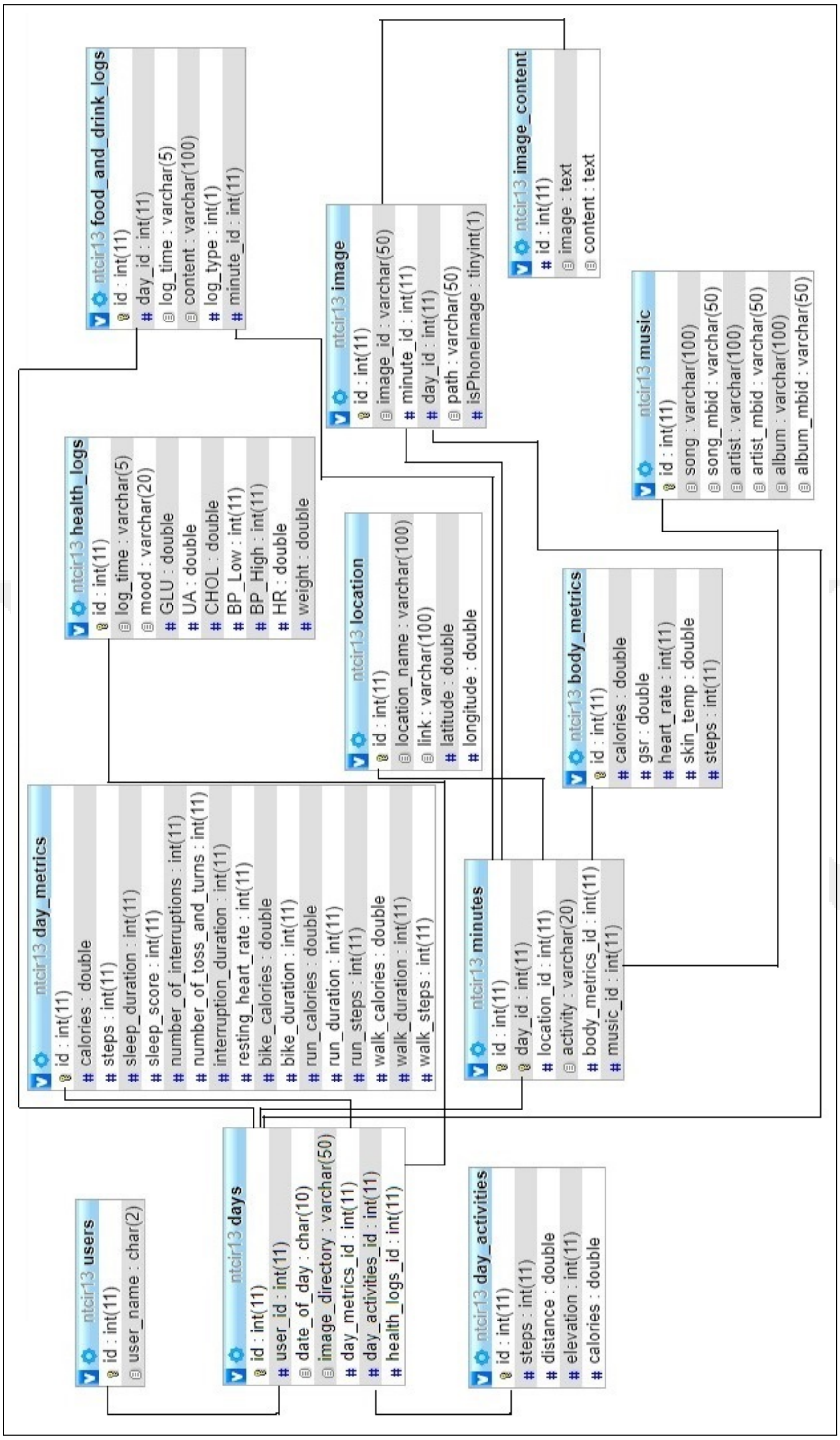



Figure 3.4: Database tables for NTCIR Lifelogging Data

### 3.3 Data Analysis

Data analysis is an essential part of knowing the data. It provides information about dataset. Besides, it affects to performance of problem solution. Thus, we make an analysis of the NTCIR Lifelogging dataset and the below Figure 3.5 shows the analysis of the dataset. According to this, there are some missing data, and the dataset needs to be a standardization process.

| | glu | weight | bp_low | bp_high | calories |
|---|---|---|---|---|---|
| count | 84960.000000 | 84960.000000 | 84960.000000 | 84960.000000 | 128814.000000 |
| mean | 5.605085 | 71.566102 | 65.559322 | 100.966102 | 1.225846 |
| std | 2.071237 | 26.274403 | 26.303001 | 40.303312 | 1.269598 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 6.000000 | 80.000000 | 71.000000 | 109.000000 | 0.000000 |
| 50% | 6.400000 | 81.000000 | 74.000000 | 116.000000 | 1.200000 |
| 75% | 6.500000 | 81.900000 | 78.000000 | 120.000000 | 1.600000 |
| max | 6.900000 | 83.100000 | 90.000000 | 128.000000 | 15.100000 |

| | steps | sleep_duration | SLEEP_SCORE | gsr |
|---|---|---|---|---|
| count | 128814.000000 | 129600.000000 | 129600.000000 | 128814.000000 |
| mean | 3.915902 | 195.888889 | 35.955556 | 0.553341 |
| std | 17.848470 | 194.955017 | 37.323033 | 3.236519 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50% | 0.000000 | 163.000000 | 20.500000 | 0.000060 |
| 75% | 0.000000 | 377.000000 | 72.000000 | 0.000204 |
| max | 179.000000 | 640.000000 | 95.000000 | 31.507900 |

| | heart_rate | skin_temp | music |
|---|---|---|---|
| count | 128814.000000 | 128814.000000 | 129600.000000 |
| mean | 38.940107 | 65.679051 | 0.005887 |
| std | 40.605891 | 37.464112 | 0.076503 |
| min | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 58.000000 | 0.000000 |
| 50% | 50.000000 | 86.000000 | 0.000000 |
| 75% | 75.000000 | 89.600000 | 0.000000 |
| max | 193.000000 | 163.000000 | 1.000000 |

Figure 3.5: Data Analysis of NTCIR Lifelogging Dataset

## 3.4 Data Preprocessing

Data preprocessing is a data mining technique that involves obtaining meaningful data from raw data. Lifelogging dataset includes incomplete and inconsistent data. Thus, data cleaning methods are applied. For example, missing values are filled, and the noisy data are smoothed. Then, standardization steps are used. I transform values such that the mean of the values is 0, and the standard deviation is 1.

The provided XML file includes music information such as song, artist and album. In this work, it is transformed into a feature that shows whether music is listened at that minute. If a minute has music information, it is represented with 1. Otherwise, it is represented with 0. This music information is used for activity prediction because psychology researchers show that there is a relationship between user's psychology and emotions containing heart rate [31], galvanic skin response (GSR)[32] and blood pressure [33].

## 3.5 Synthetic Feature Generation

By using existing features, new features are created. Firstly, the correlation among each feature is shown on Figure 3.6. According to this correlation information, related features are detected, and new features are generated from existing ones with some mathematical operation like multiply, average, square. The selected features are BP_LOW, BP_HIGH, CALORIES, GLU, GSR, HEART_RATE, SKIN_TEMP, SLEEP_DURATION and SLEEP_SCORE. Applied operations on selected features and new feature name are shown in Table 3.4. The number of features is increased from 12 existing features to 30 semi-synthetic features with respecting to applied feature generation method. This new dataset with semi-synthetic features is examined in the activity classification part.

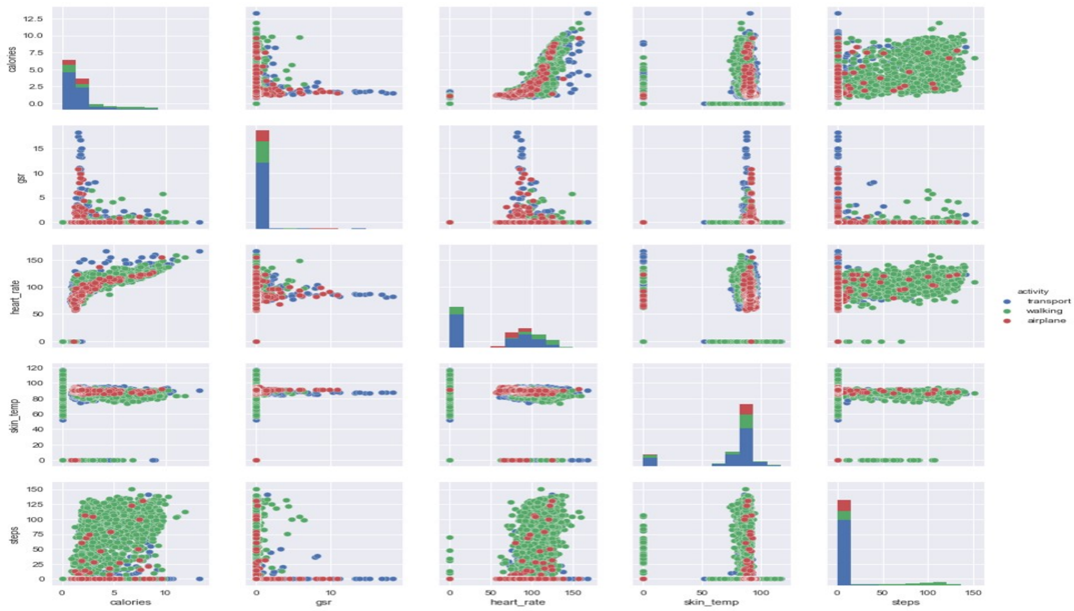



Figure 3.6: Data Analysis of NTCIR Lifelogging Dataset

## 3.6 Genetic Algorithm For Feature Selection

Feature selection is a kind of problem for selecting features from all features which are given dataset. This problem solution is used in many engineering and scientific areas. The aim of feature selection is removing useless and redundant data [6]. The whole process can be defined in four categories: generation of subset, evaluation of subset, a stopping criterion, and validation of the results [34]. Selecting a subset of $d$ features from whole $D$ features, using given criteria provides to reduce cost and running time of problem's solution. In addition to this, it gives a high accuracy rate.

There are some feature selection methods. In this work, genetic algorithms are applied as a feature selection method.

Genetic algorithms which are inspired by Charles Darwin's idea of natural selection are a search technique to find an optimal or approximate solution to optimization and search problems. Genetic algorithms are the same as the assumption that, based on the theory of evolution in the biological world. Experimental conditions that lead to better results will prevail over the worst and that some sort of random change and

Table 3.4: Generated Synthetic Features From Original Features

| **Operation** | **New Feature Name** |
|---|---|
| ( BP_LOW + BP_HIGH)/2 | bp_avg |
| ( BP_LOW * BP_HIGH) | bp_m |
| ( SLEEP_DURATION+ SLEEP_SCORE)/2 | sleep_sd |
| ( SLEEP_DURATION * SLEEP_SCORE) | sleep_sd_m |
| ( SLEEP_DURATION * SLEEP_DURATION) | sd_sqr |
| ( SLEEP_SCORE * SLEEP_SCORE) | sc_sqr |
| ( HEART_RATE + SKIN_TEMP)/2 | bm_hr_st |
| ( HEART_RATE * SKIN_TEMP) | bm_hr_st_m |
| ( HEART_RATE * HEART_RATE) | hrt_sqr |
| ( SKIN_TEMP * SKIN_TEMP) | skt_sqr |
| ( GSR + SKIN_TEMP)/2 | bm_gsr_st |
| ( GSR + HEART_RATE)/2 | bm_gsr_hr |
| ( GSR * SKIN_TEMP) | bm_gsr_st_m |
| ( GSR * HEART_RATE) | bm_gsr_hr_m |
| (CALORIES + HEART_RATE)/2 | bm_hr_c |
| (CALORIES * HEART_RATE) | bm_hr_c_m |
| (GLU + WEIGHT)/2 | glu_weight |
| (GLU * WEIGHT) | glu_weight_m |

recovery can be achieved through some kind of recombination. In other words, the best individuals which have the best environmental conditions to live hold the greatest probabilities for reproduction. Moreover, a combination of two good individuals can generate new offsprings which have better than their parents. So that, populations are created by evolving good genomes [7].

In this thesis, semi-synthetic features are generated. The number of feature is increased and some unimportant features can be created on feature generation step. Therefore, the feature selection is needed. Genetic algorithms are selected for our feature selection problem. Implementation of these algorithms is the computer simulations which include abstract representations (chromosomes, gen type of genes) of candidate solutions (individuals, phenotype), mutation strategies, reproduction methods and fitness functions that decides which individual survives to the next generation. Figure 3.7 shows genetic algorithm representations. Genes represent each feature. If the feature exists, the value of this gene is set 1. In another case, it is set 0. All feature row is represented by chromosomes (individuals), and all chromosomes create population.

Steps of genetic algorithm are shown int he Figure 3.8, briefly explain below and pseudocode of genetic algorithm is given on algorithm 1.

**Selection**: Select the most fitted individuals in a generation.
**Crossover**: Generate two new individuals, based on the genes of two solutions. These children will exist in the next generation.
**Mutation**: Replace a gene randomly in the individual.

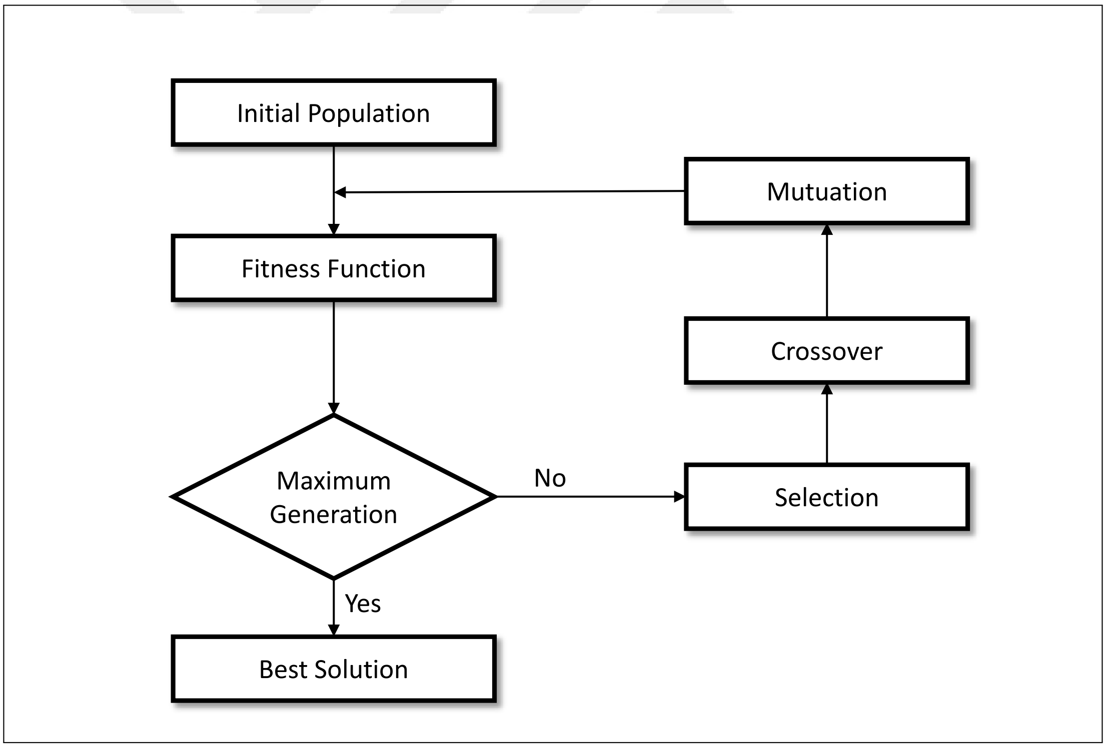



Figure 3.8: Steps of Genetic Algorithm

The approach of feature selection is both minimizing the number of features and maximizing the accuracy of a classifier. Thus it needs to make two goals. In such

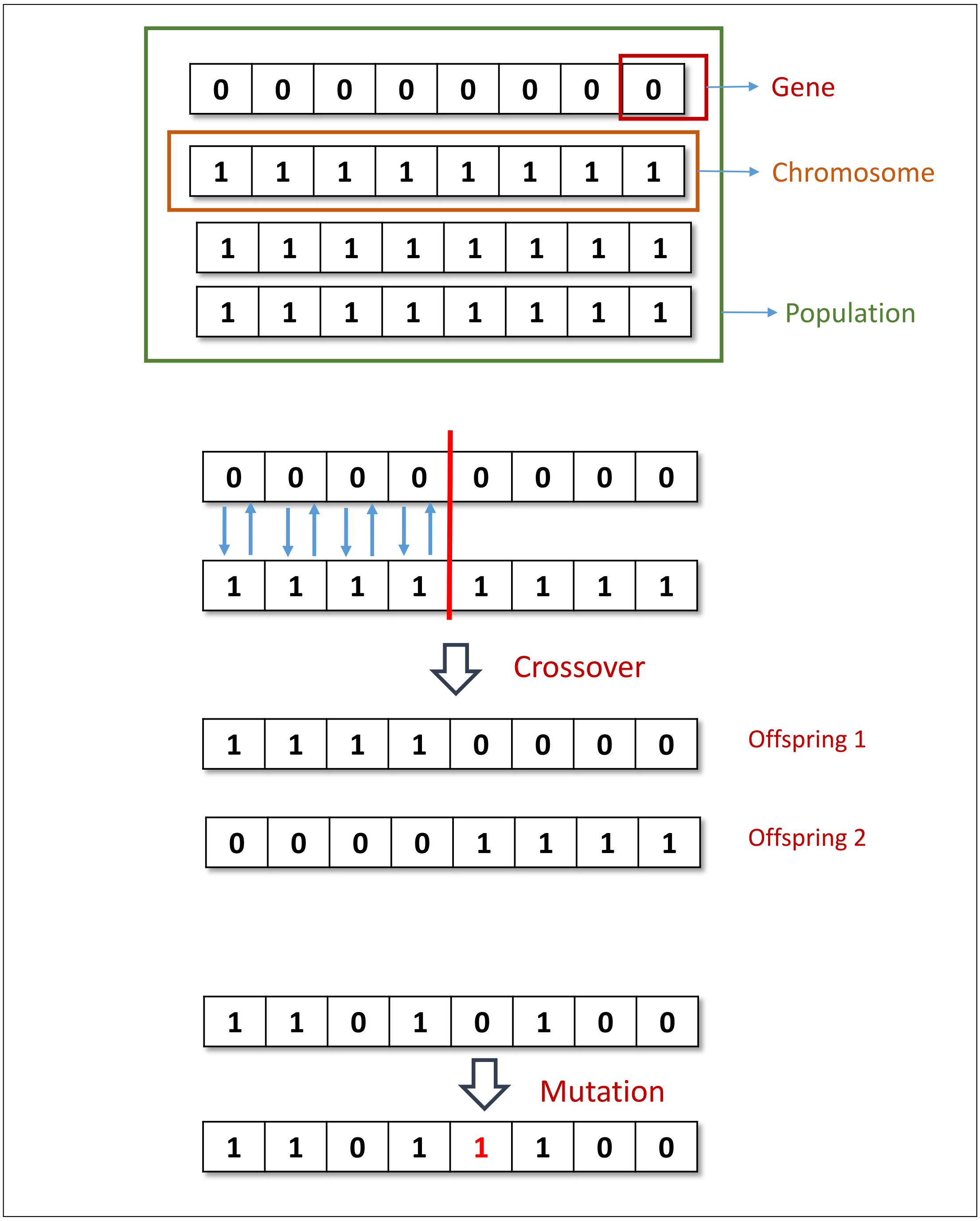



Figure 3.7: Genetic algorithm representation

**Algorithm 1:** Genetic Algorithm For Feature Selection

**Input:** Set of features

**Output:** Optimal feature subset

$P \leftarrow$ Generate initial population;

$F \leftarrow$ Evaluate initial population with fitness function;

$i \leftarrow$ iteration number;

$max \leftarrow$ maximum number of iteration;

**while** $i$ < $max$ **do**

  **for (** $eachchromosome, EvaluateFitness$ **)** {

    Create Random Forest Model;

    Train, validate and Test data;

    Fitness = Classification Accuracy;

  }

  $ParentSelect()$;

  $Crossover()$;

  $Mutation()$;

  $Elitism()$;

**end**

**return** *Optimal feature subset*

this case, an intrinsic conflict occurs between two or more problem goals. Therefore feature selection problem requires multi-objective optimization [35]. Multi-objective evolutionary algorithms have demonstrated to be highly effective about finding optimal solutions for multiple-objective problems [36] [37].

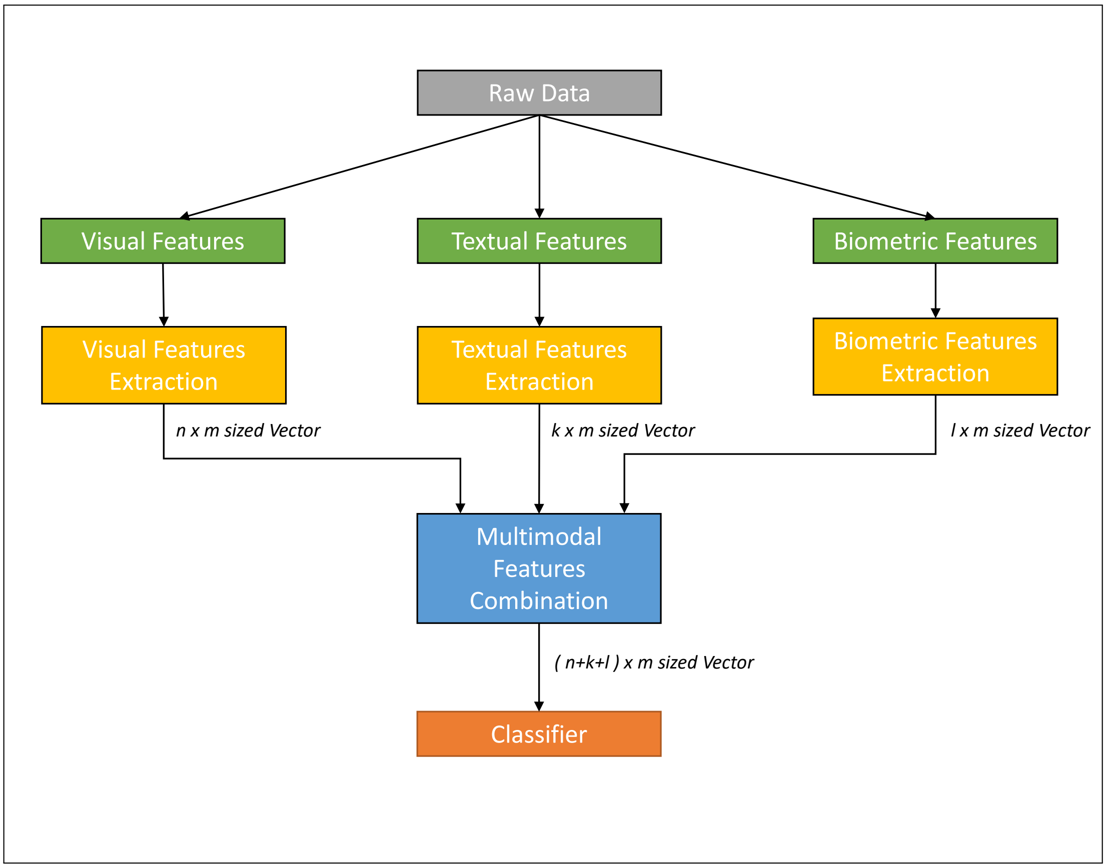
# CHAPTER 4

# FUSION OF BIOMETRIC DATA AND IMAGE DATA

Data fusion is a kind of process which combines data from multiple sources and can be applied to multimodal data. Data fusion aims to get improved information by using different sensor data. Complementary information from different sources can be brought with the fusion of multimodal data and more inferences are obtained from multimodal data rather than data from a single source [17]. Thus, it improves accuracy results and helps to make an accurate prediction on the dataset. For example, R. King et al. applied data fusion techniques on healthcare monitoring systems and new inferences are learned from multisensor inputs. Moreover, they observed an improvement of the accuracy [38]. On the other hand, fusing data from multiple sources compels running performance but this challenging task is necessary for effective utilization of multiple sources [18].

Data fusion is a commonly used technique for multisensor data and different kinds of data fusion techniques exist in the literature. The data fusion techniques are classified into different categories depends on the applied level, time and data type. In this thesis, the fusion methods are classified as early fusion and late fusion and late fusion methodologies are used on biometric data and image data of the two NTCIR lifelogging datasets containg 3 activity classes and 16 activity classes respectively. Although the biometric data have 14 activities, the image data have 16 activities. Missing two activities in class probability vector of biometric data are filled with zero to make comparable result.

Figure 4.1: Sample of Late Fusion method using maximum class probability for image data and biometric data




## 4.1 Early Fusion

In early fusion, the features from different sensor inputs are extracted with the feature extraction methods. The extracted features create the feature vectors $v$. These feature vectors are combined and fed into a classifier. Then, multimodal data created using the new features are classified.

Early fusion, commonly used a data fusion technique, is preferred for its advantages. It combines multisensor inputs and increases the dimensionality of the feature vectors. So that, it can give more inferences about the data model for classification problems. On the other hand, it has a critical disadvantage because increasing the feature dimensionality reduces run-time performance for machine learning algorithms.

## 4.2 Late Fusion

In late fusion, complementary data are brought by combining accuracy scores of models that are constructed with multisensor data [39]. Each of multisensor data is trained with different classification algorithms and pre-classified independently. Then, the final classification is done based on the fusion of the class probabilities of the trained data with a fusion methodology such as averaging, maximum, weighting, voting, sum rule, product rule or a learned model [19] [20].

Usually, late fusion is preferred rather than early fusion for two reasons. The first reason is that the feature concatenation in early fusion would result in a high dimensionality of data features and making a large multimodal dataset. The second reason is that late fusion provides flexibility in the classification model. Different data sources can be trained with different classifiers. For example, in our work, biometric data were trained with Random Forest algorithm and image data were trained with Convolutional Neural Network algorithm. This provides the best accuracy result for each modality.

In this thesis, two late fusion methodologies were applied, namely rule based and classification based late fusion on two version of the lifelogging dataset. Details of applied methodologies are given below subsections.

### 4.2.1 Rule-based Late Fusion

In rule based late fusion, probability scores of each model are combined using some rule such as maximum, avarage, mean, sum, weight etc. Final decision for classification is determined according to these rules. In our work, maximum rule was applied. $F$ is maximum function. Learning model $h$ is applied to the feature vectors $v$.The final decision is calculated with respecting to Equation 4.1

$$P = F_{max}(h_i(v_i), ....h_m(v_m)). \tag{4.1}$$

Maximum rule based late fusion was applied to biometric and image data on the NTCIR Lifelogging dataset to give the final decision about user activity. Firstly, class probability scores of the models are calculated for biometric test data and image test

**Algorithm 2:** Maximum Rule Based Late Fusion Algorithm

**Input:** Class probability vectors for image and biometric data

**Output:** Predicted activities

$I \leftarrow$ class probability vector for image data;

$B \leftarrow$ class probability vector for biometric data;

$N \leftarrow$ size of class probability vector;

**for** ( $i \leftarrow 1$ **to** $N$ ) {

  **if** $max(I_i)>max(B_i)$ **then**

    Select predicted activity of image data;

  **else**

    **if** $I_i>B_i$) **then**

      Select predicted activity of biometric data;

    **else**

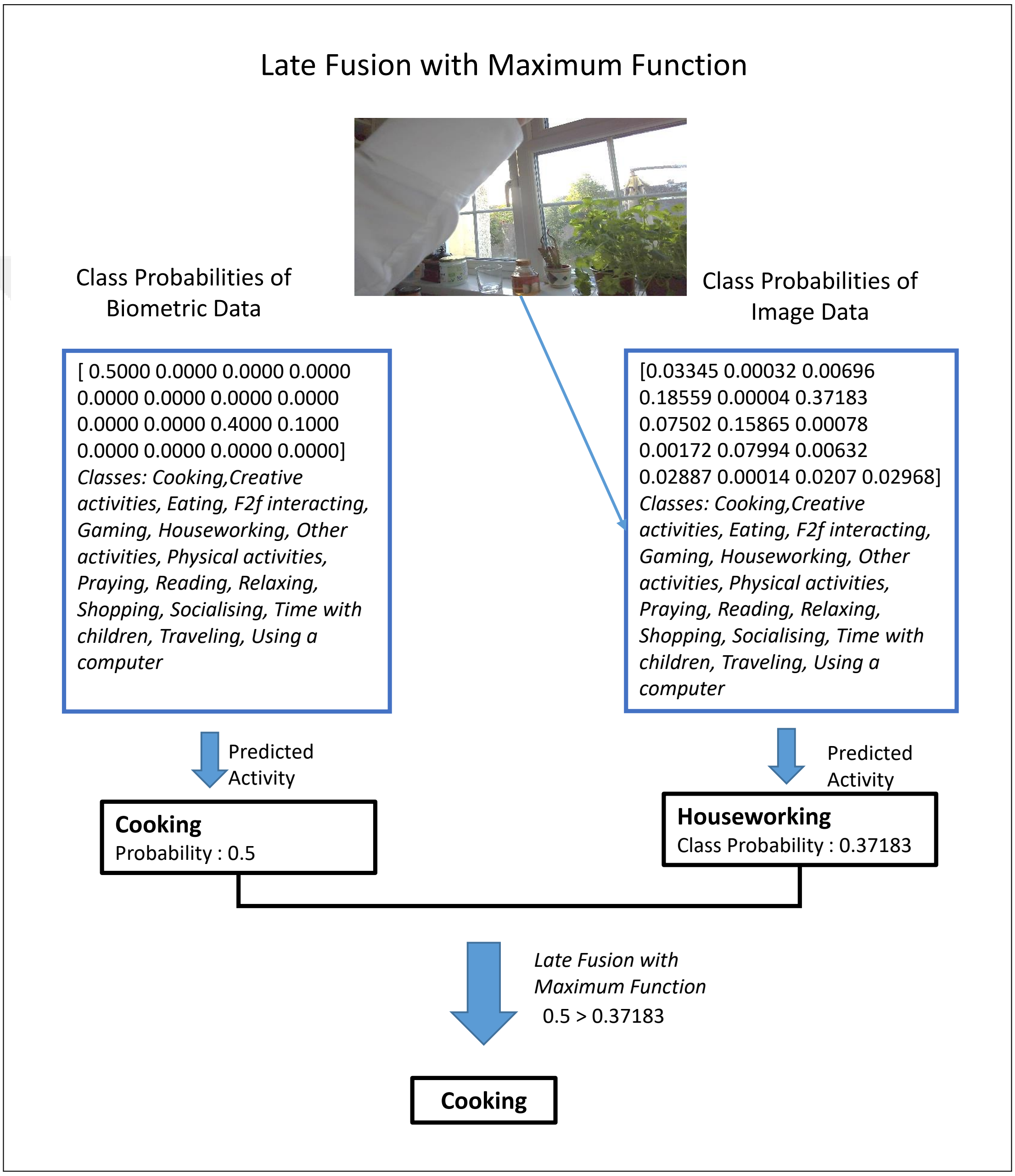
      Select one of predicted activity

**return** $resultvector(V)$;

data. Then, these class probabilities are compared with each other and the highest class probability score of the activity for biometric data and image data determines the activity class. According to Algorithm 2, maximum rule based late fusion works. The algorithm selects the class with higher probability among class probabilities of image and biometric data. In addition, the Figure 4.2 represents a sample for late fusion using maximum function. In the sample Figure 4.2, the highest probability scores are 0.5 for the biometric data and 0.37813 for the image data. Biometric data has a higher probability score than image data. As a result of applying late fusion with maximum function on these vectors, the function chooses the maximum score and the final decision is the same with the predicted activity of the biometric data. The main aim is to observe improvment on accuracy and we had better accuracy result with data fusion than learning activities separately.

In our work, we firstly applied maximum rule based late fusion on dataset containg 3 activity classes. The biometric data were trained with CatBoost classifier which is best for dataset. Then, 3 x n sized class probability vectors are obtained with 89.08% accuracy. The trained image data result is taken. It has also 3 x n sized class

Figure 4.2: Sample of Late Fusion method using maximum class probability for image data and biometric data



probability vectors and 91.79% accuracy. After applying maximum rule on the class probability scores of biometric data and image data, accuracy score increased from 89.08% to 93.72%.

Secondly, we applied same steps for the dataset containing 16 activity classes. The biometric data were trained with Random Forest classifier which is best for dataset. Then, 14 x n size class probability vectors are obtained with 60.77% accuracy. The missing two acitivities were filled with zero value. In this way, result vector for biometric data has 16 x n size. The trained image data result is taken. It has also 16 x n sized class probability vectors and 83.50% accuracy. After applying maximum rule on the class probability scores of biometric data and image data, accuracy score increased from 60.77% to 86.22%.

### 4.2.2 Classification-based Late Fusion

The other applied methodology, called classification based late fusion, is using a new classifier on the combination of class probability vectors for multimodal data. The summary of classification based late fusion strategy is shown in below Figure 4.3. Firstly, semi-synthetic features are generated for the biometric data. Feature selection is applied to the dataset which contains semi-synthetic features by using genetic algorithm. Then, classification algorithm which gives the best accuracy result is applied to the selected features and class probability vector are generated. Combined learning model is applied to image features and image annotations. This models generates class probability vector for image data. These vectors are concatenated and new vectors fed into a classifier.

In this thesis, two version of the lifelogging dataset containg 3 activity classes and 16 activity classes are used for classification based late fusion. It is applied with respecting to usage of classification based late fusion algorithm 3. Firstly, the dataset having 3 classes were trained with CatBoost and image data are trained with Convolutional Neural Network algorithms. These learning algorithms generate class probability vectors, which have 3 x n size, for each activity class. As a next, this 3 x n sized vectors from the image and biometric classifier were combined in a single 6 x n sized vector to form a multimodal decision vector. This new vector is trained

**Algorithm 3:** Classification Based Late Fusion Algorithm

**Input:** Class probability vectors for image data and biometric data

**Output:** Prediction vector

$I \leftarrow$ class probability vector for image data;

$B \leftarrow$ class probability vector for biometric data;

$N \leftarrow$ size of class probability vector;

$R \leftarrow$ class probability vector for combined results;

**for (** $i \leftarrow 1$ **to** $N$ **)** {
    $R_i \leftarrow I_i$.concatenate$(B_i)$

Given $R$ combined results vector to a new classifier $F(R)$;

$P \leftarrow$ Make a prediction

**return** $resultvector(P)$;

with three different machine learning models to give the final decision. We applied Random Forest algorithm with hyperparameter tuning and gives best accuracy results with 95.29%. Next, we trained new vector data with Support Vector Machine. It has 95.22% accuracy score. Lastly, we used Neural Network for activity learning on fused data. It gives 95.17% accuracy score and accuracy result lower than the results taken by previous applied algorithm. Same steps for the dataset containing 16 activity classes were applied with some differences. The biometric data are trained with Random Forest algorithm and image data are trained with Convolutional Neural Network algorithms.These learning algorithms generated 14 x n size and 16 x n size class probability vectors. The biometric data created 14 x n size vector because of two missing activities. The missing two acitivities were filled with zero value. In this way, result vector for biometric data has 16 x n size. Then, two 16 x n size vector were combined in a single 32 x n sized vector. This new vector is also trained with three different machine learning models to give the final decision. We applied Random Forest algorithm with hyperparameter tuning and gives best accuracy results with 88.18%. The result is improved from 60.77% to 88.18%. Next, we trained new vector data with Support Vector Machine. It has 84.89% accuracy score. Lastly, we used Neural Network for activity learning on fused data. It gives 81.04% accuracy score and accuracy result lower than the results taken by previous applied algorithm.

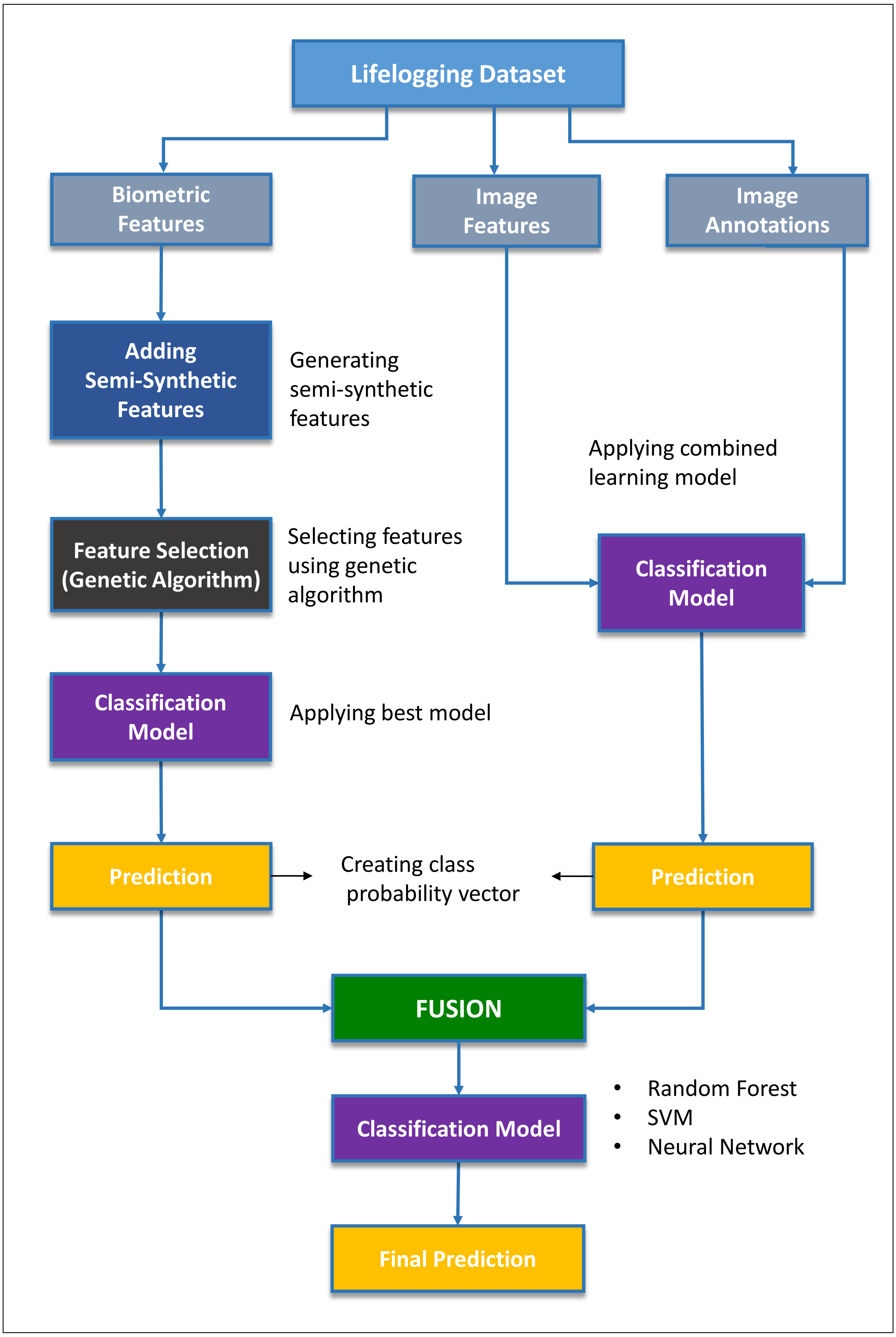



Figure 4.3: Applied late fusion method on NTCIR Lifelogging dataset

# CHAPTER 5

# EXPERIMENTS AND RESULTS

In this chapter, all applied methods and approaches to accuracy improvement are explained.

## 5.1 Evaluation of Classification Algorithms

This section mentions about details of applied classification algorithms for our activity learning problem.

### K-Nearest Neighbor Model

K-Nearest neighbor model as a classifier model is applied to the dataset. In this model, the selection of k value (number of nearest neighbors) and distance function is a crucial decision. Different distance functions exist for this model, like Euclidean distance, Manhattan distance, Hamming distance, and Minkowski distance.

The Euclidean distance measure distance between two points or based on Pythagoras' theorem. The distance is calculated according to formula 5.1]. The square root of the sum of the squared pair-wise distances of every dimension is calculated.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{5.1}$$

The Manhattan distance measures the distance based on gridlines. The Manhattan distance is also called city-block distance. Calculation technique is shown below formula 5.2]

$$\sum_{i=1}^{n} |x_i - y_i|. \tag{5.2}$$

The Minkowski distance is a general form of the Euclidean distance and the Manhattan distance. Below formula 5.3 shows how to calculate Minkowski distance. When p equals to 2, it is Euclidian distance. When p equals to 1, it is Manhattan distance.

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \tag{5.3}$$

Distance functions need a numeric attribute. In the data preprocessing part, categorical variables are converted to numeric values. Distance between data points are calculated using Euclidian formula 5.1, Manhattan formula 5.2 and Minkowski formula 5.3. The accuracy results are shown on Table 5.1. According to this table, Manhattan distance is chosen in this work to train the k Nearest Neighbor model. It has best accuracy result with 55.23%. Table 5.2 shows that distance as weight parameter for KNN model has better accuracy with 55.23% than the accuracy for uniform weight parameter. In addition, the selection of k parameter is done. For optimal number, all number from 2 to 100 is applied and compared their accuracy. For deciding the best parameters in the working model, GridSearchCV algorithm is applied as a parameter tuning method. It chooses the best smallest k among all the possible k values and distance function which is giving a minimum error rate. Relationship between k values and accuracy are shown in Figure 5.1 and figure 5.2.

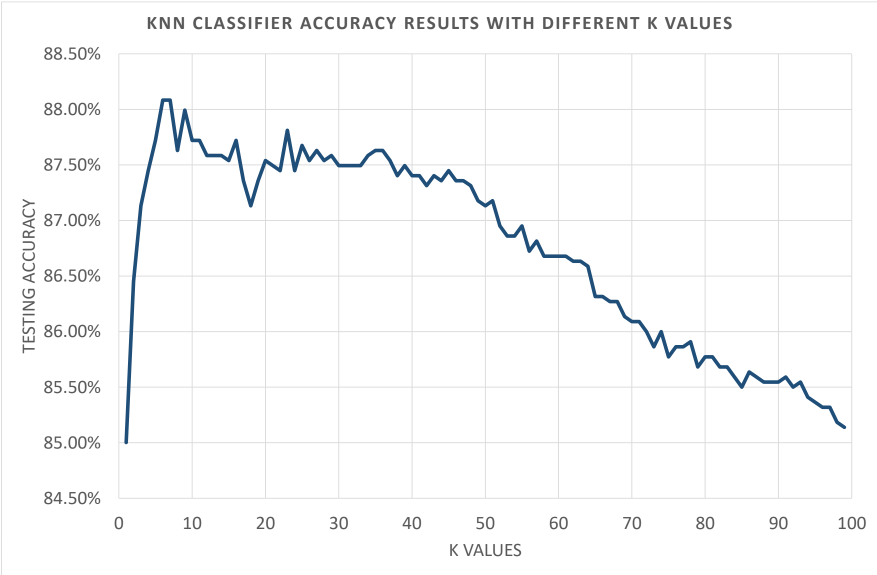



Figure 5.1: Different k values and accuracy result for the original dataset

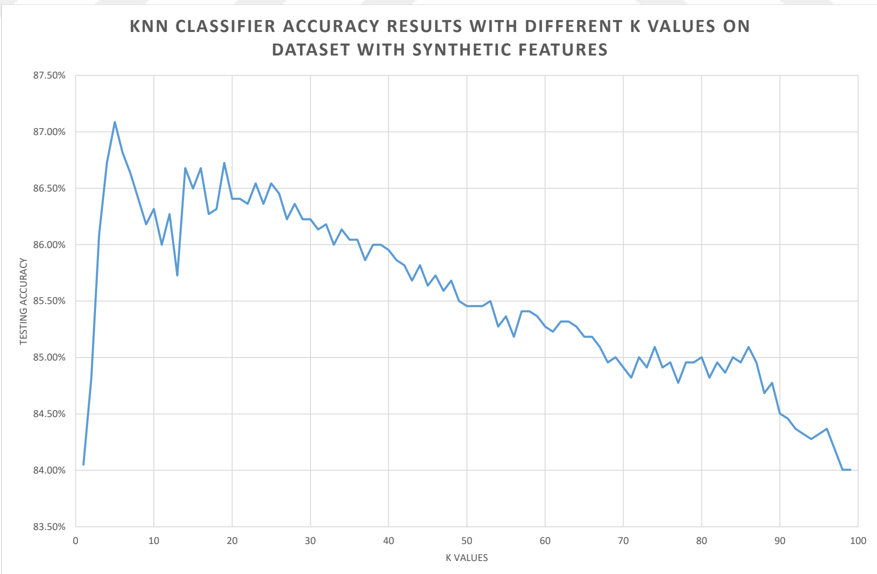



Figure 5.2: Different k values and accuracy result for the dataset, including semi-synthetic features

Table 5.1: Accuracy Result For Different Distance Metric Parameters

| **Distance Metric** | **Accuracy** |
|---|---|
| Manhattan | 55.23% |
| Euclidean | 54.93% |
| Hamming | 53.22% |
| Minkowski | 54.93% |

Table 5.2: Accuracy Result For Different Weight Parameters

| **Weights** | **Accuracy** |
|---|---|
| uniform | 46.53% |
| distance | 55.23% |

**Decision Tree Model**

Decision Tree model was used for activity classification on the NTCIR lifelogging dataset. To determine the split parameter of the decision tree model is important. Thus, different values were given to the model for finding an optimal number as shown in Figure 5.3 and four examined parameter values are stated on Table 5.3. Different max depth, min samples leaf, min samples split and criterion values were used to see the effect of accuracy changes.

Table 5.3: Examined Parameter Values for Decision Tree Algorithm

| **Parameter** | **Values** |
|---|---|
| max_depth | 50, 75, 100 and 150 |
| min_samples_leaf | 3, 4, 5 and 10 |
| min_samples_split | 5, 10, 15, 25 and 50 |
| criterion | gini and entropy |

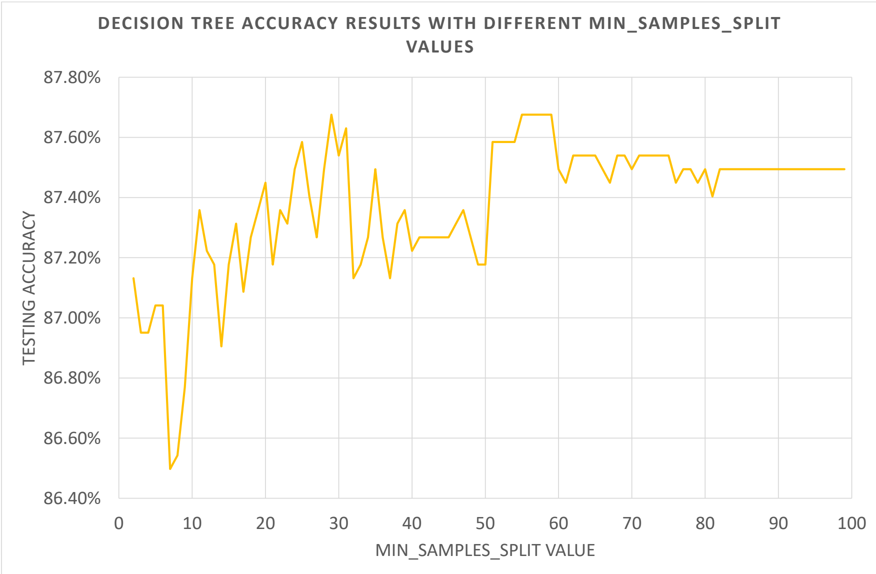



Figure 5.3: Different min sample split values and accuracy result for the original dataset

**Random Forest Model**

Random Forest model was attempted to the dataset. In this model, to determine the split parameter is important. For finding the best split parameter value, all number from 2 to 100 is applied to the model and compared their accuracy as shown in Figure 5.4 and figure 5.5. Additionally, other parameters of the model were tuned. Different max depth, min samples leaf, min samples split and criterion values are used to see the effect of accuracy changes. Below parameter table 5.4 shows examined values for each parameters.

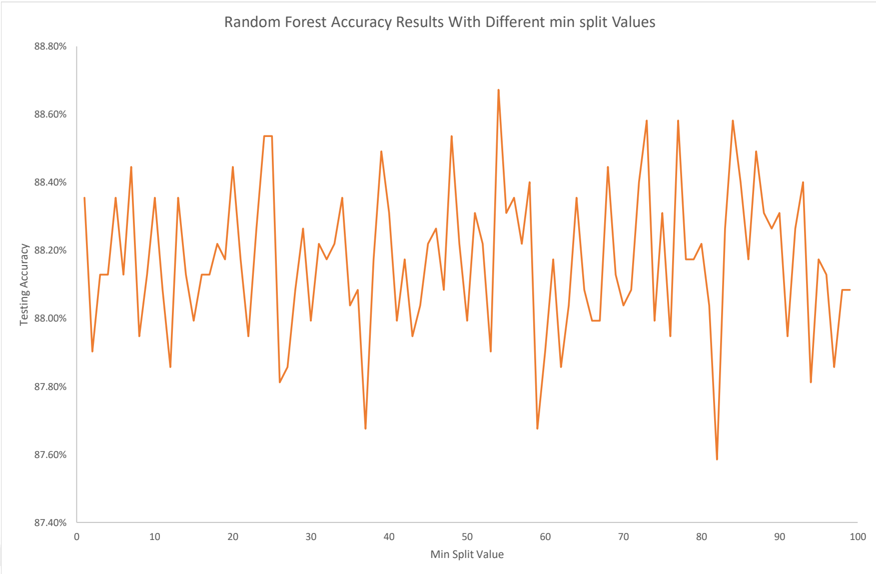



Figure 5.4: Different min sample split values and accuracy result for the original dataset

Table 5.4: Examined Parameter Values for Random Forest Classifier

| **Parameter** | **Values** |
|---|---|
| max_depth | 50, 75, 100 and 150 |
| min_samples_leaf | 3, 4, 5 and 10 |
| min_samples_split | 5, 10, 15, 25 and 50 |
| criterion | gini and entropy |

**Support Vector Machine**

Support Vector Machine was used to the dataset for activity classification problem. Different c parameter values were applied such as 0.01, 0.1, 0.5 and 1. As a result, the best accuracy result with 1 as c value were obtained.

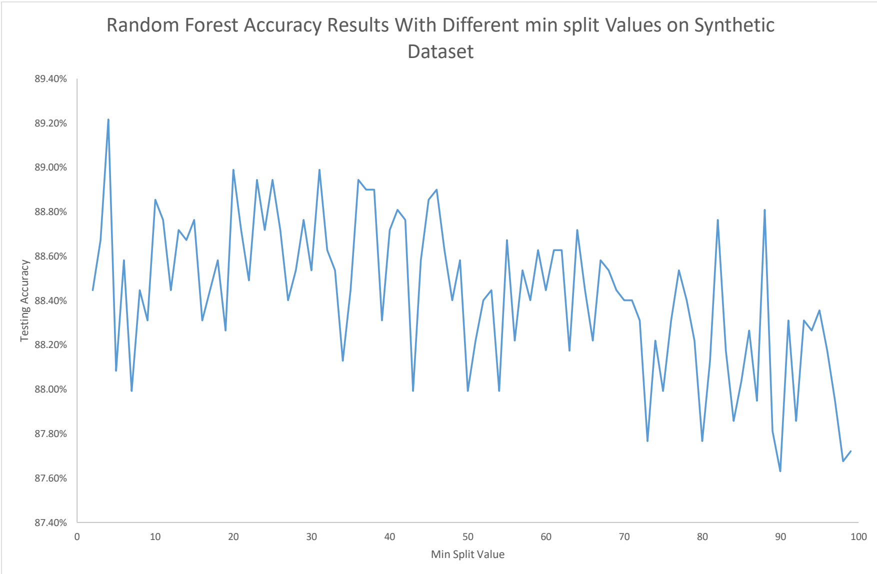



Figure 5.5: Different min sample split values and accuracy result for the dataset, including semi-synthetic features

**Neural Network Model**

Using Keras library in python, neural network model was applied to the dataset. Different kinds of neural network parameters were tuned. The lifelogging dataset variable has more than two categories in output. Therefore 'categorical cross-entropy' was chosen for loss function. To find an optimal set of weights, a kind of optimization algorithm, called Adam, was selected for its higher accuracy score with 87.75%. Table 5.5 shows all applied values for optimizer parameter and Table 5.6 shows all applied number of neurons. We obtained best accuracy score with 87.78% by using 100 neurons.

**XGBoost**

In this thesis, XGBoost algorithm was applied into NTCIR lifelogging dataset for activity prediction problem. To get better accuracy hyperparameter tuning was used and different learning rates, max depth, and the number of estimators' values were tested to see the effect of accuracy changes. Below Table 5.7 shows examined values

Table 5.5: Neural Network Accuracy Result For Different Optimizer

| **Optimizer** | **Accuracy** |
|---|---|
| SGD | 86.93% |
| RMSprop | 87.56% |
| Adagrad | 86.95% |
| Adadelta | 86.91% |
| Adam | 87.75% |
| Adamax | 87.23% |
| Nadam | 87.74% |

Table 5.6: Neural Network Accuracy Result For Neuron Number

| **Neurons** | **Accuracy** |
|---|---|
| 20 | 87.33% |
| 30 | 87.60% |
| 40 | 87.44% |
| 50 | 87.52% |
| 75 | 87.67% |
| 100 | 87.78% |

for each parameters and selected parameters are 0.1 for learning rate , 9 for max depth and 200 for number of iterations.

Table 5.7: Examined Parameter Values for XGBoost Algorithm

| **Parameter** | **Values** |
|---|---|
| Learning rate | 0.01, 0.03, 0.05 and 0.1 |
| Max depth | 2, 4, 5, 7, 8 and 9 |
| Number of estimators | 50, 100 and 200 |

The explanations of used parameters are stated below.

**Max depth** : It is depth of the tree [40].

**Number of estimators** : Number of trees to fit [40].

**Learning Rate** :It controls the model learning rate [40].

**CatBoost** CatBoost algorithm was used for activity prediction problem on the NTCIR lifelogging dataset. Besides, hyperparameter tuning of CatBoost algorithm was applied to get better accuracy. Different learning rates, max depth, and the number of leaf values were attempted to see the effect of accuracy changes. Below Table 5.8 shows examined values for each parameters and selected parameters are 0.1 for learning rate , 10 for max depth and 200 for number of iterations.

Table 5.8: Examined Parameter Values for CatBoost Algorithm

| **Parameter** | **Values** |
|---|---|
| Learning rate | 0.01, 0.05 and 0.1 |
| Max depth | 25, 50 and 75 |
| Iterations | 100, 500, 750 and 1000 |

The explanations of used parameters are stated below.

**Max depth** : It is depth of the tree. The range of supported values is any integer number from 1 to 16. It can be changed according to the type of the selected loss and the processing unit type [41].

**Iterations** : It states the maximum number of trees which can be built while solving the problems. The final number of trees should be equal or less than this parameter value [41].

**Learning Rate** :It controls the model learning rate [41].

**LightGBM** LightGBM algorithm was applied on the NTCIR lifelogging dataset for activity classification problem. Moreover, we done hyperparameter tuning on LightGBM algorithm. We tried different learning rates, max depth, and the number of leaf values to see the effect of accuracy changes. Below Table 5.9 shows examined values for each parameters and selected parameters are 0.05 for learning rate , 75 for max depth and 1000 for number of leaves.

Table 5.9: Examined Parameter Values for LightGBM Algorithm

| **Parameter** | **Values** |
|---|---|
| Learning rate | 0.01, 0.05 and 0.1 |
| Max depth | 25, 50 and 75 |
| Number of leaves | 100, 500, 750 and 1000 |

The explanations of used parameters are stated below.

**Learning Rate** :It controls the model learning rate [42].

**Max depth** : It is max depth limit of the tree [42].

**Number of leaves** : It is max number of leaves in one tree [42].

## 5.2 HyperParameter Tuning

Hyperparameter tuning is a technique that affects the performance of the applied method. Generally, changing method parameters have an impact on the accuracy and time performance of the result. Therefore, hyperparameter tuning is applied to classification models to get better accuracy results. GridSearchCV algorithm is used to train applied machine learning models with multiple combinations of training hyperparameters. It finds the best combination of parameters. Then, It generates an exhaustive set of hyperparameter combinations and train models on each combination.

According to the accuracy results of applied parameters for each machine learning model, selected best parameters are demonstrated below table 5.10.

Table 5.10: Parameters for Machine Learning Techniques

| No | Model | Parameters |
|---|---|---|
| 1 | k-Nearest Neighbor | No. of neighbors, n =32, weight function = distance, Distance Metric = Manhattan |
| 2 | Random Forest | No of Trees = 125, max depth of the tree = 100, min samples split = 4, min sample leaf = 4 |
| 3 | Decision Tree | criterion = entropy, max depth of the tree = 80, min sample leaf = 3 , min samples split = 5 |
| 4 | SVM | c=1 |
| 5 | Neural network | batch = 10, epoch=50, optimezer=Adam |
| 6 | XGBOOST | learning rate = 0.1, max depth = 9, no of estimators=200 |
| 7 | LightGBM | learning rate = 0.05, max depth = 75, no of estimators=200, no of leaves=1000 |
| 8 | CatBOOST | learning rate = 0.1, max depth = 10, no of iterations=200 |

## 5.3 Cross Validation

Cross validation is a common validation techniques in machine learning problems. In our study, we use k-fold cross validation technique to get accurate prediction results. This techniques is randomly partitioned sample data into k equal sized subsamples. Then, a single subsample is retained as the validation data for testing the model, and the remaining other subsamples are used as training data. We choose five as k value, which is common value for k. Iteratively, it repeats five times with each of the four subsamples used exactly once as the validation data. After we get five results from the folds, we calculate average results for estimation. K-fold cross validation is a good approach because all observations are used for both training and validation, and each observation is used for validation exactly once.

I. Divide sample data into k parts.

II. Use k-1 parts of data for training and the rest of them for the test.

III. Repeat the procedure k times, rotating the test set.

IV. Determine an expected performance metric based on the results across the iterations.

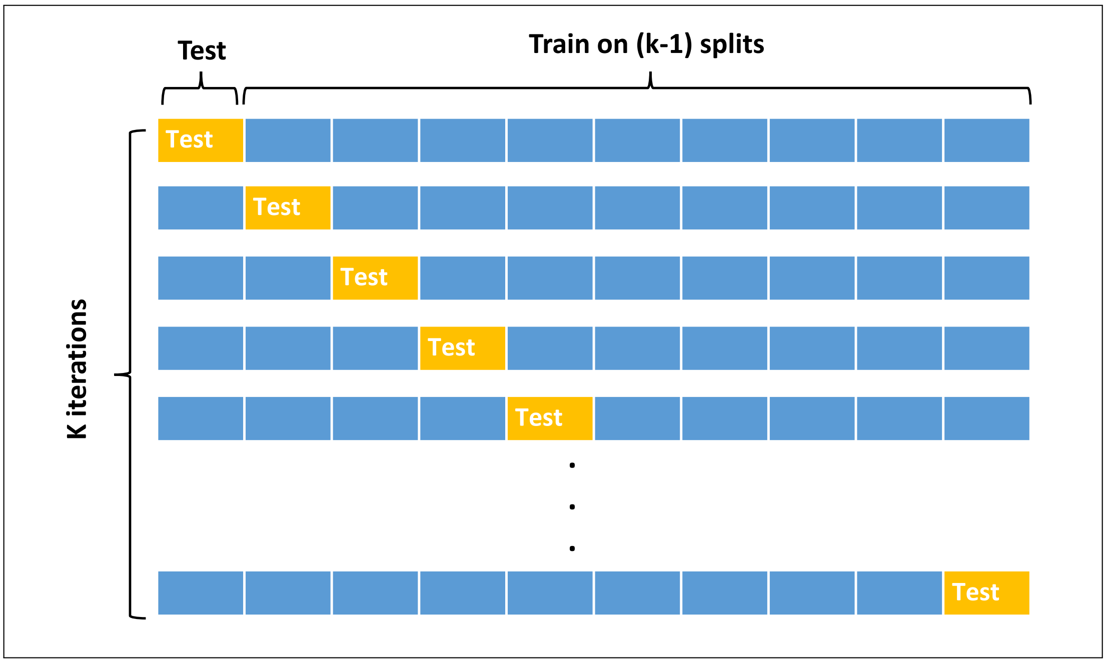



Figure 5.6: K-fold Cross Validation Overview

## 5.4 Comparison of Machine Learning Models For 3 Classes and 14 Classes

In this study, the algorithms are implemented by default parameter values and hyper-parameter tuning values on three different versions of the dataset, namely 12 features with 3 classes, 30 semi-synthetic features with 3 classes and 12 features with 16 classes and the accuracy results are shown in below Tables 5.11, 5.12, 5.13 and 5.14 respectively. Table 5.11 shows accuracy results for 3 classes dataset without hyper-parameter tuning. CatBoost works better both on the original features and the semi-synthetic features with 89.08% and 89.17% respectively. Also, this result shows the improvement of the accuracy with semi-synthetic features. On the other hand, Catboost works better on the original features while Random Forest works better on the semi-synthetic features for 12 original features. The results based on applied proposed algorithms are given on Table 5.12. Then, the hyperparameter tuning is applied on the same datasets. Table 5.13 shows accuracy result for 3 classes dataset with hyperparameter tuning. CatBoost has best accuracy score with 89.08% for the original features but Random forest give better result with 89.22% for the semi-synthetic features. Moreover, Table 5.14 shows accuracy scores with hyperparameter tuning for 12 original features. CatBoost has best accuracy score with 89.08% for the original features while Random forest give better result with 60.77% for the original features.

According to these accuracy tables, hyperparameter tuning gets better performance results because it tries the algorithm with different parameters and finds appropriate parameter values for the applied dataset. In addition, we provide confusion matrices for three different versions of the dataset in Figures 5.7, 5.8 and 5.9

Table 5.11: Accuracy comparison table of applied algorithm for 3 activity classes

| | **Accuracy Result For 3 Classes** | |
|---|---|---|
| | **12 Original Features** | **30 Semi-Synthetic Features** |
| **K-Nearest Neighbor** | 87.72% | 87.09% |
| **Naieve Bayes** | 66.15% | 62.35% |
| **Decision Tree** | 87.18% | 86.77% |
| **Random Forest** | 88.63% | 88.31% |
| **SVM** | 86.77% | 86.63% |
| **Neural Network** | 87.80% | 87.97% |
| **XGBOOST** | 88.36% | 88.31% |
| **LightGBM** | 88.99% | 88.94% |
| **Catboost** | 89.08% | 89.17% |

Table 5.12: Accuracy comparison table of applied algorithm for 12 original features

| | **Accuracy Result For 12 Original Features** | |
|---|---|---|
| | **3 Classes** | **14 classes** |
| **K-Nearest Neighbor** | 87.72% | 45.87% |
| **Naieve Bayes** | 66.15% | 5.88% |
| **Decision Tree** | 87.18% | 60.13% |
| **Random Forest** | 88.63% | 60.90% |
| **SVM** | 86.77% | 36.07% |
| **Neural Network** | 87.80% | 41.73% |
| **XGBOOST** | 88.36% | 44.81% |
| **LightGBM** | 88.99% | 51.64% |
| **Catboost** | 89.08% | 50.81% |

Table 5.13: Accuracy comparison table of applied algorithm for 3 activity classes with hyperparameter tuning

| | **Accuracy Result For 3 Classes** | |
|---|---|---|
| | **12 Original Features** | **30 Semi-Synthetic Features** |
| **K-Nearest Neighbor** | 88.08 % | 87.09 % |
| **Naieve Bayes** | 66.15 % | 62.35 % |
| **Decision Tree** | 87.66 % | 87.81 % |
| **Random Forest** | 88.67 % | 89.22 % |
| **SVM** | 85.72 % | 85.28 % |
| **Neural Network** | 87.80 % | 87.97 % |
| **XGBOOST** | 88.91 % | 88.96 % |
| **LightGBM** | 88.99 % | 88.94 % |
| **Catboost** | 89.08 % | 89.17 % |

Table 5.14: Accuracy comparison table of applied algorithm for 12 original features with hyperparameter tuning

| | **Accuracy Result For 12 Original Features** | |
|---|---|---|
| | **3 Classes** | **14 classes** |
| **K-Nearest Neighbor** | 88.08 % | 52.92 % |
| **Naieve Bayes** | 66.15 % | 5.88 % |
| **Decision Tree** | 87.66 % | 50.54 % |
| **Random Forest** | 88.67 % | 60.77 % |
| **SVM** | 85.72 % | 52.79 % |
| **Neural Network** | 87.80 % | 41.73% |
| **XGBOOST** | 88.91 % | 55.48 % |
| **LightGBM** | 88.99 % | 58.10% |
| **Catboost** | 89.08 % | 50.81 % |

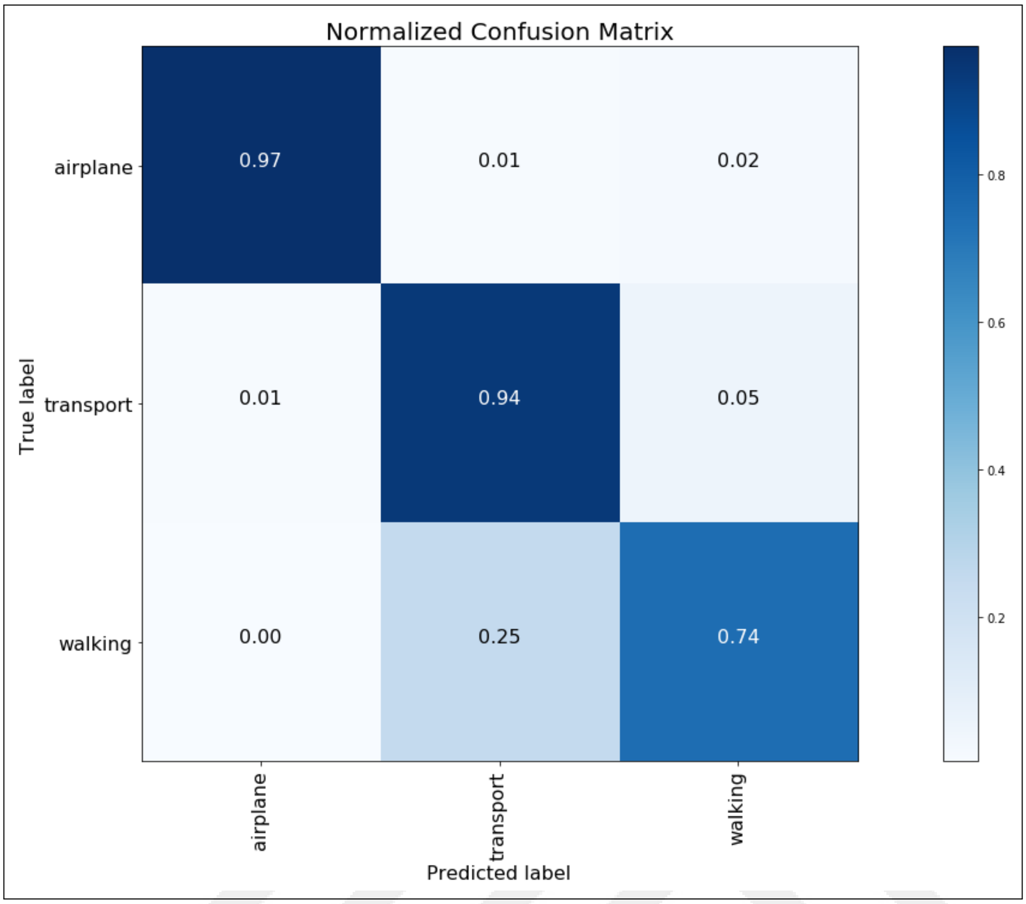



Figure 5.7: Confusion Matrix for original features with 3 activities

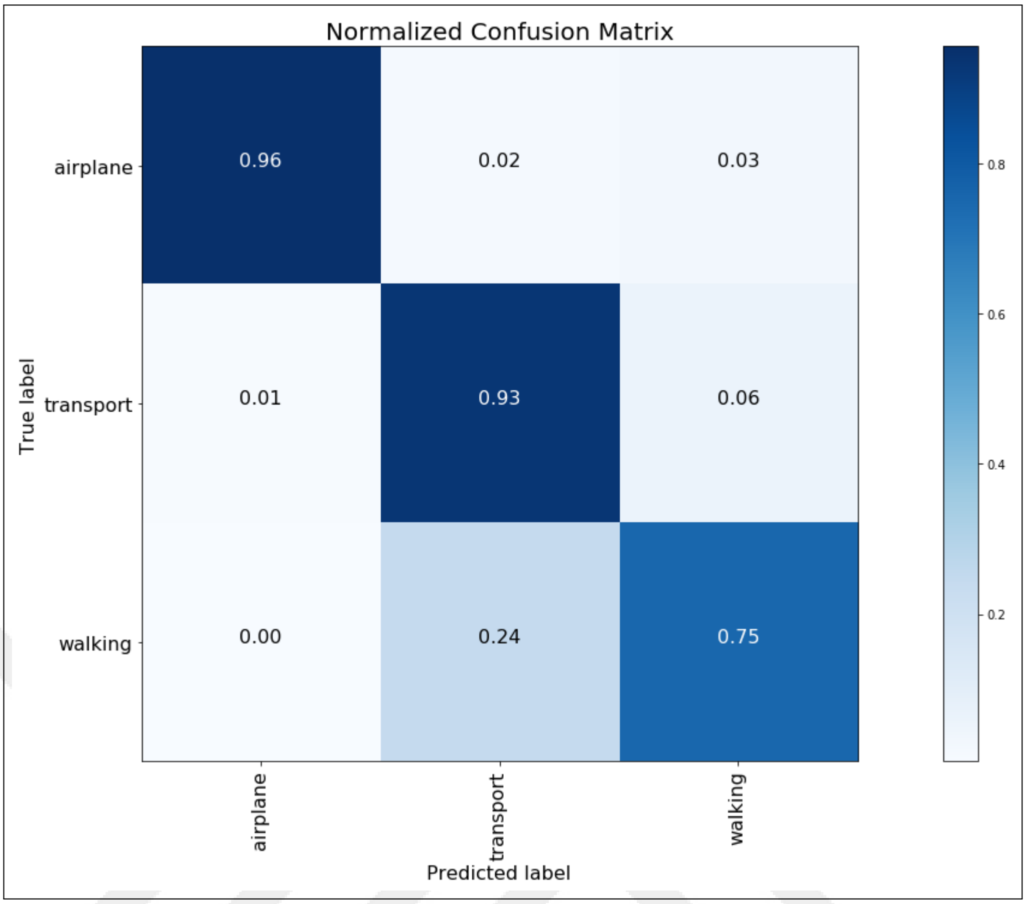



Figure 5.8: Confusion Matrix for semi-synthetic features with 3 activities

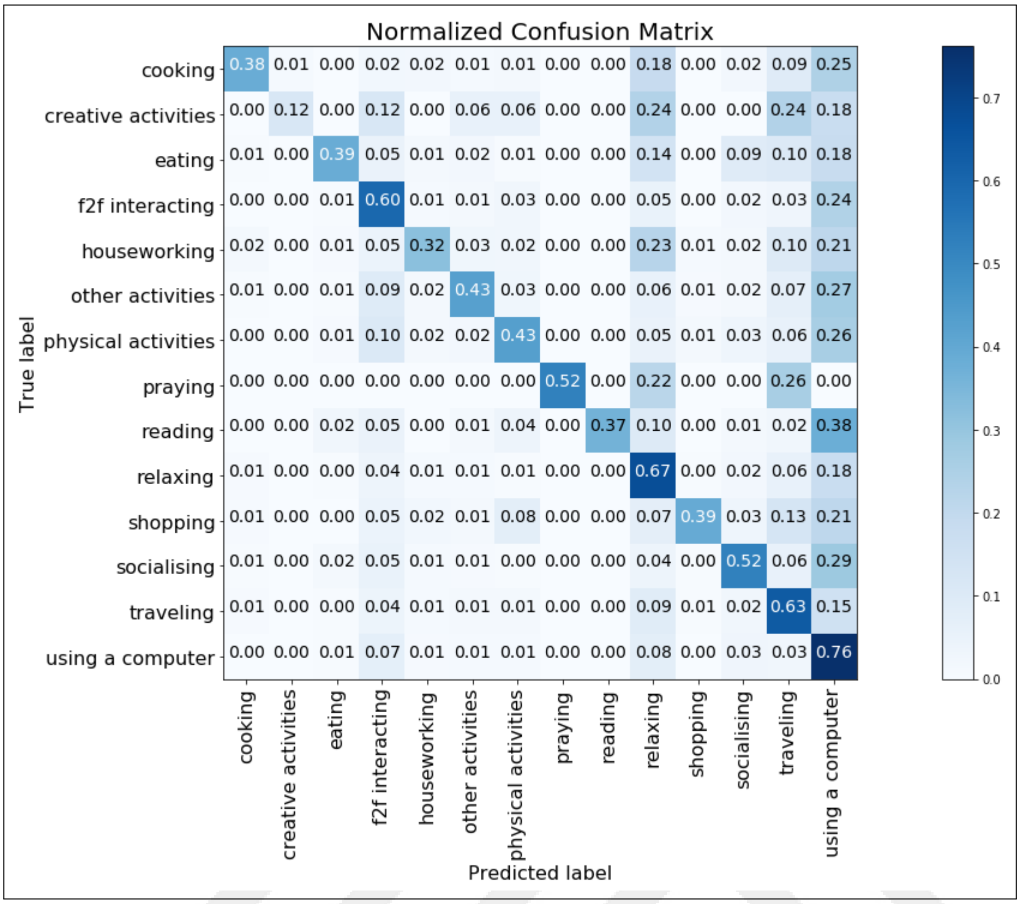



Figure 5.9: Confusion Matrix for original features with 14 activities

## 5.5 Evaluation of Feature Selection Algorithms

The MOEA Framework is an open-source java library for multi-objective evolutionary algorithms (MOEAs). The MOEA framework supports the implementation of genetic algorithms. It provides a fast and reliable implementation of genetic algorithms. Also, a parallel run of algorithms is supported. For these reasons, the MOEA framework is used at feature selection with genetic algorithm steps.

According to the result of the applied genetic algorithm, we see that semi-synthetic features are selected. It means that semi-synthetic features that are created by using original data in the dataset are meaningful and important for the improvement of accuracy. Below table 5.15 shows sample generated feature set and accuracy result. The selected features are represented with 1 and unselected features are represented with 0.

Best accuracy result is 90,91% taken by chosen 11 features among 30 features. Selected features are represented on Table 5.16. Five original features are selected namely, GLU, GSR, HEART_RATE, SLEEP_DURATION and WEIGHT. Six semi-synthetic features are selected namely, bp_avg, sleep_sd_m, sd_sqr, sc_sqr, bm_hr_st and glu_weight_m. The number of semi-sythetic features are greater than the number of original features. This shows that semi-synthetic features have complementary information for our dataset.

Table 5.15: Generated Feature Set Results For Executed Genetic Algorithms

| **Generated Feature Set** | **Accuracy (%)** |
|---|---|
| 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 | 90,91373 |
| 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 1 1 0 0 1 1 0 0 | 89,17349 |
| 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 | 89,45349 |
| 1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 | 86,95075 |
| 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 1 1 0 1 1 1 | 89,09981 |
| 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 | 89,22423 |
| 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 0 | 89,26590 |
| 0 1 0 1 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 | 89,18636 |
| 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 | 89,24563 |
| 1 0 0 0 1 1 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 1 0 | 86,05583 |
| 1 0 0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 | 89,34004 |
| 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 | 89,23160 |
| 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 0 0 0 1 0 1 | 89,29092 |
| 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 0 | 89,30962 |
| 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 1 1 0 | 89,48181 |
| 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 0 | 89,27385 |
| 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 | 89,27447 |
| 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | 88,92202 |
| 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 | 89,34032 |

1:GLU, 2: WEIGHT, 3: BP_LOW, 4: BP_HIGH,
5: CALORIES, 6: STEPS, 7: SLEEP_DURATION, 8: SLEEP_SCORE,
9: GSR, 10: HEART_RATE, 11: SKIN_TEMP, 12: BP_AVG,
13:BP_M, 14: SLEEP_SD, 15: SLEEP_SD_M,
16: SD_SQR, 17: SC_SQR, 18:BM_HR_ST, 19: BM_HR_ST_M,
20: HRT_SQR, 21: SKT_SQR, 22: BM_GSR_ST, 23: BM_GSR_HR,
24: BM_GSR_ST_M, 25: BM_GSR_HR_M, 26: BM_HR_C,
27: BM_HR_C_M,28: GLU_WEIGHT, 29: GLU_WEIGHT_M,
30: MUSIC

Table 5.16: Selected Features By Genetic Algorithms

| No | Selected Features |
|---|---|
| 1 | GLU |
| 2 | WEIGHT |
| 3 | SLEEP_DURATION |
| 4 | GSR |
| 5 | HEART_RATE |
| 6 | ( BP_LOW + BP_HIGH)/2 as bp_avg |
| 7 | ( SLEEP_DURATION * SLEEP_SCORE) as sleep_sd_m |
| 8 | ( sleep_duration * sleep_duration) as sd_sqr |
| 9 | ( SLEEP_SCORE * SLEEP_SCORE) as sc_sqr |
| 10 | ( HEART_RATE + SKIN_TEMP)/2 as bm_hr_st |
| 11 | ( GLU * WEIGHT) as glu_weight_m |

## 5.6 Evaluation of Biometric Data and Image Data Fusion

Late fusion technique was applied to activity prediction results of biometric data and image data for 3-class and 16-class lifelogging dataset. Two kinds of late fusion were used: maximum function and classification based fusion.

Firstly, the dataset containing 3 activity classes was split as train and test data for image and biometric data. The best accuracy of biometric data for 3 classes was taken with CatBoost algorithm. Therefore, CatBoost algorithm was applied to the train biometric data and the test data accuracy is 89.08% which is represented in Table 5.17 for each activity. Then, image data were trained which is done by another person. The test data accuracy of image data is 91.79% which is shown in Table 5.18 for each activity. As a next step, class probabilities of the biometric data and the image data were calculated. Each class probability was compared to image data and biometric data. The class which has maximum probability was selected as a result of late fusion. Lastly, using test activity label and fused activity label test accuracy is estimated 93.72% as shown in Table 5.19 and the Figure 5.10 shows the comparison of accuracy results for each activities. According to Tables 5.19 ,5.18, 5.17, the late fusion improves total accuracy result from 89.08% to 93.72%. Accuracy score for transport increased from 96.05% to 97.21% and accuracy score for walking sharply increased from 60.50% to 91.94%. On the other hand, accuracy score for airplane decreased from 97.31% to 91.94%. The same steps were applied on the dataset including 16 activity classes but the best accuracy of biometric data for 14 classes was taken with Random Forest. Thus, Random Forest was used as a classifier and the test data accuracy of them is 60,77% which is represented in Table 5.20 for each activity. The image data were trained and the test data accuracy of image data is 83,50 % which is shown in Table 5.21 for each activity. Then, class probabilities of the biometric data and the image data were again calculated. Each class probability was compared to image data and biometric data. The class which has maximum probability was selected as a result of late fusion. Lastly, test accuracy is estimated 86,22% by using test activity label and fused activity label. Accuracy results are shown in Table 5.22 and comparison of accuracy results for each activities are given in the Figure 5.11. The late fusion improves total accuracy result from 60.77% to 86.22%. Accuracy score decreased with late fusion for only two activities namely, praying and reading. However, the late fusion has benefits to improve total accuracy in our problem.

Table 5.17: Biometric Data Result Table For 3 Activity Classes Using CatBoost Classifier

| **Activities** | **Accuracy** |
|---|---|
| airplane | 97.31 |
| transport | 96.05 |
| walking | 60.50 |
| **Total** | **89.08** |

Table 5.18: Images Data Result Table For 3 Activity Classes Using Combined Model [29]

| **Activities** | **Accuracy** |
|---|---|
| airplane | 87.63 |
| transport | 96.32 |
| walking | 79.83 |
| **Total** | **91.79** |

Table 5.19: Fusion Of Biometric and Image Results For 3 Activity Classes Using Maximum Class Prediction Rate

| **Activities** | **Accuracy** |
|---|---|
| airplane | 91.94 |
| transport | 97.21 |
| walking | 83.75 |
| **Total** | **93.72** |

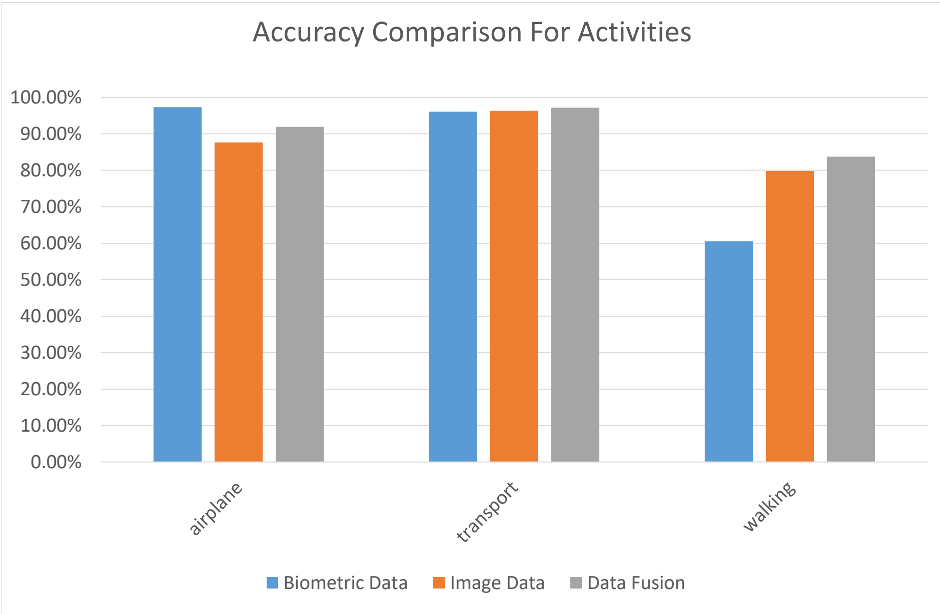



Figure 5.10: Accuracy Comparison For Biometrics, Image and Fused Data Using 3 Classes

Table 5.20: Biometric Data Result Table For 14 Activity Classes Using Random Forest Classifier

| **Activities** | **Accuracy** |
|---|---|
| cooking | 34.04% |
| creative activities | 13.04% |
| eating | 40.00% |
| f2f interacting | 56.84% |
| houseworking | 34.44% |
| other activities | 42.68% |
| physical activities | 43.88% |
| praying | 66.67% |
| reading | 42.39% |
| relaxing | 67.63% |
| shopping | 37.41% |
| socialising | 50.64% |
| traveling | 61.47% |
| using a computer | 77.15% |
| **Total** | **60.77%** |

Table 5.21: Images Data Result Table For 14 Activity Classes Using Combined Model [29]

| **Activities** | **Accuracy** |
|---|---|
| cooking | 59.65% |
| creative activities | 69.57% |
| eating | 41.05% |
| f2f interacting | 69.65% |
| houseworking | 35.14% |
| other activities | 58.47% |
| physical activities | 78.03% |
| praying | 33.33% |
| reading | 13.04% |
| relaxing | 94.46% |
| shopping | 59.86% |
| socialising | 74.63% |
| traveling | 98.10% |
| using a computer | 95.67% |
| **Total** | **83.50%** |

Table 5.22: Fusion Of Biometric and Image Results For 14 Activity Classes Using Maximum Class Prediction Rate

| **Activities** | **Accuracy** |
|---|---|
| cooking | 67.72% |
| creative activities | 73.91% |
| eating | 54.53% |
| f2f interacting | 75.60% |
| houseworking | 43.36% |
| other activities | 63.50% |
| physical activities | 80.46% |
| praying | 46.67% |
| reading | 27.17% |
| relaxing | 95.28% |
| shopping | 64.17% |
| socialising | 79.11% |
| traveling | 98.65% |
| using a computer | 96.68% |
| **Total** | **86.22%** |

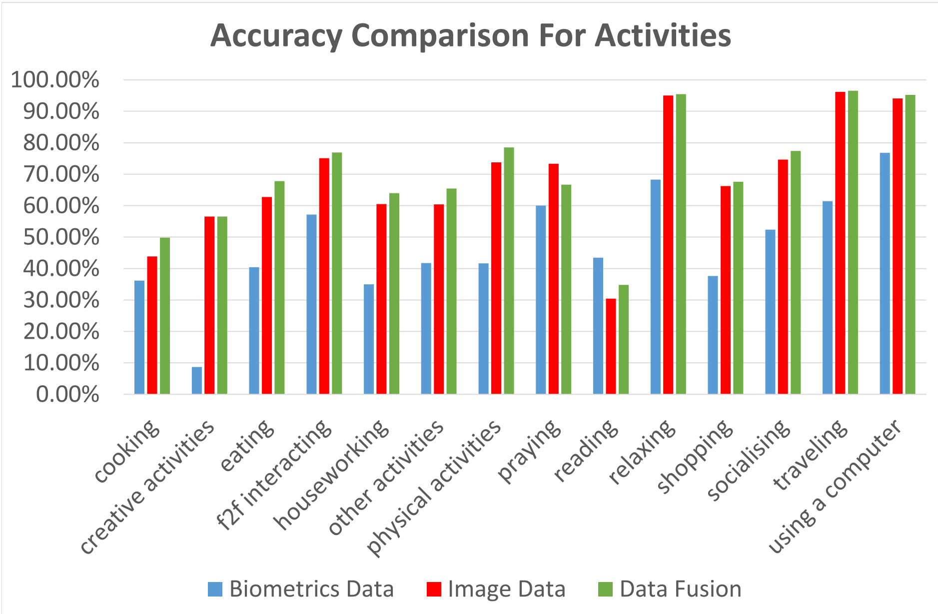



Figure 5.11: Accuracy Comparison For Biometrics, Image and Fused Data Using 14 classes

Secondly, the dataset containing 3 activity classes was split as train and test data for image and biometric data. CatBoost algorithm was applied to the train biometric data for predicting test values. According to prediction results, the class probability vector was generated. It has 3 activities. Thus, 3 x n size vector was created. For image data, generating the class probability vector was taken from another study. Then, these class probability vectors for biometric train data and image train data results were concatenated in the same minutes of images for creating new train data. After this process, we created new *n x 6* vector from two *n x 3* vectors.Then support vector machine, random forest, and artificial neural network were applied on new train data and tested with new test data. We applied Random Forest algorithm with hyperparameter tuning and gives best accuracy results with 95.29%. Then, we trained new vector data with Support Vector Machine. It gives 95.22% accuracy score. Lastly, we used Neural Network for activity learning on fused data. It has 95.17% accuracy score and accuracy result lower than the results taken by previous applied algorithm.

The results are given in Table 5.23. Same steps for the dataset containing 16 activity classes were applied with some differences. Biometric data have 14 classes because user 1 has missing 2 activities: eating and time with children. The missing activities were filled with zero probability to get the same size vector. Figure 5.12 shows a sample concatenation of class probabilities. In the order of activity names are given in vector: *"cooking","creative activities","eating","f2f interacting" ,"gaming" ,"houseworking" ,"other activities" ,"physical activities" ,"praying" ,"reading" , "relaxing" ,"shopping" ,"socialising" ,"time with children" ,"traveling" ,"using a computer"*. Test data made the same approach. After this process, we created new *n x 32* vector from two *n x 16* vectors. Then support vector machine, random forest, and artificial neural network were applied on new train data and tested with new test data, as shown in Figure 4.3. Finally, higher accuracy of the fused result is taken using Random Forest classifier with 88,18 % accuracy as represented on Table 5.24. It's better than accuracy which is calculated separately using biometric data and image data. However, using maximum prediction probabilities gets better accuracy than training data which is concatenated prediction probabilities.

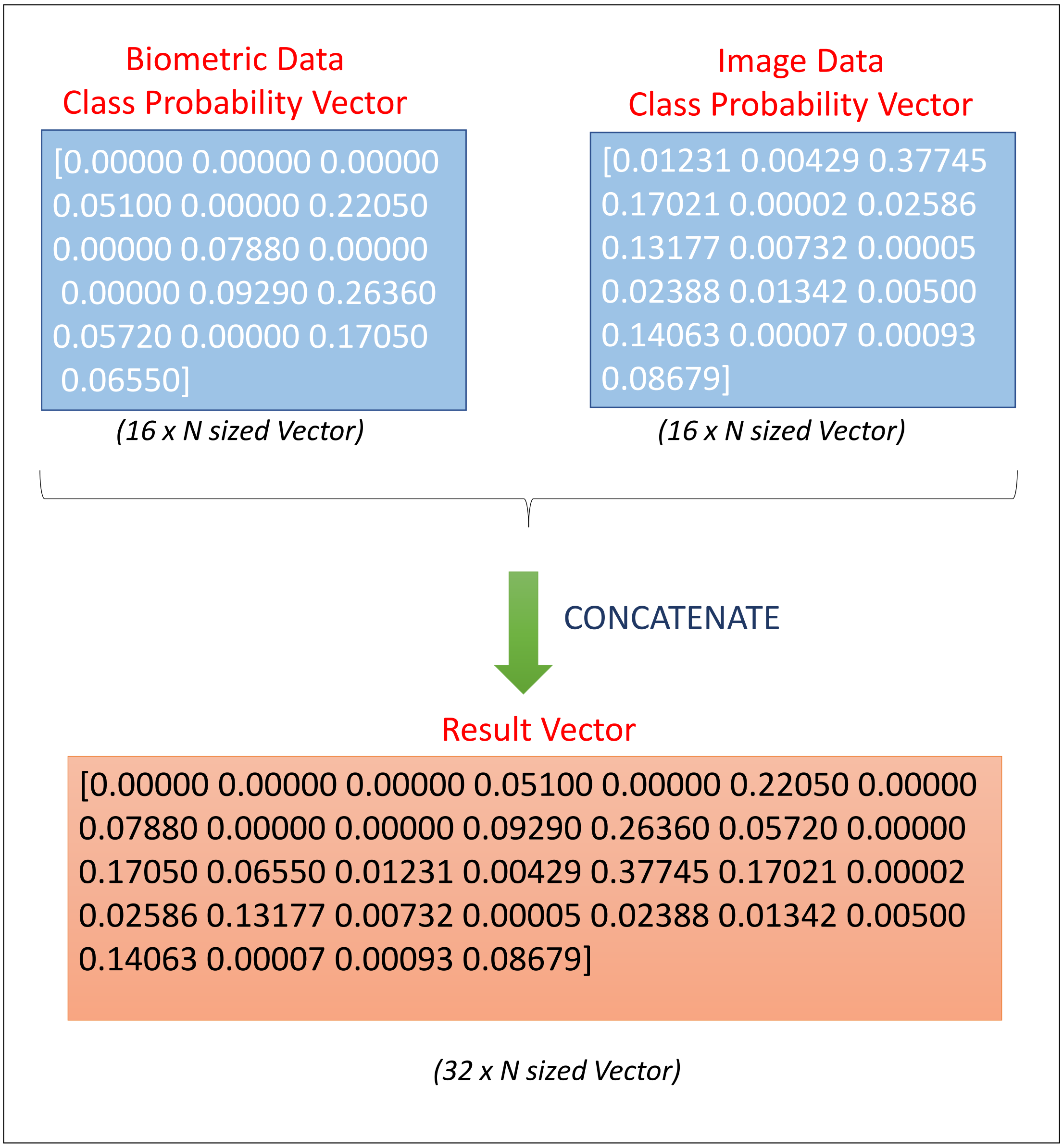



Figure 5.12: Concatenation of class probability vector for sample biometric data and image data

Table 5.23: Fusion Biometric and Image Results For 3 Activity Classes Using A New Classifier

| Applied Method | Accuracy |
|---|---|
| Random Forest | 95.29% |
| Support Vector Machine | 95.22% |
| Artificial Neural Network | 95.17% |

Table 5.24: Fusion Biometric and Image Results For 16 Activity Classes Using A New Classifier

| Applied Method | Accuracy |
|---|---|
| Random Forest | 88.18% |
| Support Vector Machine | 84.89% |
| Artificial Neural Network | 81.04% |

# CHAPTER 6

# CONCLUSION

In this thesis, human activity learning is studied using multimodal NTCIR lifelogging dataset for quality of life management. This dataset, containing the daily lifelogs of two users and their activities, is described and the effect on daily lifelogs is analyzed in detail. The daily lifelogging data are in a file in XML format. As a result, we extract the data from the XML format and import it into a newly created lifelogging database. Using the extracted data, we create a subset of data containing 12 features with 3 activities, namely airplane, transportation, and walking.

The dataset contains multimodal data: biometric data, image data, location data, and music data. We focus on the relationship between biometric data and user activity. To do this, we apply well-known machine learning algorithms, such as Naive Bayes, KNN, Decision Tree and Random Forest, as well as some state-of-art boosting algorithms such as XGBoost, LightGBM, and CatBoost. The results of the experiments show that the CatBoost algorithm has a more accurate result than the other applied algorithms.

In addition, we analyze biometric data to learn new inferences. There may be a relationship between the biometric data, as well as a relationship between the biometric data and the activity. To see this relationship, we create semi-synthetic features from existing ones by applying certain mathematical operations such as sum, multiplication, average and square. In this way, the number of features increases from 12 to 30. Then, we apply the same machine learning algorithms to the dataset, including existing and semi-synthetic features. The Random Forest algorithm gives an accuracy of 89.21% for activity learning. The use of semi-synthetic features shows that they have a positive impact on accuracy over accuracy results without the use of semi-synthetic features.

With a growing number of features, feature selection is needed to find meaningful features. We implement the genetic algorithm to reduce the unrelated features and use the Random Forest model for training of selected features. According to the result of the genetic algorithm, the accuracy rate was increased to 90.91% and 11 features were selected from 30 features. Some semi-synthetic features are formed, and the effect of the semi-synthetic features on the dataset is shown again. Moreover, we find that the exclusive use of significant features predicts accurately.

The NTCIR dataset has only 3 activities. These activities are modified using image data by the other work and new activities are defined. The number of activities is therefore increased to 16. We identify new activities in the dataset used. After adding new activities, the missing activity rate on the dataset decreases and more data are trained with the same algorithms to compare the results and see the changes. We get the best result in terms of accuracy with the Random Forest algorithm. Accuracy result reduced to 60.77%. This sharp decrease is the expected result. Because the number of the predicted classes is greater than the number of classes in the first version of the dataset. In addition, the size of the trained data increases and the model needs to learn more. Thus, the error rate also increases.

Finally, late fusion methods are applied to the results of image data and biometric data. Two late fusion methods are examined on the dataset. One of them is the late fusion based on a maximum function. Each result of prediction result of image data and biometric data is compared. The data type for which the accuracy rate is highest is selected from the image data results and the biometric data results. Due to the late fusion method, the accurate rate is increased. The other method is to combine the results of image and biometric data and to classify the combined result data using new models. As a first method, the second method improves accuracy. Late fusion learns new inferences from different sources and makes improvements to achieve better results than image data classification and biometric classification.

In future work, different fusion methods can be used for image data and biometric data. We use late fusion methods based only on a new classifier and a maximum function. Instead of a maximum function, the average function, the sum function and the weight function can be preferred in case of late fusion. Moreover, we use the Random Forest algorithm on a genetic algorithm to obtain an accuracy score with selected features. Different machine learning algorithms can be applied to the genetic algorithm for feature selection.

## REFERENCES

[1] A. Duane, R. Gupta, L. Zhou, and C. Gurrin, "Visual insights from personal lifelogs," in *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo*, pp. 386–389, 2016.

[2] M. Dodge and R. Kitchin, "'outlines of a world coming into existence': pervasive computing and the ethics of forgetting," *Environment and Planning B*, vol. 34, no. 3, pp. 431–445, 2007.

[3] M. Mohamad, D. Nasien, H. Hassan, and H. Haron, "A review on feature extraction and feature selection for handwritten character recognition," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, pp. 204–212, 2015.

[4] H. Motoda and H. Liu, "Feature selection, extraction and construction," *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, vol. 5, no. 67-72, p. 2, 2002.

[5] S. Markovitch and D. Rosenstein, "Feature generation using general constructor functions," *Machine Learning*, vol. 49, no. 1, pp. 59–98, 2002.

[6] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1424–1437, 2004.

[7] R. Leardi, R. Boggia, and M. Terrile, "Genetic algorithms as a strategy for feature selection," *Journal of chemometrics*, vol. 6, no. 5, pp. 267–281, 1992.

[8] B. W. Silverman and M. C. Jones, "E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951)," *International statistical review/revue internationale de statistique*, pp. 233–238, 1989.

[9] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[10] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[13] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, “Multisensor data fusion for physical activity assessment,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 687–696, 2011.

[14] A. Fleury, M. Vacher, and N. Noury, “Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results,” *IEEE transactions on information technology in biomedicine*, vol. 14, no. 2, pp. 274–283, 2009.

[15] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[16] A. V. Dorogush, V. Ershov, and A. Gulin, “Catboost: gradient boosting with categorical features support,” *arXiv preprint arXiv:1810.11363*, 2018.

[17] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[18] V. Elangovan, A. Alkilani, and A. Shirkhodaie, “A multi-modality attributes representation scheme for group activity characterization and data fusion,” in *2013 IEEE International Conference on Intelligence and Security Informatics*, pp. 85–90, IEEE, 2013.

[19] L. I. Kuncheva, “A theoretical study on six classifier fusion strategies,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[20] H. Gunes and M. Piccardi, "Affect recognition from face and body: early fusion vs. late fusion," in *2005 IEEE international conference on systems, man and cybernetics*, vol. 4, pp. 3437–3443, IEEE, 2005.

[21] V. Bush *et al.*, "As we may think," *The atlantic monthly*, vol. 176, no. 1, pp. 101–108, 1945.

[22] L. M. Zhou, C. Gurrin, and Z. Qiu, "Zhiwo: Activity tagging and recognition system for personal lifelogs," in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pp. 321–322, ACM, 2013.

[23] A. Bogomolov, B. Lepri, and F. Pianesi, "Happiness recognition from mobile phone data," pp. 790–795, 2013.

[24] G. Kumar, H. Jerbi, C. Gurrin, and M. P. O'Mahony, "Towards activity recommendation from lifelogs," in *Proceedings of the 16th International Conference on Information Integration and Web-based Applications Services*, pp. 87–96, ACM, 2014.

[25] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Aaai*, vol. 5, pp. 1541–1546, 2005.

[26] F. Xiao, Y. Chen, M. Yuchi, M. Ding, and J. Jo, "Heart rate prediction model based on physical activities using evolutionary neural network," in *2010 Fourth International Conference on Genetic and Evolutionary Computing*, pp. 198–201, IEEE, 2010.

[27] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," *Pattern recognition*, vol. 59, pp. 199–212, 2016.

[28] X. Liu, L. Liu, S. J. Simske, and J. Liu, "Human daily activity recognition for healthcare using wearable and visual sensing data," in *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 24–31, IEEE, 2016.

[29] K. Belli, "Activity prediction from auto-captured lifelog images," Master's thesis, METU, 2019.

[30] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, R. Gupta, R. Albatal, D. Nguyen, and D. Tien, "Overview of ntcir-13 lifelog-2 task," 2017.

[31] D. W. Johnston and P. Anastasiades, "The relationship between heart rate and mood in real life," *Journal of Psychosomatic Research*, vol. 34, no. 1, pp. 21–27, 1990.

[32] R. Vahey and R. Becerra, "Galvanic skin response in mood disorders: A critical review," 2015.

[33] D. M. Davydov, R. Stewart, K. Ritchie, and I. Chaudieu, "Depressed mood and blood pressure: The moderating effect of situation-specific arousal levels," *International Journal of Psychophysiology*, vol. 85, no. 2, pp. 212–223, 2012.

[34] Jović, "A review of feature selection methods with applications,"

[35] F. Jiménez, G. Sánchez, J. M. García, G. Sciavicco, and L. Miralles, "Multi-objective evolutionary feature selection for online sales forecasting," *Neurocomputing*, vol. 234, pp. 75–92, 2017.

[36] L. G. C. Coello, V. D.V., "Evolutionary algorithms for solving multi-objective problems," 2002.

[37] K. Deb, "Multi-objective optimization using evolutionary algorithms," 2001.

[38] R. C. King, E. Villeneuve, R. J. White, R. S. Sherratt, W. Holderbaum, and W. S. Harwin, "Application of data fusion techniques and technologies for wearable health monitoring," *Medical engineering & physics*, vol. 42, pp. 1–12, 2017.

[39] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3021–3028, IEEE, 2012.

[40] *XGBoost website*. `https://xgboost.readthedocs.io/en/latest/python/python_api.html`.

[41] *CatBoost website*. `https://catboost.ai/docs/concepts/python-reference_parameters-list.htmls`.

[42] *LightGBM website*. `https://lightgbm.readthedocs.io/en/latest/Parameters.html`.