TECHNISCHE UNIVERSITÄT CHEMNITZ

Faculty of Electrical Engineering and Information Technology

Department of Digital Signal Processing

Proposal for Master's Thesis

Improvement of prediction accuracy by considering outliers

Mohit Tembe

Chemnitz, 5th May 2021

Supervisor 1: Prof. Dr.-Ing. Gangolf Hirtz
Supervisor 2: Prof. Dr.Sebastian Maneth
Advisor 1: M.Sc. Tobias Scheck
Advisor 2: M.Sc. Martin Prinzler

TECHNISCHE UNIVERSITÄT CHEMNITZ

Studentenservice – Zentrales Prüfungsamt

Selbstständigkeitserklärung

Name: Vorname: geb. am: Matr.-Nr.:	<u>Bitte beachten:</u> 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: ……. Unterschrift: ……

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.

Preface

This thesis is written while visiting Arbeitsgruppe Datenbanken at the University of Bremen. The group of AG Datenbanken works in the field of Data Science. Large datasets from different domains like the eye-tracking data and the marine research data are used for making predictions by applying machine learning methods. This thesis works with the eye-tracking data in the project 'Schau mir in die Augen'.

Abstract

The study includes working with eye movements for user identification. The main objective of the thesis is to improve the accuracy by modifying a pre-existing pipeline to recognize and take advantage of outliers. For this goal, the reasons for outliers and their particularities are investigated. Various methods are used and compared for detecting the outliers. It is found that removing the outliers from the pipeline decreases the accuracy. Instead, it is studied to include them in the pipeline. For this, a new classifier is added into the pipeline along with fixations and saccades, which deal with outliers. All the experiments are performed on the Bio-Tex dataset and lead to an improvement of accuracy. The final pipeline is tried on the Bio-Ran dataset, but there are no conclusive results, and more experiments need to be performed. Two classifiers are used in the thesis, majorly is Random Forest, and the other is Radial Basis Function Network. The Bio-Tex shows an improvement in accuracy by considering the outliers for the two classifiers. For Bio-Ran, there are no conclusive results obtained. Outliers play a vital role in eye movements and show the potential to improve the accuracy in the user identification field with more research.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

The thesis is based on using eye movement Biometrics in the field of User identification. There is already pre-existing work in the field of user identification, but they come with drawbacks. Either some of the Biometrics used for user identification can be forged, or some provide less accuracy. Eye movement is a comparatively new biometric in the field of user identification. This is because of the advantages and uniqueness of eye movement as a biometric that research is in the process of developing an algorithm that can work with eye movements for user identification. An algorithm has been developed by the working group of AG Datenbanken, the University of Bremen, for using eye biometrics in user identification. Further research and development need to be done in the project to improve the prediction accuracy. This thesis includes research and work on outliers in eye movement data and improve the prediction accuracy.

1.1 Background of Eye movement Biometrics

Biometric is a measure that is generally used to identify a person uniquely based on individual traits or characteristics. Biometric data is specific and unique to each user. The primary purpose of Biometrics is to authenticate the person or ensuring that the person is who he claims to be. Thus identifying and authenticating the person is one major task, and this can find applications in various fields. The best example of authenticating the person or user identification is a mobile phone where the person is recognized depending on his fingerprint. This is the most basic example of user identification using Biometrics. Research is being carried out on other biometric features which can be used for user identification. One such feature is the eye movement of a person.

Eye movement is an indication of behavioural and physiological aspects of a person, and thus forging or replicating eye movements is difficult. The other popular biometrics like fingerprint or iris recognition are based on physiological aspects only. Therefore these characteristics can be replicated or forged and thus are not completely safe to use. Eye movements are mostly based on behavioural features, and thus it is very difficult to forge such movements. Eye-tracking research focuses on gaze point data for understanding user attention. Every individual has different scan paths or eye movements for the same view. This scan path consists of fixations and saccades. Fixations are movements of the eye during which eyes are relatively stable,

while saccades are rapid movements of the eye. The duration, position, number, order of these fixations and saccades vary from user to user.

Eye movements are an indication of the relation between the eyes and mind of a person. That is, when a person is looking at some object or an image, the location of the gaze point on that image is an indication of what a person is thinking about. It is proved that whatever a person is seeing is decided by the brain as nerves from the eyes are connected with brain muscles [1]. However, in tasks like reading, visual attention may lead to or lag.

There are various applications of eye movement Biometrics, but the main applications include identification and verification [2]. In identification, an input eye movement pattern is compared with all the eye movements in the dataset, and then the best comparison is searched for. In verification, the eye movement pattern is compared and checked whether it is of the person who he claims to be.

1.2 Background of the thesis

The thesis is based on pre-existing work presented by the authors of [3]. The project titled 'Schau mir in die Augen', also referred to as SMIDA, was developed for the purpose of user identification using a machine learning-based pipeline. In SMIDA, initially, a trajectory of eye movements is considered for a set of users. Eye movement trajectory is the point-wise movement of eyes while seeing an object/picture/video or something similar. This trajectory is divided into fixations and saccades movement, where fixations indicate points at which the eye movement is stable for a prolonged period while saccades are rapid movements of eyes. There are various other eye movements involved, but fixation and saccades are proven to be most informative and easy to distinguish and are thus used in the experiments.

From these two types of eye movements, features are extracted and then trained separately on two different classifiers. Hundreds of features can be evaluated from the data like the duration, mean values, standard deviations, and a lot more, but only a few are selected and worked on, which can be seen in detail in later chapters. Each of the classifiers predicts a user at its output for a trajectory input. The output of both of these classifiers is considered, and the final output is the average prediction of both of these classifiers. An improvement to this pre-existing work can be made by considering the outliers. Outliers are basically points that are away from the normal distribution of data.

1.3 Objective of the thesis

The aim of the thesis is to work on the outlier points and try to improve the accuracy achieved till now. The different objectives included in this thesis are mentioned below:

- To do research on outliers in eye movement trajectories. Study various reasons involved due to which the outliers are present in trajectories.
- To detect outliers from eye movement trajectories. These outliers could be present in the trajectories at different places and can be of different lengths. To detect these outlier points correctly and not to include fixation and saccade points into the outliers is the major task of the thesis.
- To remove outliers from the data and check the accuracy of the original pipeline.
- To add a new classifier in the pipeline to work with outliers. This classifier should be able to work with random data size as few users can have a lot of outliers, and some can have no outliers. So adding a classifier and selecting its parameters correctly is again a major task.
- To compare the results at different stages as mentioned below:
    1. Original accuracy
    2. Accuracies after removing outlier points from fixation and saccade
    3. Final accuracy that is accuracies after adding new classifier for outliers

This comparison is an indication of the behaviour of outliers on the performance of the whole pipeline. So analyzing these accuracies and also studying the reason for outliers in data is beneficial. The evaluation of the result is based on the mean and standard error of the mean of accuracies for a fixed number of seeds. The standard error of the mean tells how precise is the standard mean compared to the population mean.

Evaluating the pipeline and analyzing the results for different conditions is a major task in the thesis. More details about the implementation of each step, the possible modifications to the original pipeline are mentioned in the thesis.

1.4 Thesis organization

The thesis is divided into a total of seven chapters. A short overview of each chapter is mentioned below.

1. The Chapter 1 is the introduction that introduces the problem which is solved or rather dealt with in this project.
2. Chapter 2 is the background of the thesis where eye movements, work in the field of user identification based on eye movements is discussed, and also the outlier detection approaches.

3. Chapter 3 is SMIDA which is an introduction to the already existing pipeline and an overview of the methods used in the pipeline.

4. The Chapter 4 is an introduction to the outliers and the methods used in the pipeline for outlier detection.

5. Chapter 5 shows the new proposed pipeline to deal with outliers and explains modifications in each block of the pipeline.

6. The Chapter 6 presents the results for different methods, different pipelines, and different classifiers. Reasons for the achieved results are discussed here.

7. The Chapter 7 is the conclusion to the thesis, possible future work to further improve the accuracy and limitations of the work.

2 Thesis Background survey

In this chapter, initially, the basic types of biometrics are discussed and why it was needed to do research on a new type of biometric. Then the pros of eye movement biometrics are discussed along with different types of eye movements present. In the next step, the current research in eye movement biometrics for user identification is discussed. The thesis aims to work with the outliers, and thus, different approaches to deal with outliers are discussed along with their pros and cons.

2.1 Biometrics

"The term biometrics refers to a measurable, physical characteristic or personal behavioural trait used to recognize the identity of a person" [4]. Authentication is done in different ways. For example, by using some information particular to an individual like passwords or by some object, the person has like keys, or by something the person is like the person's fingerprints, voice. The following requirements [5] should be satisfied to consider a human feature as biometric:

1. Any human feature can be considered as biometrics if it can be unique from person to person.
2. The other requirement is that biometric should be recordable.
3. The third requirement is that these biometric features should be constant in a person for an extended period.

Biometrics has several advantages [5] over the traditional system of passwords or keys. Biometrics are very hard to fake or steal, like passwords. Biometrics is very easy and convenient to use. Biometrics cannot be lost by a person like the keys.

But there are a few disadvantages [5] of using biometrics which needs to be dealt with. It is not easy to set up the biometric system and make it run. If biometric data is not recorded correctly, it can lead to false recognition. Also, biometric data can be hacked and manipulated. It could create an issue if the biometric feature used by the person is affected. For example, if the finger is injured, it is impossible to use the fingerprint biometric system, which can create problems.

Many different types of biometric features can be used [6]. The selection of the biometric feature entirely depends on the application. Biometrics can be divided into two subgroups: the physiological and the behavioural groups [7]. This can be seen in

Figure 2.1. Behaviour biometrics are comparatively less expensive, while physiological is more accurate. Physiological biometrics include fingerprint recognition, face recognition, DNA, Palmprint, Hand Geometry, and Iris recognition.

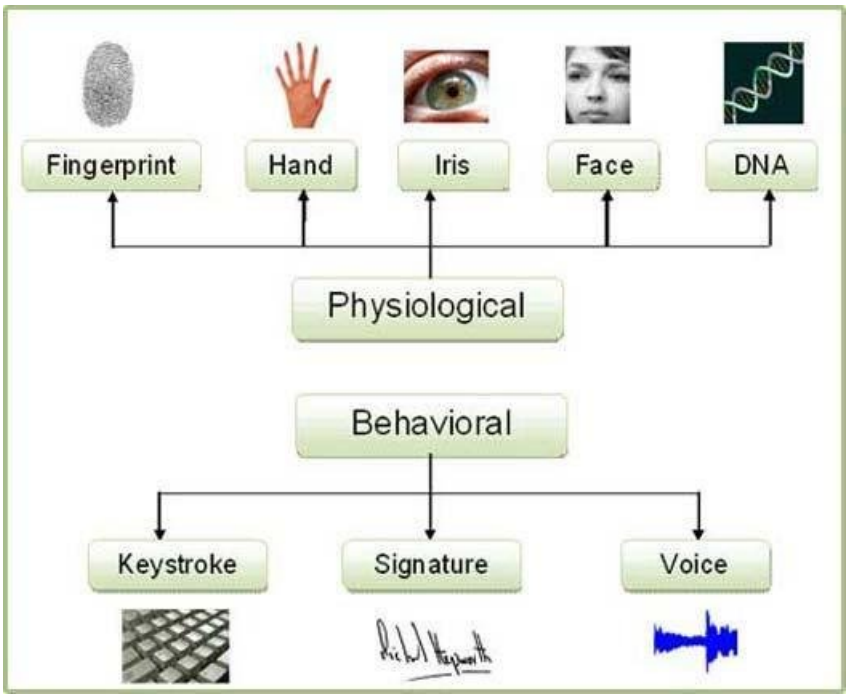


Figure 2.1: Examples of Physiological and Behavioral biometrics [8]

1. Fingerprint recognition: It is one of the most popular methods present. The epidermis and dermis are two layers of the human skin. The dermis again has two layers papillary and reticulated layer. The papillary layer is unique from person to person, and two humans can not have the same papillary pattern [9].

2. Face recognition: There are many different characteristics based on which human faces can be distinguished. There is a total of 80 nodal points on the human face. Some of them are the distance between eyes, the nose's width, the depth of the eye sockets, and the length of the jawline [6].

3. DNA: It is a part of the cell that contains genetic information and is unique for each person [6].

4. Iris recognition: Iris is unique for each person, and it does not change in the complete lifetime [6].

5. Hand recognition: For hand recognition, palm print can be considered, which is different for each person. There are different principle lines and intersection points on the palm which are used to differentiate each palm [6].

Behavioural biometrics include the keystrokes rhythm, voice, or signature.

1. Keystrokes: The method of using the keyboard differs from person to person. For example, some have fast typing speed, finger placement, pressure while typing, etc.

2. Voice: People can be recognized using their voice. The form, tone, texture of voice is different from person to person and depend on various factors.

3. Signature: Signature reveals the writing style, the pressure applied while writing, and many such factors unique for each person.

Applications [9] of biometrics are nowadays increasing in many sectors due to increased demand for security. The field of biometrics application includes a criminal department where a criminal can be detected based on his biometrics like face or fingerprint. At airports, border entry recognition of a person based on his biometrics is done. These days passports also have biometrics in them. While making government documents, biometrics are recorded and saved. These biometrics play an important role while identifying the user. These biometrics are also be used in financial applications where biometrics are needed to have access to financial statements. The most popular and most used application of biometrics is in mobile phones. Where fingerprint, face recognition can be used to unlock the device. Nowadays, all phones have an in-built fingerprint sensor. With the increasing use of biometrics, more and more research needs to be done to increase identification accuracy. Still, there are many drawbacks with the biometrics mentioned above.

The fingerprint method can lead to failure if the papillary layer is damaged. In the same way, face recognition needs proper lighting conditions for capturing data. Also, change in hairstyle, use of spectacles can lead to problems. DNA is a comparatively expensive method, and the results of the DNA test need some time to come. Hand recognition also needs a costly set up to scan the hand under proper lighting conditions. For all the behavioural types of biometrics, accuracy is very low, and they can be used in significantly fewer amount of applications [9]. So, seeing the disadvantages of the biometrics used, more research is going on to develop some new biometric feature that can give high accuracy and is of low cost. One such biometric is the eye movements of a person.

2.2 Eye biometrics

In this section discussion about the working of an eye is present. Along with this, different eye movements are discussed.

2.2.1 Oculomotor system

The coordination between eye movements is managed by the oculomotor system of the human eye. The oculomotor has six muscles that support the horizontal and vertical eye movements [10]. The two muscles, namely the medial rectus and lateral rectus, are responsible for horizontal eye movements. Contraction of the medial rectus leads to the movement of the eye towards the nose, while contraction of the lateral leads to the eye's movement away from the nose. The superior rectus is used for eye elevation and medial rotation, while the superior oblique produces eye depression and medial rotation. The inferior rectus is for eye elevation, while the inferior oblique produces eye depression. Both are for lateral rotation of the eye. All of these can be seen in the Figure 2.2.

With the help of this oculomotor system, the image seen by the eye is projected onto the retina of the eye. The retina has light-sensitive cells which convert the light into signals which are then given to the brain [11]. The light-sensitive cells are not evenly distributed with maximum density at the centre. This area of maximum density at the centre is called the fovea. A detailed vision for objects that fall on this centre part or the fovea and vision decreases outside this region. Therefore, for other objects which do not fall on the fovea, the eye movement needs to be adjusted. Using this principle, a gaze trajectory can be defined.

2.2.2 Different eye movements

A gaze trajectory is the representation of the eye movement while seeing a particular object of interest. A gaze trajectory consists of the number of fixations and the number of saccades. A fixation is a point at which eye movement is stable for a particular period. During this time, the object on the fovea is analyzed by the brain. The standard duration for fixation can be between 200 ms to 300 ms [11]. Then there is a rapid eye movement during which the eye moves from one fixation to another fixation. This rapid eye movement is termed a saccade. A saccade is part of eye movement, which is between two fixations. These saccadic movements are the fastest movement in the human body with peak angular velocities of 900 ° per second [12].

Research in this field proves that eye movement biometrics can provide high accuracies. Research is also going on in integrating eye movement along with iris recognition system [13]. There are different eye movements along with the fixation and saccades. During fixation, like already mentioned eye is not completely still, but

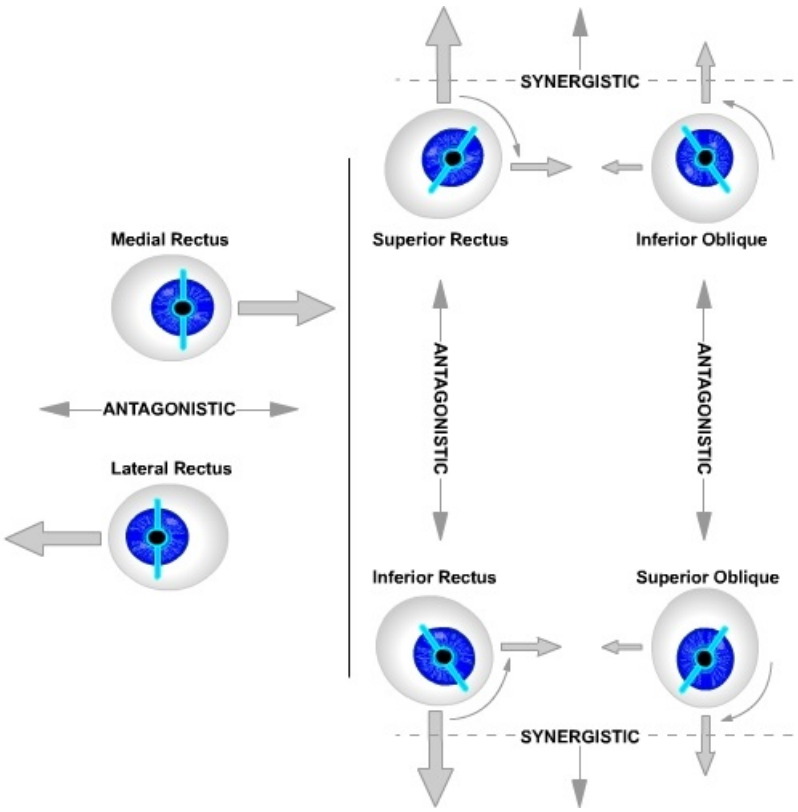


Figure 2.2: Oculomotor system of human eye showing 6 muscles that support horizontal and vertical eye movements[10]

some movements are present. These movements are called tremors, drifts, and microsaccades [14]. Tremors are movements due to muscles; drifts are movements taking eye movement away from the centre of fixation, while microsaccades are movements bringing the eye back to the centre and compensating the drift. During the saccade, there is a rapid eye movement from one point to other. The eye movement does not stabilize exactly at the destination point but wobbles a little before stabilizing at the destination point. These movements are called glissade or post-saccadic oscillations. Smooth pursuit is another type of eye movement that is generated when eyes follow an object like a bird. These are generally slow movements and are different from saccades [14]. Typical duration for each of these eye movements can be seen in Table 2.1. The duration of smooth pursuit is not given because it can differ depending on object a person is seeing. Similary tremors depend on brainstem, therefore it may vary from person to person and the duration varies depending on individuals.

Type	Duration (ms)
Fixation	200-300
Saccade	30-80
Glissade	10-40
Smooth pursuit	-
Microsaccade	10-30
Tremor	-
Drift	200-1000

Table 2.1: Duration of various eye movements. Smooth pursuit duration is not fixed and vary depending on objects. While tremor duration differ from person to person [14]

2.3 Previous work on user identification using eye movements

The most initial research in the field of eye movements was done by [15] who developed the separation of eye movement trajectory into the fixations and saccades. After further research in the field, a relation between eye movements and the way in which the brain processes things was established. It is very important to study where and when does eye movements take place. Where indicates the ending position of saccades or where exactly fixations are placed. To study this concept [16] studied how the next location is selected from the previous location.

Even though more and more research was carried on in the field of eye movements, no one saw it as an application for identification until the study by [17, 18] proved that individually developed similar scanpaths when exposed to similar environments. In a study of 1000 participants, it was observed that individual characteristics of eye movements are highly unique and persist over across various experimental sessions [19]. Further research on the amount of data required to get unique eye movement trajectories proves that approximately after reading ten lines of text, unique data is generated [20].

Even research to use other micro-movements like the microsaccade or the tremor is being carried out [21]. Even though microsaccades exist for all users but the amount of drift and tremor which differentiates individuals is not clear [22], and thus more study is required before using these movements for identification purpose. Detection of these micro-movements is also not reliable [22], and the way these micro-movements can be varied from each other is arbitrary [23, 24].

Few of the major initial research in the field of identification are mentioned in Table 2.2. The table gives an overview of the used features and the used stimuli in

each of the methods. A competition was held in 2015 to develop an algorithm based on eye movements for identification purposes. It has been described in paper [25], the different pipelines that were used by the participants of the competition. One participant directly used the trajectory for classification, while six participants divided the trajectory into fixation and saccades. The ones who divided the trajectory into fixation and saccade proved to get a higher accuracy compared to the others. Various methods have been used for classification as well, like the neural networks, SVM, weighted scores, KNN, but Neural networks proved to give the best result. The pipeline used by [12] proved to be the best and achieved very high results, and thus this method was used as a base pipeline in SMIDA.

2.4 Study on Outliers

Detecting outliers is a critical problem and needs to be dealt with. Outliers are basically abnormal points present in the dataset. One major thing to remember is that outliers and noise are not the same concepts. While noise is data that is generated due to errors, but outliers are data that can be generated due to errors as well as natural variations [42]. It can be seen from Figure 2.3 that noise are points near to the data and are disturbing the data, while outliers are some surprising points away from the data. These surprising data could be meaningful and important for

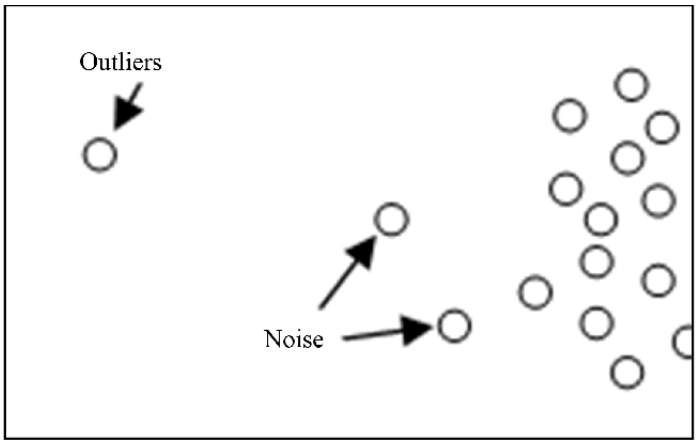


Figure 2.3: Difference between outliers and noise [43]

the study of the application.

Applications of outliers can be found in various fields as mentioned below:

1. Fraud detection: By analyzing the purchasing behaviour, abnormal use in the card can be detected.

Authors	Used features	Visual stimuli
Kasprowski and Ober, 2004[11]	Cepstrum transform	jumping-point-of-light
Bednarik et al., 2005[26]	FFT/PCA on pupil diameter/variation, eye distance , gaze velocity	Static cross
Silver and Biggs, 2006[4]	number of fixations, fixation duration, saccadic velocity/duration, gaze position	Reading while typing
Kinnunen et al., 2010[27]	Gaussian mixtuure models on histograms of velocity directions in short-term windows	Text, video
Komogortsev et al., 2010, 2012 [28][29]	characteristics of the Oculomotor Plant Model	jumping-point-of-light
Holland and Komogorstev, 2011, 2013 [30][31]	fixation number/duration, saccadic amplitude/velocity, scanpaths	jumping-point-of-light, Rorschach images, cognitive dot patterns,text
Cuong et al., 2012[32]	Mel-frequency Cepstral Coefficients on eye: position, difference, velocity	jumping-point-of-light
Rigas et al., 2012[33][34]	Wald-Wolfowitz runs test on graph-based representations of fixation points	Face images,jumping-point-of-light
Liang et al., 2012[35]	acceleration, geometric, and muscle properties of eye movements	Video clips of a moving white ball
Zhang and Yuhola, 2012[36]	saccadic amplitude, accuracy, latency, velocity and acceleration	jumping-point-of-light
Holland and Komogortsev, 2013[31]	fixation duration/position/velocity, saccadic amplitude/duration	Text
Komogortsev and Holland, 2013[37]	saccadic dysmetria, compound saccades, dynamic overshoot, express saccades	jumping-point-of-light
Rigas and Komogortsev, 2013[38]	probabilistic maps of attention- fixation density maps	Video
Yoon et al., 2014[39]	Hidden Markov models on gaze velocity	Cognitive-dot stimuli
Cantoni et al., 2015[40]	Frobenious norm of graph representation of fixation points using density/duration weights	Faces images
Rigas et al., 2015 [41]	Multi-source weighted fusion scheme using different methods	jumping-point-of-light, text, video

Table 2.2: Prior art in eye movement biometrics [25]

2. Medicine: A sudden change in the value of some results or some unusual symptoms may indicate health problems.

3. Sports statistics: Various parameters of players are tracked to evaluate the performance of the player. So players with values that differ from others could be extraordinary players.

4. Measurement errors: While taking the reading from sensors or some instrument, a sudden change in value could indicate measurement error.

5. Network performance: Detecting outliers in the network traffic could help to detect attacks on servers.

Outliers arise because of various reasons like errors in the measurement device, calibration errors, natural deviations in data, human error, or some application-specific errors. It is very important to detect these outliers and to deal with them. Sometimes it is beneficial to eliminate the outliers, or sometimes it is needed to consider them with the original data. Incorrect handling of these outliers in machine learning may lead to a longer training process of data, less accurate models, and eventually degrading results [44]. In some techniques like [45, 46], it is preferred to visualize the data initially to decide the impact of outliers. In some techniques [47], a univariate technique is used to search for data that contain extreme values on just a single variable. The rest use the multivariate techniques to search for outliers. Once these outliers are detected, an important approach to follow is how to deal with these outliers. In some cases, outliers may be needed to be included as part of data [48]. In some cases, outliers need to be dealt with depending on their dimension and application [49]. There are various approaches to detect outliers, and a few of them are described below:

1. Global versus local approaches: In the global approach, all the data points are considered, and outliers are detected by taking into consideration all the points. In the local approach, outliers are searching for from a small subset of data and not the whole dataset [50]. In the Figure 2.4, it can be seen that each cluster group has its own outlier called a local outlier, and when the overall image is taken into consideration, global outliers are obtained.

2. Labeling vs scoring: In labelling, the output is binary, indicating whether a point is an outlier or not an outlier. While in scoring, an outlier score is calculated for each point. An outlier score is a score indicating chances of a point being an outlier, and the score is returned to the user [50]. That is, the probability is returned, and not the decision if a point is an outlier.

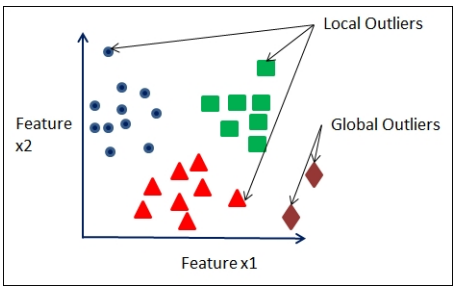


Figure 2.4: Difference between Local and global outliers [50]

3. Modeling approach: In the model-based approach, a model is designed to represent normal working; points that don't fit in this model are outliers. Proximity-based approaches examine the spatial proximity of each point to detect outliers, and if the proximity of a point deviates from the proximity of others, the object is an outlier. In the angle-based approach, the angle is calculated for a point with all other points, and if there are high fluctuations, then the point is an outlier [50].

2.5 Most common methods to detect outliers

There are various methods to detect outliers. A few of the most common methods are mentioned and discussed in this section.

2.5.1 Density based methods

The basic assumption of density-based methods is that outliers are found in low, dense regions while non-outlier points are in high-density regions. The density of a point is compared with the density of nearby points, and then a decision is made whether a point is an outlier or not. There are many famous methods based on density-based approaches. One of the most famous or the most used approaches is the Local Outlier Factor(LOF) [51]. In this method, outliers are computed using the K-nearest-neighbor. One another approach is the Connective based Outlier Factor(COF) [52]. It is based on LOF, but the only difference is that here chaining distance is used to calculate the shortest path and detect outliers. One of the approaches mentioned by [53] named as Local outlier probabilities provides the outlier score with a probabilistic and statistical oriented approach.

- Advantages: This method efficiently removes points near dense cluster areas, and only a minimum of prior knowledge is required to implement this method.

- Disadvantages: For varying density regions, it is quite a complicated process to get correct outliers. The runtime of these algorithms is very high due to the high computational complexity

2.5.2 Statistical based methods

In the statistical methods, all the points are modelled using stochastic distributions, and outliers are detected depending on their relationship with the distribution model. There are two types of statistical-based approaches; the first one is the parametric approach, where a comparison of the data points is made with some distribution model, and the second is the non-parametric approach, and in this approach, there is no model to be compared with. In the method introduced by [54], a maximum likelihood estimate is done to get the mean and variance of the gaussian distribution in the training stage, and in the test stage, mean-variance or box-plot tests are done. In another method, a regression model is developed during the training stage that fits the data, and in the testing stage, the data points are compared with this model. At points where there is a high deviation, they are labelled as outliers as mentioned by [55]. In the non-parametric method, a kernel density estimation is done as mentioned by [56]. It is a type of unsupervised approach. Some of the most common methods are histogram, Boxplot, Trimmed mean, etc.

- Advantages: The evaluation time is very less. They have improved performance given to the previous method because of the probabilistic approach.

- Disadvantages: The results received are unreliable due to a lack of preceding knowledge regarding the underlying distribution. They don't perform effectively in multivariate scenarios.

2.5.3 Distance based approaches

As the name suggests, in this approach, a distance is computed between the points, and then the outliers are recognized. If the point is far from the nearest point, then it is termed as an outlier. The most commonly used distance-based detection method is the K-nearest neighbor [57]. They are mostly used to detect global outliers. There are many variations based on K-nearest neighbour based on ranking the neighbours as mentioned by [58] or to make a model learn about outlier behaviour from a dataset and then find outliers for incoming dataset [59].

- Advantages: They don't depend on any model and are easy to compute. Also, they work well in multidimensional spaces.

- Disadvantages: In some algorithms, it is not possible to detect some complex outlier points. It becomes complicated in high-dimensional space.

2.5.4 Clustering based approach

In this method, small size clusters are made, and then the density of clusters is evaluated. Clusters with smaller densities are described as outliers. The formation of clusters is a major task, and there are few methods describing how the clusters can be formed.

1. Partitioning Clustering methods: They are also known as distance-based clustering algorithms. Examples of this method are mentioned by [60, 61, 62].

2. Hierarchical Clustering methods: In this method, the points are classified into groups of different levels, and a tree-like structure is formed. For this, usually, the maximum number of clusters is mentioned, and clustering or splitting is done until the desired number of clusters are achieved [63, 64].

3. Density-based Clustering methods: In this type of method, usually the radius for clusters is given, and then clustering takes place [65, 66].

- Advantages: As they are an unsupervised method, they are highly popular and used. They are robust to different data types

- Disadvantages: There is no quantitive indication of an outlier. Specifying the number of clusters or the radius is a difficult task.

There are few other methods like the ensemble-based approach or learning-based approach, but they are not much studied and used. They are complicated compared to the previous algorithms and are usually used in machine learning [67].

3 SMIDA: Schau Mir In Die Augen

This chapter is an overview of the previous work done by the data group of AG datenbanken, University of Bremen. The project 'Schau mir in die Augen', also referred to as SMIDA, is developed to work with the eye movement data for user identification. In this chapter, an overview of the pipeline used in SMIDA is presented. Different steps done to convert the original eye movement data to the desired form and then to perform user identification using a machine learning pipeline are presented in this chapter.

The pipeline for SMIDA is mentioned in Figure 3.1. The implementation of eye movement as biometrics for user identification requires different steps to be performed. The current project is based on the work of [12]. The various steps include:

1. Selection of Visual stimuli
2. Pre-processing the data from the dataset
3. Eye movement segmentation (Segmentation into fixations and saccades)
4. Feature extraction
5. Classification

The original raw data available is usually the viewing angle of the users recorded using an eye-tracking device. Some of the datasets used in the SMIDA are explained in Section 3.1. This raw data cannot be directly fed to the classifiers for user identification. Some pre-processing steps need to be performed to convert the data in the correct form that can be feed to the classifier. These pre-processing steps are explained in Section 3.2 After performing pre-processing, now from the trajectory available, similar parts are grouped together. Grouping these similar parts makes it easier to perform a machine learning algorithm on them. Grouping these similar parts known as fixation and saccades is yet another important task, and details about the same can be seen in Section 3.3. Then these fixations and saccades are feed for the feature extraction. From a set of around 300 features, few are selected. The selected features are described in Section 3.4 along with the different methodologies used for feature selection. Then these features are fed separately into the classifiers. Two different classifiers are used, one for the fixation and one for the saccade. The classifiers and their parameters are mentioned in Section 3.5. All these fixation and saccades can be visualized by a visualization tool developed by the members of the

data group. Different capabilities of the tool can be studied in Section 3.6 In the end, an averaging method is used to merge the output from the two classifiers. The final result is the predicted user. The resulting accuracy can be seen in Section 3.7. These are the main steps involved in the project, and a detailed explanation of all these steps will be discussed in this chapter.

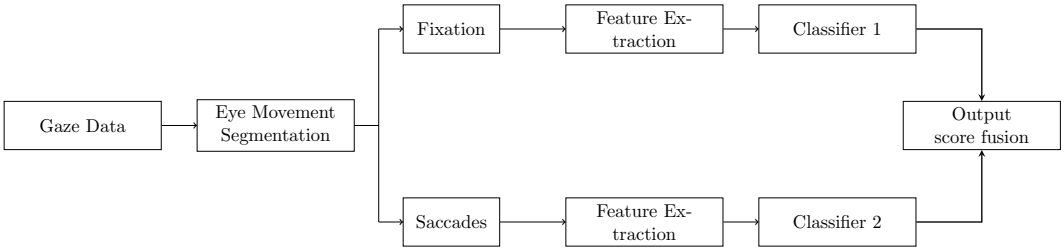


Figure 3.1: Original Pipeline

3.1 Visual stimuli

Visual stimuli are basically the images or videos shown to the participants for recording the eye movements. Therefore, the selection of visual stimuli is a very important task as it is the representation of both physiological and cognitive characteristics. The visual stimuli developed are made considering the oculomotor system of the eye. For example, in images with text, the direction of eye movement is in the direction of the text. While in images of surroundings, it reflects what a person thinks when he observes an image and thus replicates his cognitive behaviour. It has been observed that when training is done on one particular stimulus and testing is performed on different stimuli, a very low level of accuracy is obtained. Such tasks are known as strongly task-independent classification. While when training and testing are done on similar but different images, then such tasks are called weakly task-independent classification. Examples of the different visual stimuli can be seen in Figure 3.2

Different datasets indicating the visual stimuli used in the project are given below:

1. Bio-Tex and Bio-Ran
2. Where Humans Look
3. Visual search task

Among these experiments are performed only on Bio-Tex and Bio-Ran dataset, so both of them are described below.

Bio-Tex and Bio-Ran dataset

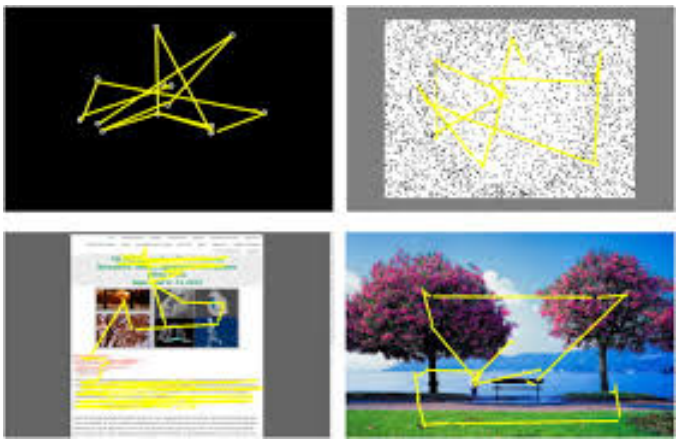


Figure 3.2: Examples of different Visual stimuli [25]

The Bio-Tex dataset is based on text excerpts, while Bio-Ran is based on random 'jumping' points of light. In the Bio-Ran dataset, random points of light are shown to the user each after a specific time interval. The example of Bio-Ran can be seen in Figure 3.3. The text scripts contain a poem as shown in Figure 3.4. Both of these datasets were used in bio-eye competition 2015 [25].

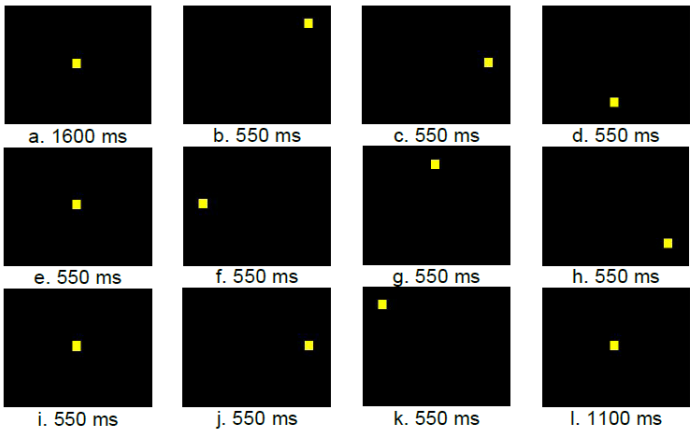
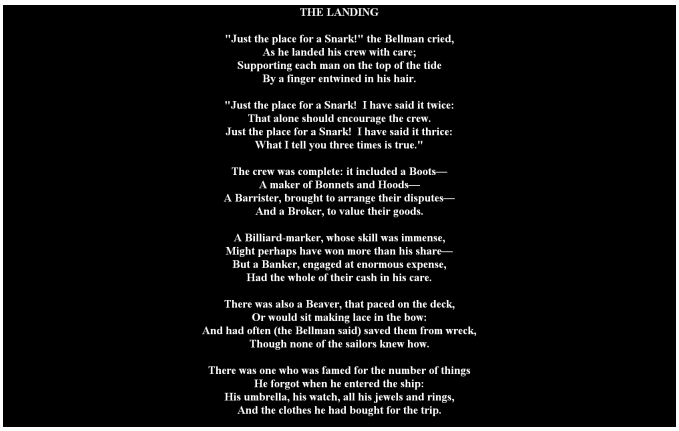
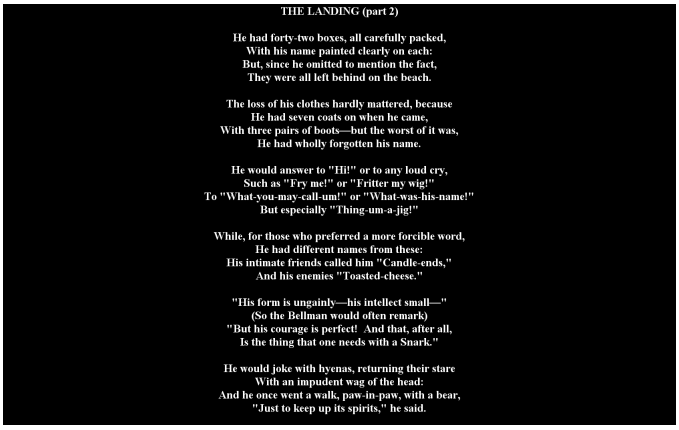


Figure 3.3: Example of Bio-Ran dataset. The point of light is visible at a different location after a set time interval [68].



(a) Session_1



(b) Session_2

Figure 3.4: Bio-Tex_Stimulus. Up: Stimuli used for training, Down: Stimuli used for testing

The Bio-Ran dataset generates random points of light on the screen, thus giving a brief overview of horizontal and vertical saccadic movements of the eye with varying velocities. On the other hand, the Bio-Tex data set also gives horizontal and vertical saccadic movements on eyes but in a guided manner as the reading direction is from left to right and then onto the next line. The saccadic movements along with velocities can be seen from Figure 3.5. It can be seen that there is random movement in the horizontal and vertical directions. That is, from point 1 to point 2, the direction is from left to right and top to bottom, while for points 2 to 3, it is from right to left and bottom to top. On the other hand, for Bio-Tex, the main movement for all the points is from left to right. The vertical change is small compared to the horizontal until the user moves to the next line.

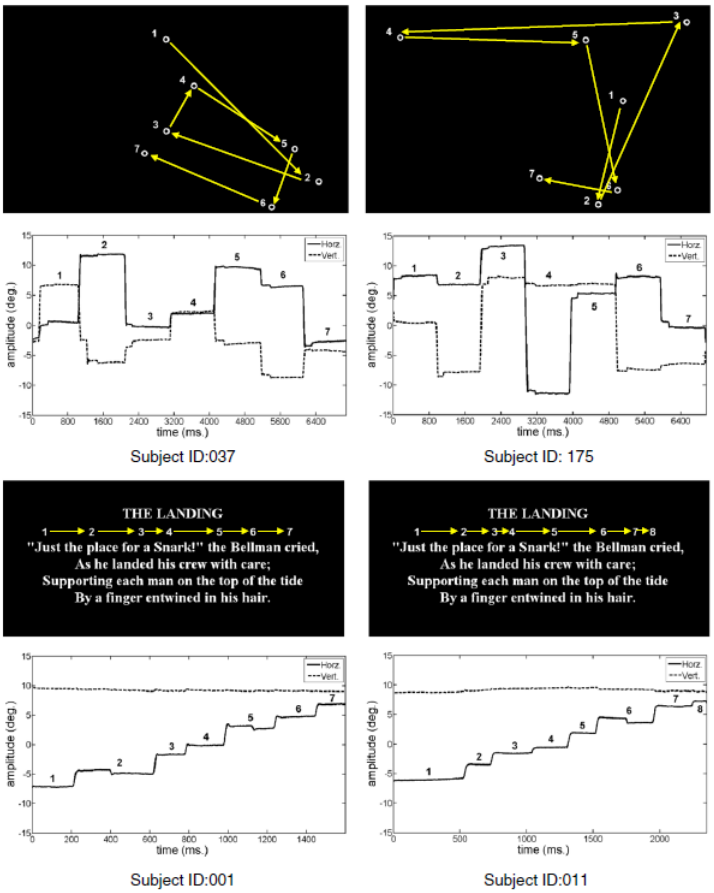


Figure 3.5: Velocity variations for both datasets. The velocity variations are for two users. One on the right and the other on the left. The above two rows are for the Bio-Ran dataset, and below two rows are of Bio-Tex dataset [25]

Eye Tracking data can be captured in a lab-controlled environment or an open environment. Capturing in an open environment requires a lot of work to be done on the datasets to compensate for the noise in the image due to external factors like lighting conditions. A lab-based environment can be used to capture data with high-quality [69]. For Bio-Tex and Bio-Ran, a lab-based environment is used with chin rest and head-bar to avoid head movements. Eyelink 1000 [70] eye tracker is used to capture eye movements. It has different options like the tower, primate, LCD Arm, Desktop mount. Among these, the desktop mount is used as shown in Figure 3.6. It has a spatial accuracy of 0.5° typical, a sampling rate of up to 2000 Hz depending on the mount used. A spatial resolution of 0.1° for 1000 Hz and 0.2° for 2000 Hz. The tracking distance, camera eye distance, allowed head movement depends on the mount

used for the device. More detailed information about the device can be obtained from the manual [70]. Eyelink 1000's is used in various experiments like [71, 72, 73]. For the experiments, a monocular mode was used, and movements of the left eye were captured.

(a) Desktop mount angle [70]

(b) Desktop mount level [70]

Figure 3.6: Desktop mount present in the eye tracker for recording eye movements [70]

The size of the dataset plays an important role. The dataset needs enough data to be divided into train and test. The small size of the dataset could lead to overfitting. Bio-Tex and Bio-Ran both have a high number of participants and thus are a big enough dataset. To be specific, there are a total of 306 participants, among which 165 were male and 141 were female. 154 participants had corrected vision sight while 152 had an uncorrected vision. Among the corrected vision sight, 63 used glasses, and 91 used contact lenses.

All these participants are shown some images on a screen. The images are of size 1680×1050 pixels, and the screen size is $474\,\text{mm} \times 297\,\text{mm}$. The distance between the participant and the screen is 550 mm. The exact setup can be seen in Figure 3.7. The above parameters lead to a horizontal viewing field of $46.6°$ and a vertical viewing field of $30.2\,°$. For the RAN dataset, the points move within a range of $\pm\ 9\,°$ vertical and $\pm\ 15\,°$ horizontal.

The duration of the dataset depends on the amount of information needed to extract the features and the type of behaviour that needs to be studied. The RAN dataset has a duration of 1 minute and 40 seconds, with the point of light changing every 1 second. The TEX dataset has a duration of 1 minute. The recording for participants was again taken after a particular time interval. Two-time intervals were considered; a short time interval was for 19.3 and 19.6 minutes for two datasets, while the long time interval was for 10.5 months. The second session is used for testing purposes. So there are a total of 3 sessions. The first one is the initial session,

the second is the one taken after 20 minutes approximately, and the third one is the one taken after 10.5 months. For the first session, a total of 306 participants were considered; for the second session, all these 306 participants were again considered, but for the third session, only 76 participants were considered. Therefore there are four combinations present for performing the experiments:

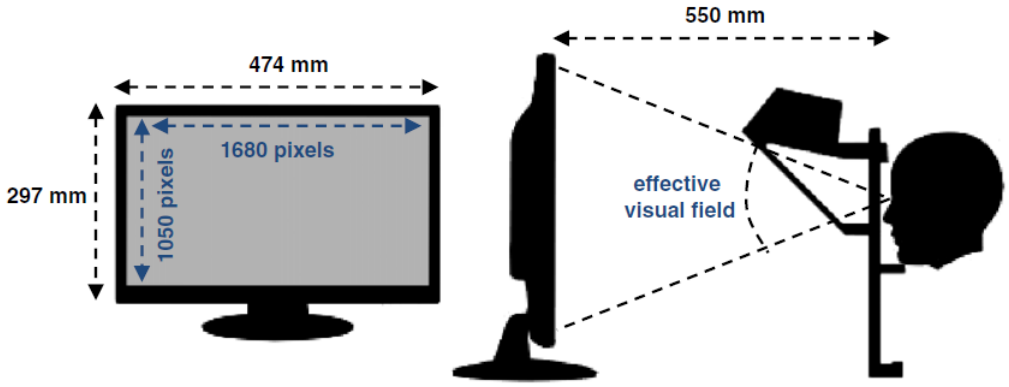


Figure 3.7: Viewing setup [25]

1. Train on Bio-Tex (session 1) and test on Bio-Tex (session 2)
2. Train on Bio-Tex (session 1) and test on Bio-Tex (session 3)
3. Train on Bio-Ran (session 1) and test on Bio-Ran (session 2)
4. Train on Bio-Ran (session 1) and test on Bio-Ran (session 3)

Two different aspects were considered for analyzing the datasets. The first one was the calibration accuracy, and the second was recording validity. For each participant, before the main recording, a calibration test is performed. For calibration, 9 points are shown on screen, and participants need to fixate on this point till the next point arrives. A calibration map is derived for each user, and then this map is used during the main recording. Calibration accuracy is important to be calculated to see the possibility of error in the gaze trajectory. Due to some problems like the device-specific problems or the user-specific problems (blinking, loss of attention), there are high chances that the recording equipment is not able to record the data or successfully capture the samples. Therefore recording validity is an indication of successfully captured samples. Users with low recording validity were removed. From the pool of 322 participants, 306 were registered.

3.2 Pre-processing of raw data

Before the data could be used further for extracting, these data need to be processed. Different pre-processing steps like filtering, interpolation are explained in this section. A flowchart of the steps can be seen in Figure 3.8

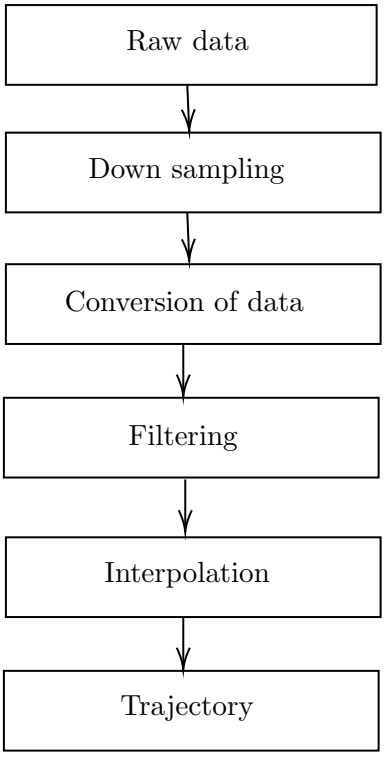


Figure 3.8: Pre-processing steps

- The input data available is recorded at a frequency of 1000 Hz. This high-frequency data is first downsampled to 250 Hz for Bio-Tex and Bio-Ran dataset.
- For the Bio-eye dataset, the data is available in terms of viewing angle. The angle at which the eye moves from the center is available in the dataset. So it is necessary to convert the angle to pixel values which can then be used to form the trajectory. Angles are measured from the center viewpoint in the horizontal and vertical directions. The formula used for this is:

$$xy = (\frac{\text{pixel}}{\mathsf{mm}}) \times (\tan(\mathsf{viewing\ angle})) \times (\mathsf{screen\ distance}) + (\frac{\mathsf{screen\ resolution}}{2}) \tag{3.1}$$

where,

1. *pixel/mm* is the screen density in terms of x and y
2. *viewing angle* is the angle representing the eye movement
3. *screen distance* is the distance between the participant and screen
4. *screen resolution* is resolution in pixels

- The raw data available may contain noise. This noise is mainly due to high-frequency components available in the dataset, and to overcome this, filtering is done. These high-frequency components will affect the velocity and acceleration profiles. A Savitzky-Golay filter [74] is used for this purpose. This is a digital filter used for smoothing the data points. Smoothing is achieved by using the convolution and linear least-squares method. They have high accuracy and have very low least square error while fitting a polynomial to frames of noisy data. The default value used for polynomial order is 6, and the frame size is 15.
- The next step is an interpolation that is adding the data points within the trajectory. This is done to join two trajectory points placed away from each other.
- After these steps, the trajectory is achieved, which can be used for detecting fixation and saccades.

3.3 Eye movement segmentation

The trajectory generated from the steps described above needs to be divided into fixations and saccades. Fixations are periods during which eyes are relatively stable on a particular point, while saccades are rapid eye movements. Saccades can be defined as movements between two fixations. Fixations have very low-velocity values, while saccades have high-velocity values. The minimum duration for identifying the fixation points is set to 0.1 seconds to avoid false identification during small positive values of velocity. Differentiating the position values will return the velocities. The most basic and most commonly used algorithm to differentiate between the fixations and saccades is the I-VT(Velocity threshold identification) algorithm [75]. I-VT algorithm uses threshold values for differentiating between fixations and saccades. Many more complex algorithms are available for differentiating between fixations and saccades, but I-VT is proven to give significant results [25]. Other algorithms for this task are:

1. Hidden Markov model identification (I-HMM) algorithm [76]
2. Kalman filter identification (I-KF) algorithm [77]

3. Dispersion-threshold identification (I-DT) algorithm [78]

4. Minimal spanning tree identification (I-MST) algorithm [79]

The first two are based on velocity characteristics, while the rest two are based on dispersion characteristics. IV-T algorithm is given in 1.

Algorithm 1: IVT algorithm [12]

```
Result: Res
Constants: VT= Velocity threshold, MDF= Minimum duration for fixation
States = [FIXATION; SACCADE] ;
fixationStart = 1;
Velocity = smoothDiff(data);
N ← Number of samples of data;
for index ← 1 to N do
    if Velocity[index] < VT then
        currentState = FIXATION
        if lastState ≠ currentState then
            fixationStart = index
        end
    else
        if lastState = FIXATION then
            duration = data(index,1)- data(fixationStart,1)
            if duration < MDF then
                for i ← fixationStart to index do
                    res[i] = SACCADE
                end
            end
        end
        currentState = SACCADE
    end
    lastState = currentState res[index]=currentState
end
Res ← res
```

The algorithm is quite simple, but it takes quite a few things into consideration, like the minimum duration of fixation and the threshold value. The algorithm has two stages, the current state and the last state indicating the previous state. This state can be either Fixation or Saccade. N is the number of samples present in the dataset. Velocity is taken by differentiation as described above. For each sample, if the velocity is less than the threshold value, then it is considered to be a fixation. For each time consecutive fixations occur, the value of the first index for fixation is saved. When there are changes from fixation to saccade, the duration of previous consecutive iterations of fixation is calculated. If this duration is less than the minimum duration

of fixation, then the index values for these fixations are replaced with saccades. If this duration is more than the minimum duration of fixation, then the fixation values are kept as it is. At the end of every iteration, the last state is replaced with the current state, and the value is appended in a list. This output list contains the starting point for each fixation and saccade and contains the duration for fixation and saccades. There are some post-saccadic oscillations that can be removed. This is done by removing saccades with a duration of fewer than 12 milliseconds. In this way, fixations and saccades are extracted from the trajectory in a very easy way.

3.4 Feature extraction

This step is the most important step in the pipeline. After the extraction of fixation and saccades, features can be extracted from both. It is possible to directly extract features from the trajectory, but usually, fixations and saccades are extracted as they give information about the specialized characteristics in the similar parts in the trajectory. Fixations and saccades have their respective amplitude, velocity, and acceleration profile. From these profiles, more features can be calculated, like the average velocity between two fixation/saccade and so on. It is generally observed that saccades have more information compared to fixations [31]. The vertical and horizontal saccadic movements correspond to the cognitive behavior [80]. The Table 3.1 indicates features extracted from fixations and saccades for Bio-Tex and Bio-Ran. A total of 51 features are used for both fixations and saccades. All the features are described below:

1. Duration: It is the time for which a single fixation or saccade lasts or is present. It is obtained from the result of I-VT.

2. Standard deviation: It is the dispersion value for the dataset relative to its mean value.

3. Path length: It is the total length of the path traveled on the screen. It is given by formula Equation 3.2

$$\text{Path length} = \sum N - 1i = 1\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{3.2}$$

4. Angle with the previous fixation: It is the angle formed by the centroid of current fixation with the previous fixation.

5. Angle with the previous saccade: It is the difference between the saccadic angle of current and previous saccade.

6. Saccadic angle: It is the angle obtained from the first and last point in a saccade. It can be given by formula Equation 3.3.

$$\text{Saccadic angle} = \tan^{-1} \frac{y_N - y_1}{x_N - x_1} \tag{3.3}$$

7. Saccadic ratio: It is given by the formula Equation 3.4.

$$\text{Saccadic ratio} = max \frac{(\text{Angular Velocity})}{(\text{Saccadic duration})} \tag{3.4}$$

8. Saccadic amplitude: It is given by the formula Equation 3.5.

$$\text{Saccadic amplitude} = \sqrt{(x_N - x_1)^2 + (y_N - y_1)^2} \tag{3.5}$$

9. Distance from previous fixation/saccade: It is the euclidean distance between current and the previous fixation or saccade respectively.

10. Skewness: Skewness is the distortion of the data points from the expected distribution (for exam bell shape) of the data points. It is the indication of whether data points are symmetrical that is equal on the left and right side of the center point.

11. Kurtosis: Kurtosis is a measure of how much amount of data is present at the tail compared to the normal distribution .

12. Dispersion: Dispersion is a measure of the statistical spread of the values from a particular point. It can be given by Equation 3.6.

$$\text{Dispersion} = (max(X) - min(X)) + (max(Y) - min(Y)) \tag{3.6}$$

13. Average velocity: It is the average distance traveled in particular time duration. It can be given by Equation 3.7.

$$\text{Average velocity} = \frac{\text{Path Length}}{\text{Duration}} \tag{3.7}$$

14. Acceleration: It is the average rate of change of velocity for a particular time duration

Different methods used in the project

Various methods are used to extract features. Originally there are approximately 300 features from which the above-mentioned features are used for the Bio-eye dataset. These 51 mentioned features are extracted using the method mentioned by [12]. All the methods available are mentioned below:

Skewness(X)	M3S2K(Angular Velocity)	Saccadic ratio
Skewness(Y)	M3S2K(Angular Acceleration)	Saccadic angle
Kurtosis(X)	Standard Deviation(X)	Saccadic amplitude
Kurtosis(Y)	Standard Deviation(Y)	M3S2K(Velocity_X_direction)
Average Velocity	Path length	M3S2K(Velocity_Y_direction)
Saccadic/ Fixation duration	Angle with previous saccadic	M3S2K(Acceleration_X_direction)
Dispersion	Distance with previous saccadic	M3S2K(Acceleration_Y_direction)
M3S2K-Statistical features: Mean, Median, Max,Std,Skewness,Kurtosis		

Table 3.1: Total 51 features used for both fixations and saccades

1. Score level Evaluation: In a total, 40 features were extracted from saccades, and 9 features were extracted from fixation for the Bio-Ran dataset. In a similar way, 43 saccadic features and 9 fixation features were extracted from the Bio-Tex dataset.

2. Our-append method: In this method, the calculation is done for all the available features that are for approximately around 300 features.

3. Paper-append method: In this method, all the 51 features considered for Bio-Tex in the table above are considered for all the datasets.

In all the experiments, the paper-append method is used; that is, all the 51 features are used.

3.5 Classifiers

Various classifiers are used in the project. The input to the classifier is, in this case, the fixation and saccade features. Based on these features, the classifier is trained to predict the correct class at the output. Therefore, the classifier plays a major role in the pipeline. All the four classifiers that are Random Forest, Radial Basis function network, Naive Bayes, Support Vector machine, along with their parameters, are mentioned in this section.

3.5.1 Random Forest

Random forest [81] is built on the principle of decision-making tress. Various decision-making trees are used to predict the final output of the classifier. A decision-making tree works on the principle of classifying the classes based on some feature. When multiple decision-making trees are taken into consideration for calculating the final result, then the algorithm is called the Random Forest classifier. Each individual tree in Random Forest makes a prediction and classifies the output as one of the

predicted users. The final output is based on voting from each of the decision trees. The user with maximum votes is the final output of the classifier. All the trees are uncorrelated and therefore perform differently and give a different set of output. For example, if there are a total of 4 features given to the input of the classifier then, all the decision trees will be trained on a different set of these features, and therefore the final output of all the decision trees will be different. Each tree based on a feature further splits and classifies itself until it reaches a conclusion where it can predict a final user. Therefore as it further splits and each tree has a different set of features based on which it works, therefore the correlation between these trees is very less. The biggest advantage of less correlation is that even if one tree is making a wrong prediction, it does not affect the result of the other. Parameters of RF are described below:

1. n_estimators: These are the number of trees in the forest. This has been set to 200 for most of the experiments.

2. max_depth: It is the max depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

3. max_features: These are the number of features to be considered while splitting. This has been set to sqrt that is sqrt of n number of features is taken.

4. min_samples_leaf: The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches.

5. min_samples_split: minimum number of samples required to split an internal node.

3.5.2 Radial Basis Function Network

An RBFN [82] consists of 3 layers defined as the input layer, the hidden layer, and the output layer. This can be seen in the Figure 3.9

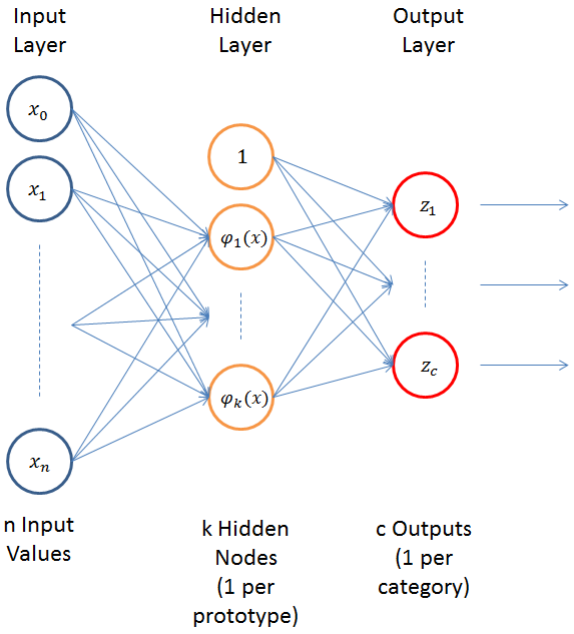


Figure 3.9: RBFN architechture showing 3 layers [83]

The input layer consists of the training data, which in this case are the feature vectors. These feature vectors are then given to the hidden layer. The hidden layer consists of a radial basis function centred on a point with the same dimensions as the predictor variables. The output layer consists of the weighted sum of the output from the hidden layer. Each new training item's target value is calculated depending on its distance from nearby items. K means clustering is used along with RBFN to calculate the centres of the neurons. Euclidean distance is calculated for each new incoming point. The gaussian activation function of each neuron is given by Equation 3.8.

$$\varphi(x) = e^{-\beta\|x-\mu\|^2} \tag{3.8}$$

A radial basis function is applied to this distance to calculate the weight for each classifier. The main parameter of RBFN is n_clusters which indicates the number of clusters as well as the centroids to be generated. The cluster size selected is 32.

3.5.3 Naive-Bayes

Naive bayes algorithm is based on the Bayes theorem [84]. Bayes thorem can be explained from the Equation 3.9,Equation 3.10.

$$P(Ck \mid X) = \frac{P(X \mid Ck)P(Ck)}{P(X)} \tag{3.9}$$

$$P(Ck \mid X) = P(X_1 \mid Ck) \times P(X_2 \mid Ck \times ..... \times P(X_n \mid Ck) \times P(Ck) \quad (3.10)$$

Here, $P(Ck \mid X)$ is the posterior probability, $P(X \mid Ck)$ is the likelihood, $P(Ck)$ is the prior probability of a class, and $P(X)$ is the prior probability of the predictor. Here $P(Ck \mid X)$ means that the probability of Ck to be true when X is true. In Naive Bayes, the assumption is that the features are independent of each other, and thus one feature is taken at a time [85]. A common decision rule is used to select the final prediction, for example, a maximum a posteriori decision rule [86]. It has the advantage that it works well even when the data size is small, and it is a very simple and fast algorithm.

3.5.4 Support Vector Machine

SVM [87] is a supervised machine learning algorithm used for classification and regression. Each data point is plotted in the n-dimensional space with its value to be the feature value. Then the classification is done based on the hyperplane. A linear hyperplane does the work of dividing the points into classes as shown in the Figure 3.10. Selecting the correct hyperplane plays a major role in SVM. Two criteria need to be considered in selecting the correct hyperplane; first are all the points getting divided correctly, and the second is to check the margin or the distance to the nearest data point to the hyperplane for each class. In cases where the linear hyperplane doesn't work, a kernel function is used. The incoming data is then converted using the kernel function to higher-level data as shown in Figure 3.11b and then classification is done as shown in Figure 3.11c. The complete functionality can be seen in Figure 3.11

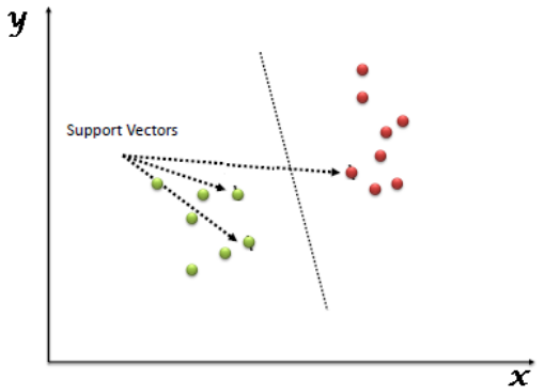
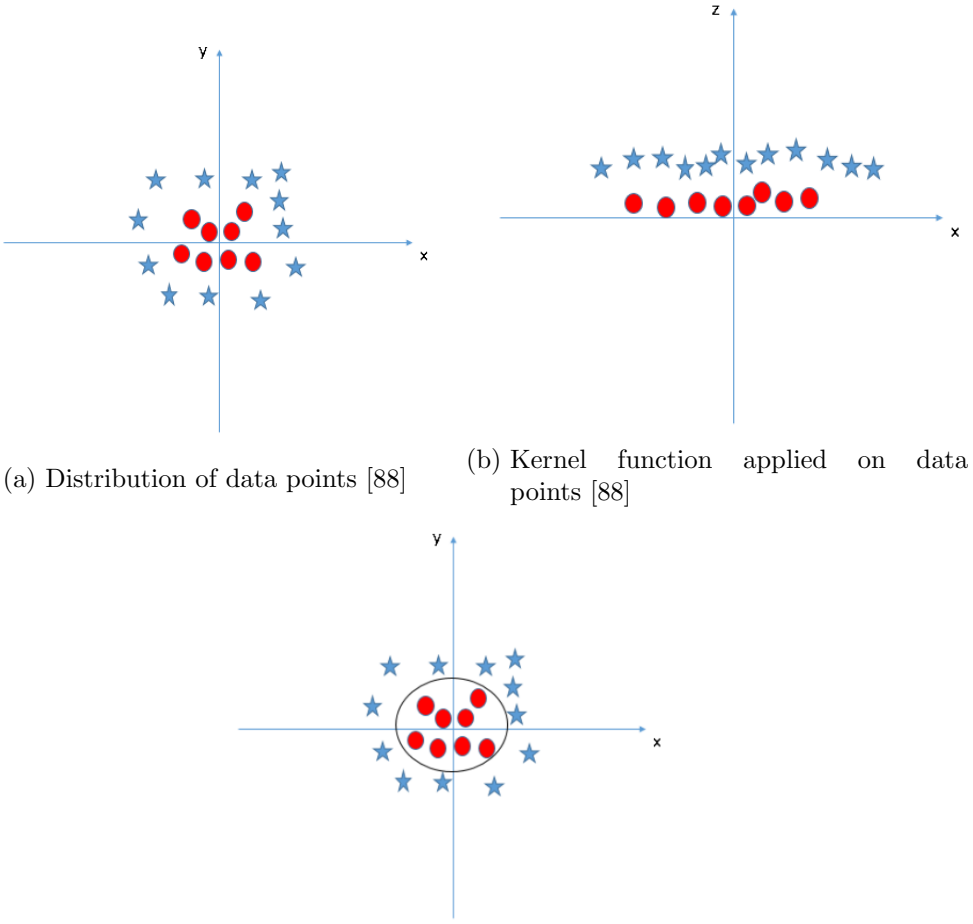


Figure 3.10: Support vectors and hyper plane [88]



(a) Distribution of data points [88]

(b) Kernel function applied on data points [88]

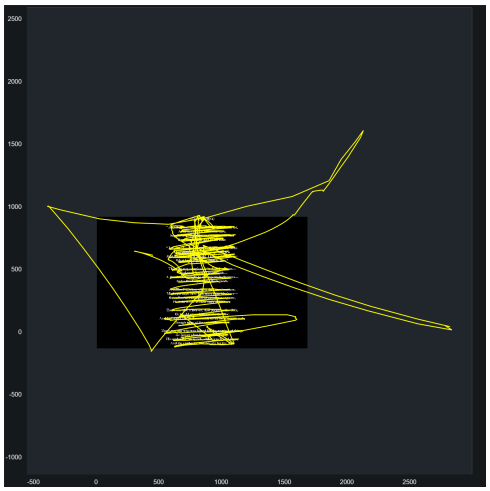
(c) Hyperplane for SVM [88]

Figure 3.11: Working of Support Vector Machine [88]

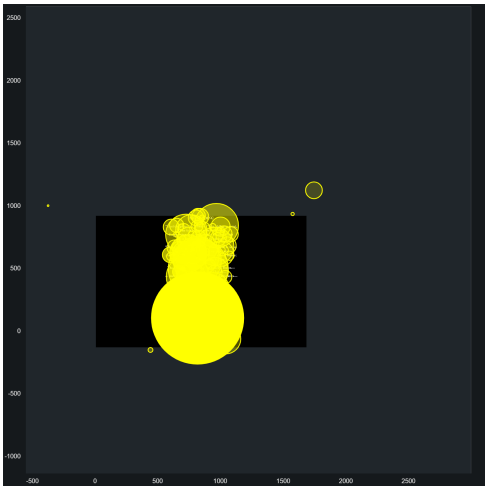
3.6 Visualization

A visualization tool was developed by the data group of AG Datenbanken. The purpose of creating the visualization tool was to be able to view the complete trajectory on the image area as shown in Figure 3.12a. Along with this, fixation and saccade position can be viewed on the image frame Figure 3.12b, Figure 3.12c. Along with the fixation, its count can also be seen in the visualization. This enables to understand what objects exactly the user sees on the image area. Various options are available in this visualization tool. A button is available for viewing the trajectory, viewing fixation/saccade, or cropping the trajectory. Various other options like selecting the

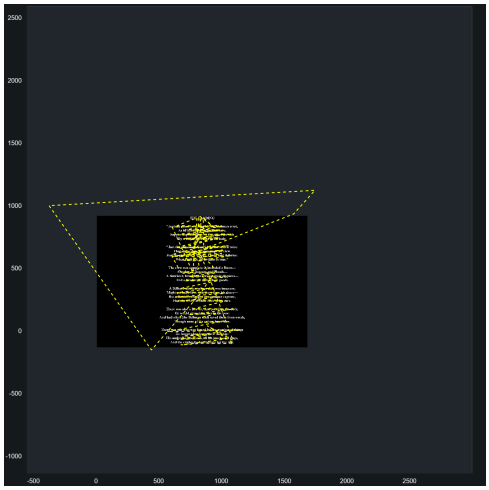
dataset, selecting a particular session, a particular participant are available. Such options make it very easy to change the dataset or the user and easy to use for any one. This makes the visualization a very strong tool.



(a) Complete Trajectory



(b) Only fixations



(c) Only saccades

Figure 3.12: Different options to view trajectory in the visualization tool

3.7 Results

This section explains the previous results achieved in the SMIDA yet for the Bio-Tex and Bio-Ran datasets. The results presented are for 100 users for both datasets. The

	Bio-Tex	**Bio-Ran**
RF	91.8 0.58	94.4 ± 0.4
RBFN	92.4 ± 0.49	95.6 ± 0.76

Table 3.2: Original accuracy

results are excellent and prove that the use of eye movements for user identification is a promising field of research. More detailed research and work in this field can lead to better results. This thesis aims to increase the prediction accuracy for user identification. Table 3.2 describes the results achieved using SMIDA. The results for user identification can be seen for RF and RBFN classifiers as these two classifiers perform best among all. Equal weights have been given to both fixation and saccade classifiers as both of these features are of equal importance.

Results for cross-evaluation have also been calculated that is training on Bio-Tex and testing on Bio-Ran and vice-versa. But in both cases, the results are not significant. Therefore a user trained on a particular stimulus may have a different approach to viewing objects on different stimuli. More research needs to be done here.

4 Outlier Detection

In this chapter, a discussion about what are outliers, why they are present in eye movement data, and how they can be dealt with is presented. Various methods to overcome outliers and the procedure implemented in this thesis are discussed.

4.1 Outliers

There are many definitions for outliers. One states outliers as data points that are far away from the mean or from clusters of data. Another definition is as given by Hawkins [89], which states *"outlier as an observation that deviates so much from other observation as to arouse suspicion that it was generated by a different mechanism"*. To explain it with an example, in Figure 4.1 most of the data follow a linear line of increase and are close to each other. But there is a single point that is far away from this set of data. This data point can be termed as an outlier. Outliers can be added to the data because of various reasons like errors in measurement, problems in sampling the data, or similar reasons [90]. But these outliers can affect your algorithm as there can be a change in the mean value or the standard deviation of data. They can also affect the classification accuracy and are therefore not a desirable input to the algorithm.

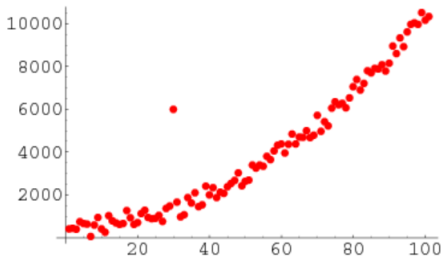


Figure 4.1: A outlier point visible in data away from all other points [90]

4.2 Anamolies in eye movement trajectories

Anomalies in eye movement trajectory exist because of various reasons like calibration errors, faults in the eye-tracking device, blinking, and so on. The reasons for outliers can be divided into two sections, the first one can be the reasons due to machine failure. The second one can be reasons due to natural causes. A few of the reasons will be discussed in this section in detail and possible approaches to counter this type of problem.

4.2.1 Outliers due to faults in device

These outliers are due to problems with the eye-tracking device. Sometimes the eye tracking device is not able to properly detect the eye position, and this leads to the generation of outliers in the data. Systematic or variable errors could be generated, which are due to inaccurate eye-tracking devices, or sometimes outliers are generated due to problems with calibration. The same errors are explained in the below sections.

4.2.1.1 Variable and Systematic errors

Variable errors are the dispersion that occurs in the data points around the fixation. Ideally, the fixation points need to be at a single position because that indicates the stability of the eye, but sometimes these points are spread due to the lack of precision of the eye tracking device. Variable errors can be treated by averaging the nearby gaze points to calculate the fixation. Systematic errors are a drift in the data points from the original fixation point. Systematic errors are due to the loss of calibration of the eye tracking device, which leads to a drift in the data point. Systematic errors can be observed to follow the same pattern throughout the data. Examples of both are given in Figure 4.2.

4.2.1.2 Eye tracker accuracy deteriorates

Sometimes the calibration is lost when the track of the pupil or the corneal reflection is lost. This may lead to data points generated outside of the image area, even though the participants are looking at the screen [92]. There are various experiments performed to address this type of error. Besides others things, authors suggest removing such points from the trajectory [93, 94], do calibration at the start of every session [95] or do recalibration in midsession [92].

4.2.2 Outliers due to natural reasons

Sometimes the outliers are generated due to some natural human movements, which are very difficult to avoid, like the head or body movements, blinking, and so on.

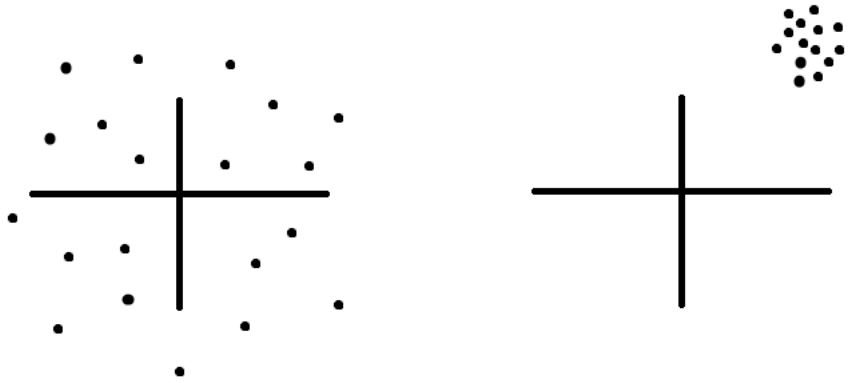


Figure 4.2: Left: Variable error showing spread of points Right: Systematic error showing points are shifted [91]

Errors generated due to such reasons are explained in this section.

4.2.2.1 Blinking

Blinking is a phenomenon in the human body that leads to the closing of the eyelid automatically after a particular time interval. Blinking can be divided into three types [96], the first one is a spontaneous blink in which the eyelid closes automatically without any internal effort or external reason. The second one is reflex blink; this usually happens due to some external reason, like when water is flashed on the face or touching the cornea. The third one is the voluntary blink, where a person closes the eyelid because of their own wish. These are usually for more time than the first two blinks. The blinking rate differs from person to person; for children, it is very less compared to that of adults. It usually increases throughout childhood.

To study the motion of the eye during a blink, a study has been made by [97]. It suggests that during the closing phase of blink, the eyes rotate downwards for an angle of 1-2 degrees. The eye movements during blinking are also faster than the original movements. Till the leads are closed, the downward rotation of eye movement is completed, and when blinking is finished, it again starts at its primary position. The upward rotation of the eye to get back to the primary position is usually missing. The downward rotation usually leads to extreme values and the closer time leads to missing values. Thus blinking leads to missing points or some extreme values in the data.

Usually, during eye-tracking, only the first and third types of blinking are visible because, as such, there is no external reason for blinking. The first type of blinking that is the voluntary blink can be recognized by studying the behaviour of a participant, like the missing data points or extreme values that may occur at a fixed frequency. These frequencies may vary from user to user. So a blinking rate calculated for one person can not be used for another. It is also not possible to predetermine the blinking rate without studying the trajectory.

Modern eye-tracking devices compensate for the blinking values, and many methods are available to deal with these blinking values like the extrapolation as mentioned in paper [98]. For the missing values in blinking, the neighbouring values are considered, that is, values before and after the missing points. Then these missing points are replaced with an average of the neighbouring points. Another effect can be extreme values, and to compensate them, taking mean or median is suggested. That is, extreme values can be either for the x-axis, y-axis, or both. So the axis with extreme value, mean of other values for that axis is calculated, and then the extreme value is replaced with this mean value.

These extreme values generated by the blink can be treated as outliers. Also, one significant research in blinking is that the extreme values or the missing data would be the same for both of the eyes as blinking occurs similarly for both eyes [99]. But it is very difficult to differentiate these extreme values due to blinking as such values can be generated because of head movements or system errors also. If blinking could be measured from the eye movement data, it would be great progress as each user has a different blinking rate, and this would help in user identification.

4.3 Eye movement segmentation

This section presents how the outliers are implemented in the existing eye movement segmentation algorithm. A modification of the previously used algorithm mentioned in 1 is necessary to classify outliers along with fixations and saccades. As can be seen in the Figure 4.3, the outliers can be divided into two parts. Outliers which are present outside the image area or outside the stimuli, and outliers that are present in the area of stimuli as shown in Figure 4.3. Both of these steps are explained in detail in the sections below. The newly made algorithm is called Outlier IVT. The input to this algorithm is the offset values, border point values, and the trajectory. Initially, the outlier positions are obtained, and then from these outlier positions, we get the final output in the form of sample start and sample type. After complete processing Figure 4.3 can be understood in more detail in the coming sections.

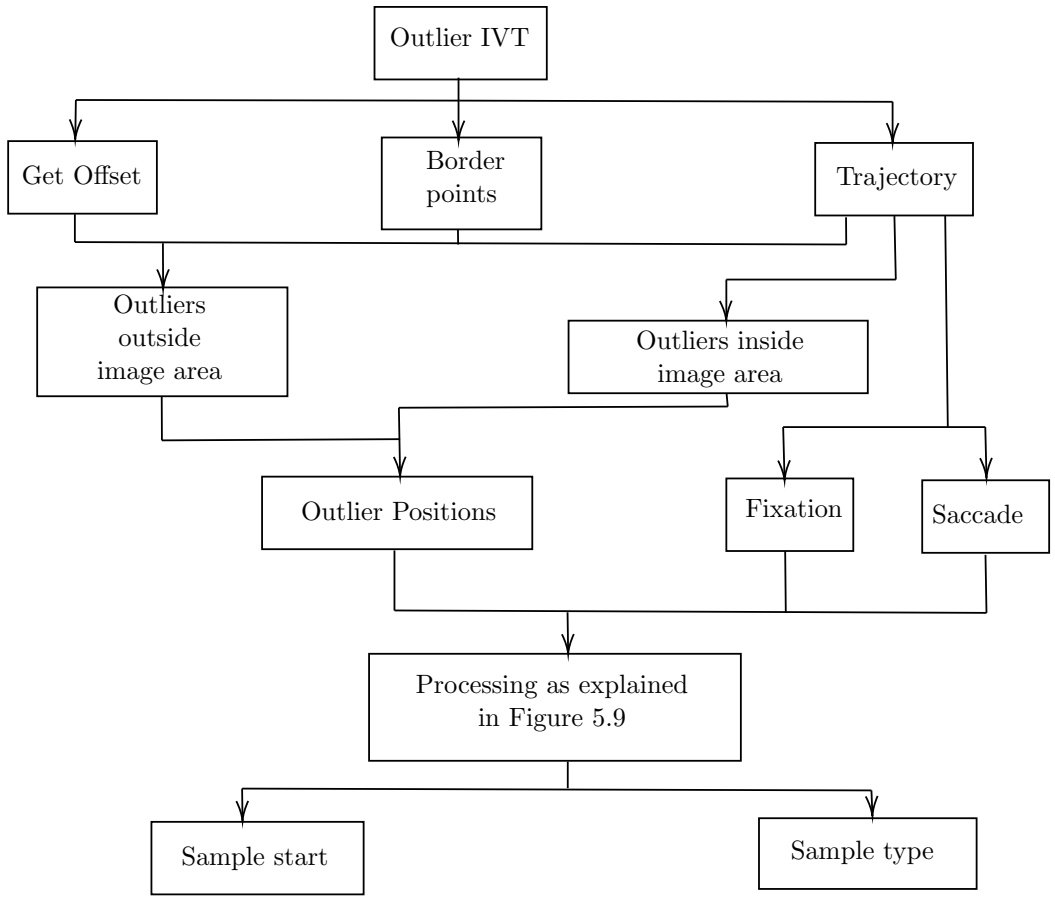


Figure 4.3: Outlier IVT

4.3.1 Outliers outside the image area

For outlier detection, the complete trajectory is considered as shown in the Figure 4.3. Initially, the outlier position needs to be calculated as depending on the position, the sample is decided to be part of fixation or saccade. If the outlier point is part of fixation in the original trajectory, then it is labelled as outlier fixation, and if the outlier point is part of the saccade in the original trajectory, it is labelled as outlier saccade. The three different inputs for detecting the outlier outside the image area are described below:

1. From the original trajectory, using the basic IVT algorithm, as mentioned in 1, fixation and saccades are calculated.

2. Offset values are values indicating how much the image needs to be shifted to

align with the trajectory on the axis. Offset values are different for all users, and they are provided in the dataset itself.

3. A small value is added to the image size to ignore the border points. Some points are present at border whose some part is present inside the image, and some part is present outside the image. For these points, a border value needs to be added so that these values are considered as part of the image.

Considering an image size of 1020×1680 as an example, then if any point on the y-axis is more than 1020 or less than 0, then it will be considered an outlier. Similarly, if the x-axis point is greater than 1680 or less than 0, then it will be considered an outlier. For example, the offset value and border values are not considered.

An explanation for each point in Figure 4.4 can be seen below:

- Point A is less than 0 for the x-axis; therefore, it is an outlier.
- Point B is less than 0 for the y-axis; therefore, it is an outlier.
- In the same way, point C is more than 1680; therefore, it is an outlier.
- Point D is greater than 1020 and so an outlier.
- Points I and J are within the region and satisfy all the conditions and therefore are not outliers
- Ponts E, F is also not outliers outside the image area. These points are maybe outliers inside the image area, and an explanation about the same can be seen in the Subsection 4.3.2
- Points G and H are exactly on the border and thus are also inside the image and also outside the image. To work with such points previously mentioned, border value is added. After adding the value, these points are now inside the image area.

So the new axis limits after adding the border value of 15 become from -15 to 1035 for the y-axis and -15 to 1695 for the x-axis. Similarly, considering the offset of 100 for both x-axis and y-axis, the new axis limit for the y-axis would be -115 to 935, and for the x-axis would be -115 to 1595. An illustration of the same can be seen in Figure 4.5.

4.3.2 Outliers within the image area

Detecting the outliers within the image is quite a challenging task. Precautions need to be taken to avoid detecting points within the range of interest. False detections could lead to false predictions and thus decreasing the accuracy. These points could

Figure 4.4: Points indicating outliers. Blue points: Outliers. Orange points:Border points. Green points: Maybe outliers inside the image. Pink points: Are not outliers

be generated because of various reasons, which are already mentioned in Section 4.2. For the extreme values, if a proper complete path is present as shown in Figure 4.6 by the colour yellow, then the person may be actually looking beyond the expected range of values. But if some random extreme points are present, then these points could be because of reasons like blinking shown by red colour in Figure 4.6. Such values should be considered as outliers.

Different types of algorithms can be used for outlier detection as mentioned in Section 2.5. But, from Figure 4.6, it is clear that the statistical algorithm will not work here as the major factor that needs to be considered is the nearby points. So different algorithms to find outliers using the density in nearby areas in mentioned in Subsubsection 4.3.2.1. Also, after using these algorithms, limits need to be added to avoid outlier detection in the main region of interest. Different methods to add limits are mentioned in Subsubsection 4.3.2.2.

4.3.2.1 Clustering algorithms

To check whether a path is present or not, the density of nearby points should be considered. It can be seen from Figure 4.6 that when a proper path is present, the density of nearby points is more. Therefore clustering algorithm like K-means, LOF, DBScan or OPTICS needs to be used which checks the density of nearby points.

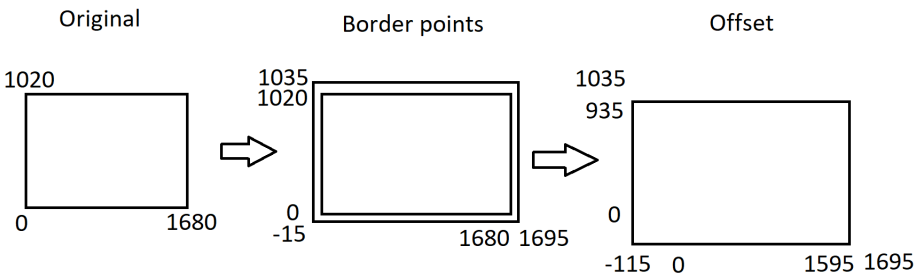


Figure 4.5: Changes in Image size. Left: Original size. Center: After adding border values of 15. Right: After adding offset of 100.

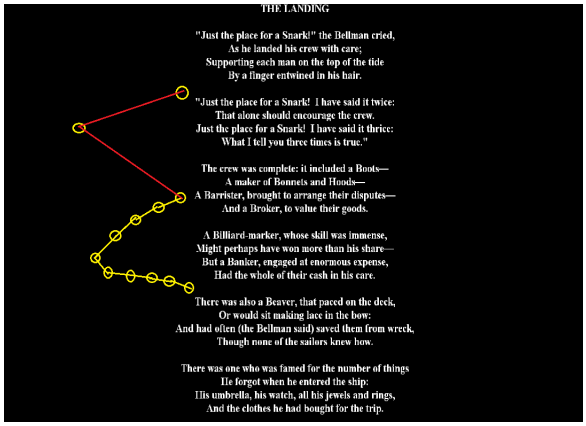


Figure 4.6: Differentiating outliers inside the image. Red: Random point outside indicating outlier. Yellow: Proper path present

K-means needs the number of clusters to be defined, and this cannot be done in the case of detecting outliers. Therefore density-based clustering [100] like LOF, DBScan, and OPTICS algorithm are studied further for finding the outliers.

DBScan Clustering

The DBScan (Density-Based Spatial Clustering of Applications with Noise) algorithm [65] considers the distance of data points from other data points. It forms a cluster of data points and then considers the density of the cluster. Initially, a cluster is formed for each data point. Then the number of samples in that particular cluster is calculated, and then it is decided whether the data point is an outlier or not. The main advantage of this algorithm is that it can form clusters in any shape like a circle, oval, concave, etc. The main parameters of the DBScan algorithm are the

min_samples and the *eps* (epsilon region). The epsilon is the distance to be considered around each data point to form a cluster, and the *min_samples* is the minimum number of data points that should be present in the cluster to classify the data point. If the *eps* value is kept too low, everything will be considered as a cluster, and if it is too high, everything will be considered as noise. Euclidean distance is calculated in the DBScan clustering[101]. All the data points present in the trajectory can be classified into three types:

1. Core point: If there is a minimum number of samples in the epsilon region, then the point is classified as the core point

2. Border point: If there are no minimum number of samples in the epsilon region, but a single core point is present, then the data point is classified as a border point.

3. Outlier: If there are no minimum number of samples in the epsilon region and neither is there any core point in the epsilon region, then the data point is considered to be an outlier.

It can be seen in Figure 4.7 the differentiation between core points, border points, and the outliers.

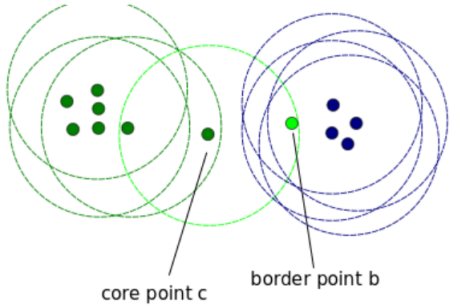


Figure 4.7: DBScan method showing core and border points [101]

OPTICS

Optics (Ordering Points To Identify the Clustering Structure) [102] is an extension to the DBScan algorithm present in the sklearn library. The difference between the two is that in Optics, only one parameter is needed: the minimum samples. It is very difficult to calculate the epsilon value. The epsilon value is calculated using the algorithm itself. These values are varied depending on the input. This type of algorithm is suitable for large datasets. The epsilon value may vary from user to user, depending on the type of data. Therefore it is necessary to select the correct

value. Two new concepts are added to the Optics that is the core distance and the reachability distance. Both of these concepts can be explained with the help of Figure 4.8

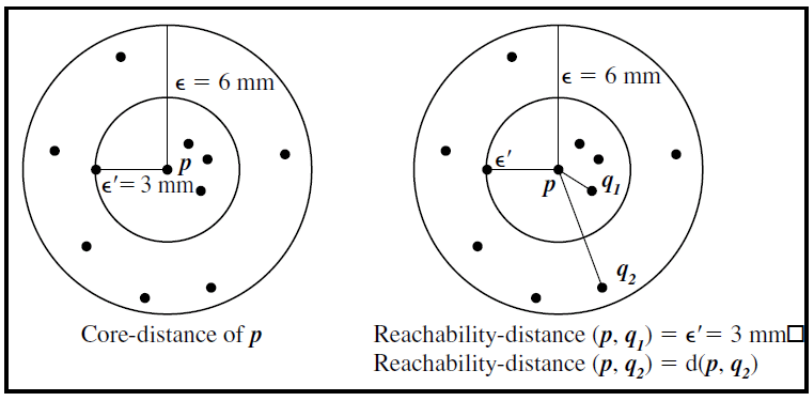


Figure 4.8: Core distance and reachability distance for OPTICS [102]

1. Core distance: Core distance is the minimum distance to a core point within the cluster, or it can be defined as the minimum value of epsilon that is needed to make a point as a core point. If there are no core points within the epsilon region, then these values stay undefined.

2. Reachability distance: For a point p in the dataset, its reachability distance to another core point in dataset O is the smallest distance between the points, which cannot be smaller than the core distance. This can be understood from the distance to points q1 and q2 from point p in the Figure 4.8.

Once clusters are formed, the reachability plot is calculated. The first step is the core distances are calculated, and then the reachability distances for each point. In this way, clusters are kept close to each other, and in this way, the Optics algorithm works. Optics has the major advantage that it automatically selects and varies the value of *eps* according to the data. The only major disadvantage of optics is that it doesn't have a well-defined concept of noise like the DBScan, and therefore it can fail sometimes.

Local Outlier Factor(LOF)
In LOF [103], the calculation is based on the reachability distance, which is the same as the one mentioned in OPTICS. So in LOF, also reachability distance is calculated for the nearby points. The average reachability distance tells how far are the nearby points from a particular object. The more the reachability distance, the farther are

the points from the object. That is, the average reachability distance is an indication of how far is the nearby cluster. Local reachability distance is the inverse of average reachability distance. LOF uses the concept of local reachability distance (LRD). A low value of LRD is an indication that the closest cluster is far away from the point. The formula for local outlier factor is given by Equation 4.1 which defines LOF as the ratio of average LRD of the K neighbours of a point X to LRD of X.

$$LOF_k(A) = \frac{sumxj \in N_k(A) LRD_k(X_j)}{\parallel N_k(A) \parallel} \times \frac{1}{LRD_k(A)} \tag{4.1}$$

To explain how these formula works are quite simple, if the average of LRD of neighbours is equal to LRD of point, then it is not an outlier. But if the average LRD of neighbours is more than that of the point, then it is an outlier. So if LOF is greater than one, then it is an outlier. Even sometimes, points close to the cluster are considered as outlier. This can become an advantage as well as a drawback, depending on the application.

4.3.2.2 Limits

By using just clustering algorithms, there are few outliers in the range of interest, that is, the area of the poem for Bio-Tex. Outlier detection should happen only outside this range. The probable reason for this is that during extremely fast eye movements, the path formed has points a little far from each other. Therefore, some limits should be added to exclude the range of the poem, and this may affect the accuracy. Various such limiting algorithms like adding manual limits, using standard deviation, using absolute deviation, using box plot method or percentile can be seen in this section.

Manual limits
Initially, manual limits were added that is points only outside the range of the poem will be considered for outlier detection. The range decided was outside 450 and 1250 as shown in the Figure 4.9. The limits have only been decided for the x-axis as for the y axis, the end of the poem is the end of the image, so the points become outside the image area. For other datasets, the y-axis range also needs to be defined.

Adding manual limits leads to making the code specific to a particular dataset. Limits should be added using some generalized method. This method could be some statistical method that can find points outside the range of interest. A few of the methods are discussed below.

Standard deviation method
Mean and standard deviation values are calculated for the complete set of data, and then data points in the range of more than standard deviation or twice the

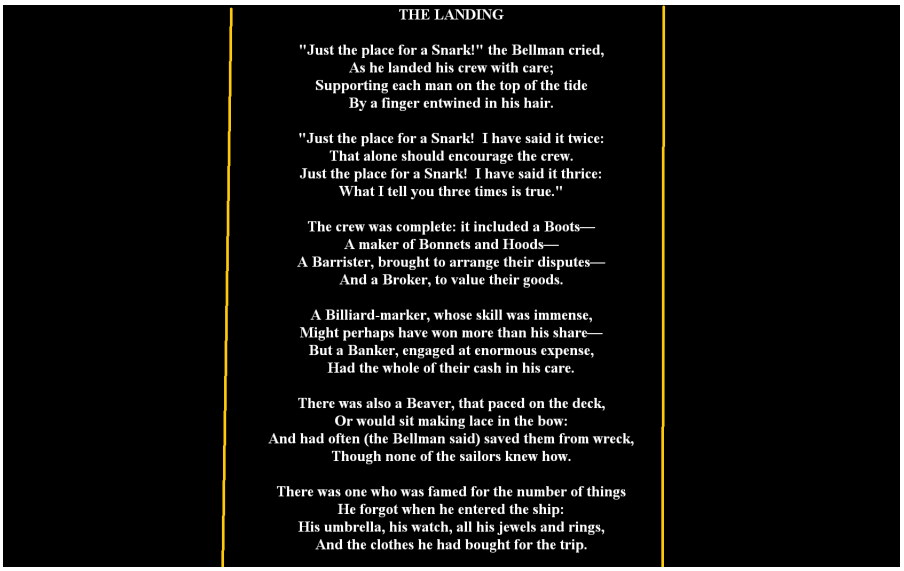


Figure 4.9: Image boundary after adding manual limits

standard deviation are considered as outliers. It can be seen from the Figure 4.10, points outside twice the value of standard deviation (5% of data approximately) as mentioned by authors of [104] can be considered as outliers. Or, in some cases, thrice the value of standard deviation (0.3% of data) as mentioned in [105], can be considered as outliers. So the choice actually varies depending on the application and data [106]. The main problem of using this approach is that the assumption is that the distribution is normal considering the outliers. Secondly, outliers affect the mean and standard deviation, and thus false detections can take place. Therefore for outlier detection, it is not a good choice as the basic consideration of mean value is itself affected. In this case, the standard deviation of 2 or 3 to be used will also vary depending on the user to user. Therefore, it is not a good indicator of the outliers.

Absolute deviation

Absolute deviation from the median is a methodology suggested by [107]. The reason for using the deviation value around the median is due to the fact that outlier values may affect the mean value. If an infinite value is present in the calculation, then the mean also becomes infinite, but the median remains unchanged. This is the reason for using deviation around the median values and not the mean. Median values will be affected if more than 50% values are infinite. The median absolute deviation(MAD) is not affected by the sample size. These properties made MAD suitable for detecting outliers [108]. The median absolute deviation works very well when used as median plus or minus 2.5 times the standard deviation for outlier detection [106]. In a similar

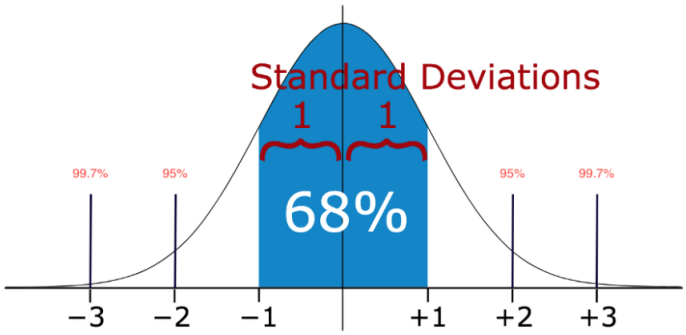


Figure 4.10: Standard Deviation method showing that after taking 2 standard deviation 5% data is excluded. After taking 3 standard deviation 0.3% data is excluded [106]

fashion, results can be calculated for mean absolute deviation [109]. It is similar to a median absolute deviation, with the only difference is that in the final step mean is calculated. Both of these are then taken around the median that is Median ± Mean absolute deviation or Median ± Median absolute deviation.

Box plot method

Box plot method is the one in which the interquartile method is used [110]. All the data is divided into 3 points and 4 intervals. The 3 quartile points are the lower quartile, the second one being the median of data, and the upper quartile. The lower and upper whiskers can be multiple of the interquartile point. Any data outside these whiskers are outliers Figure 4.11 explains the box plot method. The main advantage of this method is that it is simple, and it only takes the median into account, and the rest is calculated from the median. One problem with this approach is that points outside the whiskers need not be an outlier, but it can be a point that behaves differently from the majority of data [111, 112]. Another problem is that the quartile range cannot be tailored according to the requirement [113].

Percentile

The percentile of data can be calculated, which is the same as the quartile. 25 quartile is equal to 0.25 percentile. Percentile shows how much percentage of data lies below a particular score. For example, a score of 90 out of 100 may stand as 85th percentile as 85% of people lies below a score of 90. Thus percentile indicates the data distribution, that is, the amount of data below a particular score, and thus it is different from the percentage value. 50% percentile is the median value for that particular dataset. Therefore percentile can be a good indicator to calculate the limits. The formula for

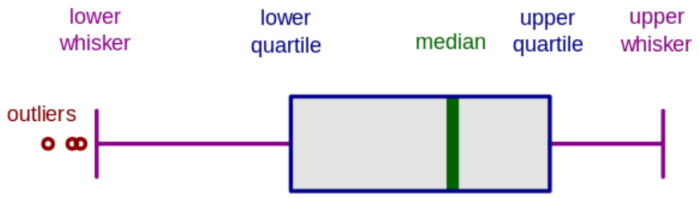


Figure 4.11: Box plot method showing quartiles and whiskers [111, 112]

percentile is given in Equation 4.2, but for applying the formula, the data must be sorted first, and then the formula must be applied.

$$\text{Value} = \frac{\text{Percent}}{100} * \text{Total number of values} \tag{4.2}$$

5 Implementation

This chapter describes the different methodologies used to achieve the results. One of the important tasks of the thesis is to detect outliers from the data and to segment these outliers into a new classifier along with the existing classifiers mentioned in Chapter 3. To achieve this, various methods have been implemented in each step of the pipeline mentioned in Figure 3.1. In this chapter, how the outliers are visualized will be studied in Section 5.1. There are two new proposed pipelines, one with the addition of one classifier and another with the addition of two classifiers. Details about the same can be seen in Section 5.2. Then the implementation part of the previously discussed Outlier IVT will be seen in Section 5.3. After this, the data augmentation part Section 5.4 will be discussed where details about dealing with users with missing outliers are discussed. Then the final part Section 5.5 is the evaluation where how the results from different classifiers can be merged is discussed.

5.1 Visualization

In the eye-tracking trajectory, outliers could be because of various reasons, like if a person is looking outside the image area, faults in the eye tracking device, or blinking. These outliers need to be viewed in the visualization along with the fixation and saccades. This section is an overview of how exactly these outliers are visualized. An additional option has been added in the visualization tool to view outliers as shown in Figure 5.3. It can be seen from Figure 5.1 that the dotted line is indicating the outliers. A similar visualization of outliers as part along with fixation and saccades can be seen in the heat map, too, as shown in Figure 5.2. The red color indicates the outliers. The red color outliers can be seen in the orange circle. Now the visualization indicates fixation with a solid line, saccade with a dashed line and outliers with a dot dash line.

For visualization, only the fixation points are considered. For viewing the fixation, directly the fixation points are available, but for viewing the saccade, a straight line is drawn between two consecutive fixation points. This makes the complete process quite simple of viewing the fixation and saccades. In the case of outliers, the only option to view outlier saccade has been added. So for viewing outlier saccade, outlier fixation points need to be joined with each other. The flowchart for the same can be seen in Figure 5.6.

The first step is to check if any outliers are present or not. If not, then print “no

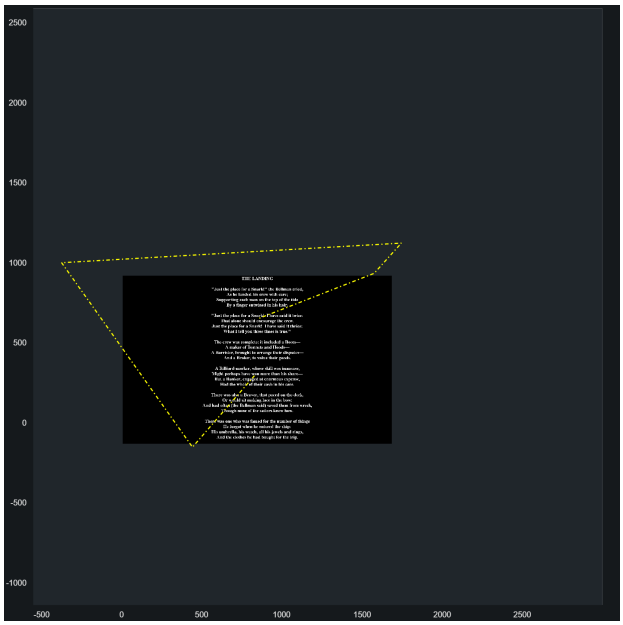


Figure 5.1: Outlier in Visualization

outliers present" in the text window. The next step is to check whether the first or the last element in the list is outlier fixation or not. If not, then as can be seen from step 2 of Figure 5.5 the fixation values needs to be replaced with nan values as we don't want to visualize the fixation. It is considered that fixation and saccade appear alternately. Therefore if the first outlier fixation is at ten, then we should add four nan values at positions 2,4,6,8. So each alternate position of fixation should consist of a fixation value if outlier fixation is present or then the nan. In the next step, for each fixation start, the previous value should be considered as shown in step 3 of Figure 5.5. If it is not considered, then the trajectory will be as shown in Figure 5.4. Here the points from which the outliers originate within the trajectory are not visible. Similarly, one future value should also be considered for the last fixation from the set. This can be seen from step 3 in the figure. This is the final vector available. The trajectory will now have its originating points from within the image as shown in Figure 5.4

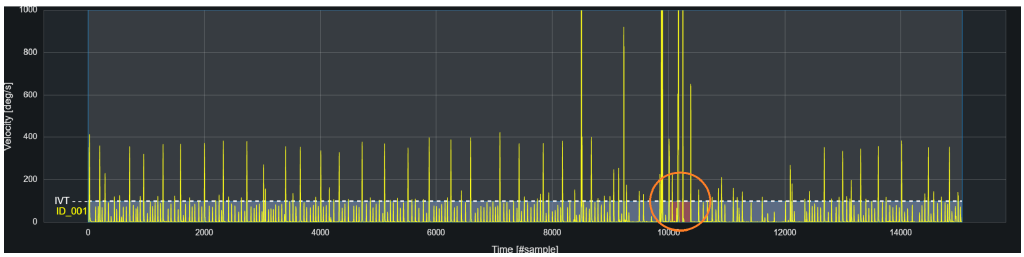


Figure 5.2: Outliers in heatmap indicated by red color in orange circle

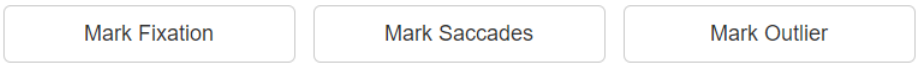


Figure 5.3: Mark Outlier button added along with option to mark fixation and outliers

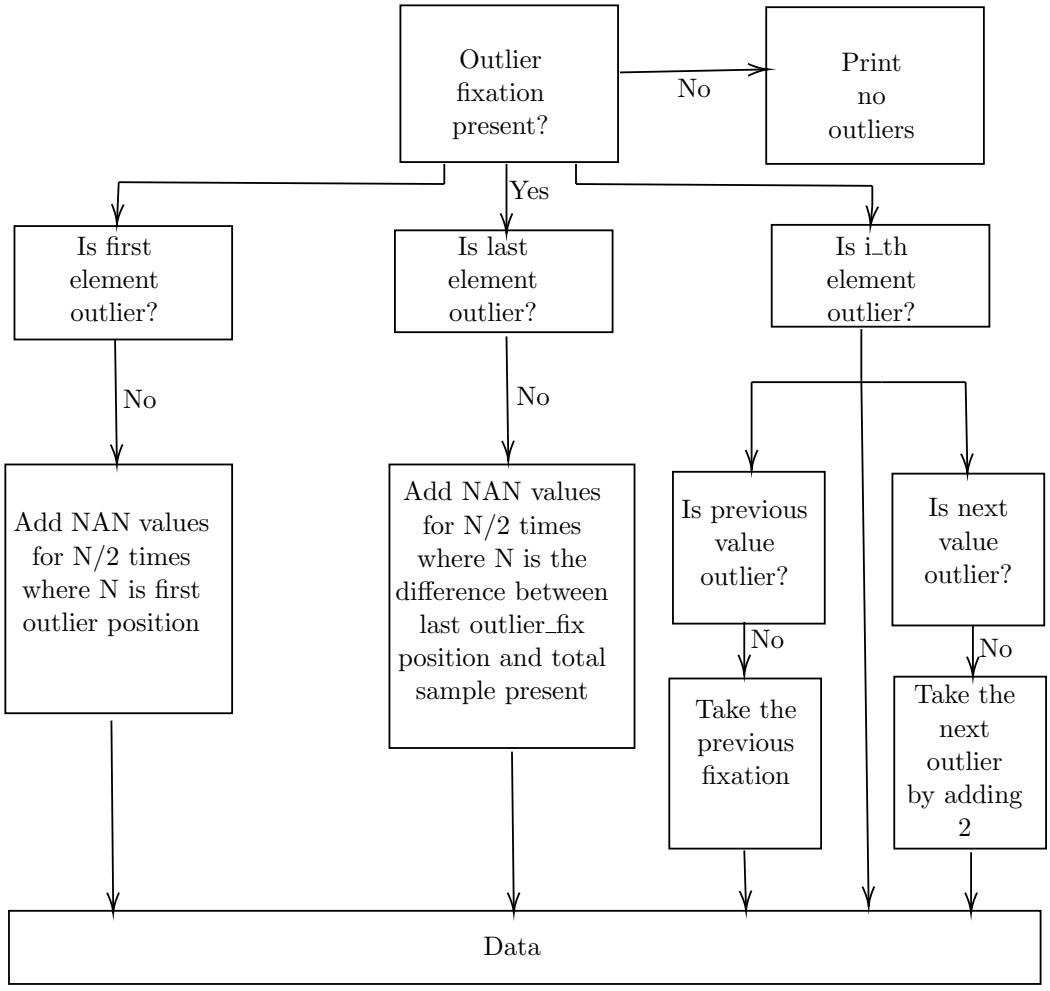


Figure 5.6: Flow chart for Outlier visualization

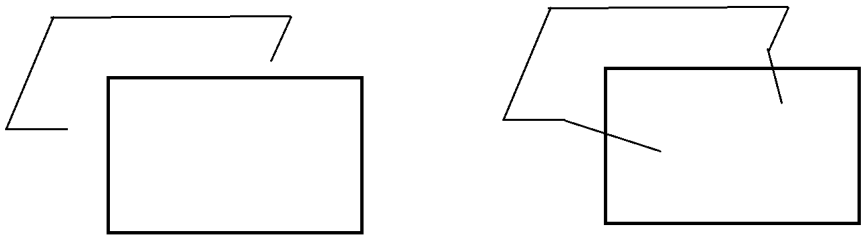


Figure 5.4: left:trajectory without considering the previous values, right:trajectory with considering the previous values

```
step1
[f1 f2 f3 f4 of1 of2 of3 f5 f6 f7 of4 of5 f8 f9 f10]
step2
[nan nan nan nan of1 of2 of3 nan nan nan of4 of5 nan nan nan]
step3
[nan nan nan f4 of1 of2 of3 f5 nan f7 of4 of5 f8 nan nan]
```

Figure 5.5: Examples of vectors after each step in visualization. f-fixation,of-outlier fixation

5.2 Proposed pipelines

This section consists of the two proposed pipelines for implementing the outlier detection by adding new classifiers into the pipeline for outliers. The basic blocks have been explained in Chapter 3. The two pipelines mainly differ in terms of the number of classifiers.

5.2.1 Three classifiers

In this method, one classifier is added to the pipeline as shown in Figure 5.7. This new classifier deals with the outlier fixation and outlier saccade in the single classifier itself. The output is, therefore, the average from all the three classifier results.

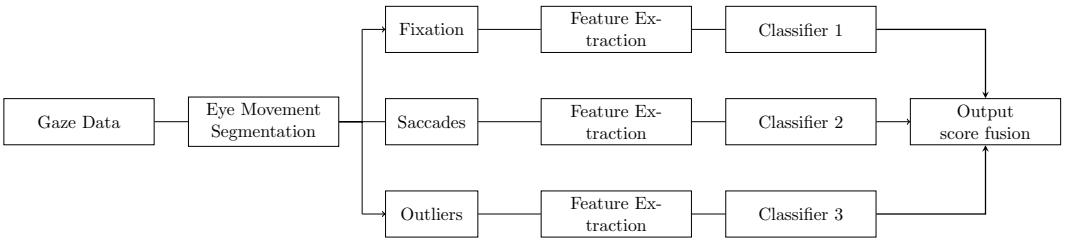


Figure 5.7: Proposed pipeline with 3 classifiers

5.2.2 Four classifiers

In this method, two more classifiers are added to the pipeline as shown in Figure 5.8. One classifier is for the outlier fixation, and the other classifier is for the outlier saccade. So now the output is the average of results from all the four classifiers.

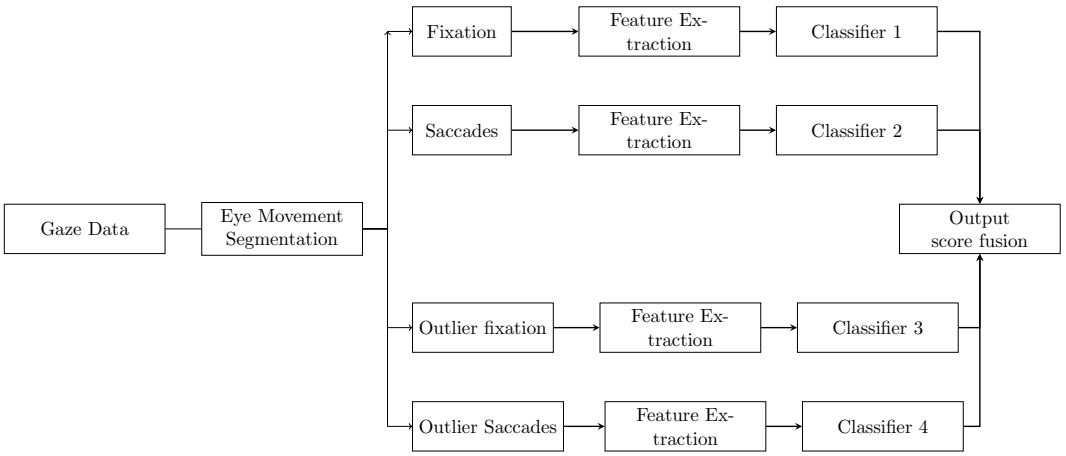


Figure 5.8: Proposed pipeline with 4 classifiers

5.3 Eye movement segmentation

The Outlier Ivt algorithm explained in Section 4.3 returns the positions of outliers using various algorithms. So it is necessary to find from these positions whether a particular outlier point belongs to outlier fixation or outlier saccade. Section 4.3 explains the theoretical part of various algorithms used. But the actual implementation of these algorithms will be studied in this section.

5.3.1 Classification of outliers into outlier fixation and outlier saccade

In this section, how exactly outlier fixation and outlier saccade can be extracted is discussed using the Figure 5.9. Fixation and saccades can be generated from the trajectory using the logic used in 1. For each position of an outlier, the position value needs to be searched in the available matrix of fixation and saccade. If the position value is in fixation, the particular outlier belongs to outlier fixation, and if the position value belongs to the saccade, then it is part of the outlier saccade. For example if fixation vector is as [1 2 5 6 10] and saccade vector is like [3 4 7 8 9]. Then if outlier position is 2 then it belongs to fixation. So the outlier fixation matrix will have a value 2 and original fixation matrix would be [1 5 6 10].

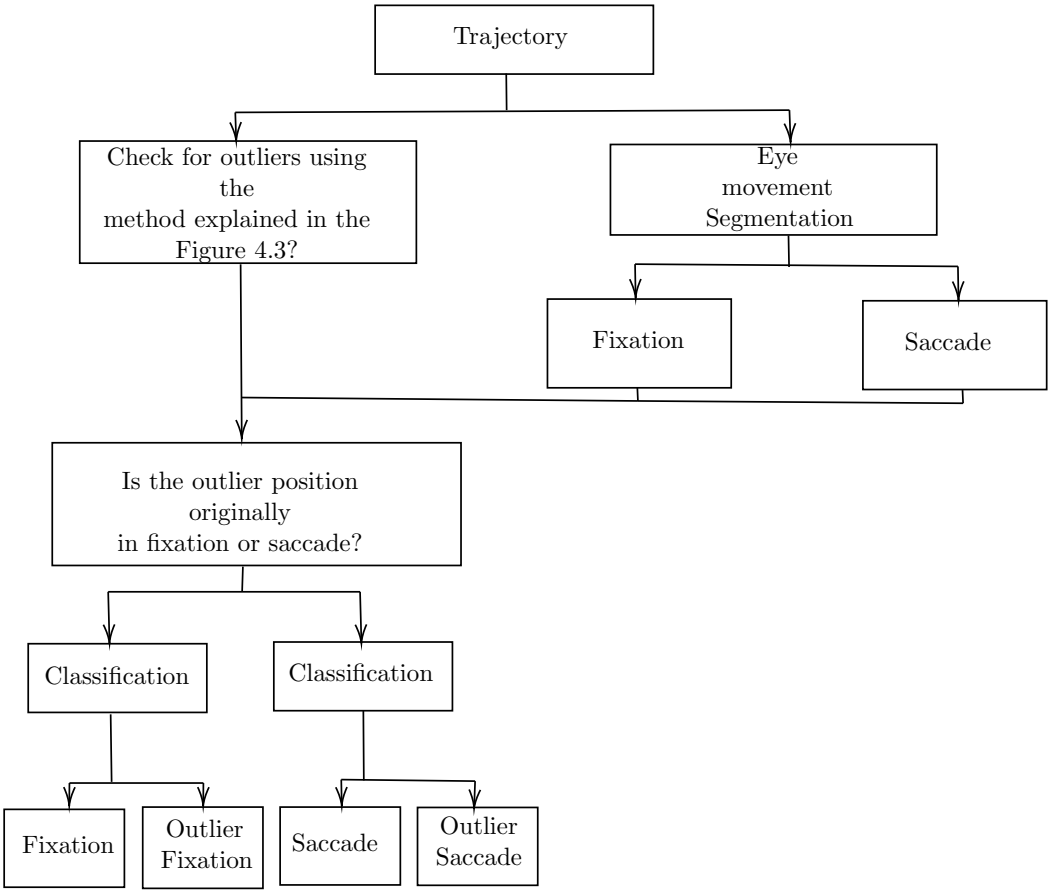


Figure 5.9: Outlier IVT algorithm

5.3.2 Implementation of different methods

This section gives an overview of implementing the methods explained in Section 4.3. This includes finding outlier outside the image area and also for the methods discussed to find outliers within the image area using clustering algorithm and limits. Most of the mentioned algorithms are directly available in the scikit learn library [114]. Scikit learn library is open source and available for all, and it is specially designed to deal with machine learning.

5.3.2.1 Outlier outside the image area

As already discussed, outliers outside the image area can directly be detected by considering the size of the image. A code snippet can be seen in Figure 5.10 where offset values (x offset value, y offset value) are added, and a border compensation value (15) is added to the image size (fov). The multiple loops are for working with positive and negative offset values. Once the position of outlier points is detected, these points are checked whether they belong to fixation or saccade. If they belong to fixation, then they are added to a new group called outlier fixation, and if they belong to the saccade group, then they are added to the outlier saccade group. In each of these cases, when these points are added to the outlier fixation or outlier saccade group, they are popped out from their original fixation or saccade group. In this way, the total number of points in the trajectory remains the same. Later a similar process is done in calculating the sample start and duration for outlier fixation and outlier saccade as it is done for fixation and saccade.

```
for li, item in enumerate(trajectories_outlier):
    if x_offset_value >= 0 and y_offset_value <= 0:
        if item[0] > (fov[0] - x_offset_value + 15) or item[1] > (fov[1] + y_offset_value + 15) or item[0] < (
                x_offset_value - 15) or item[1] < (y_offset_value - 15):
            outlier_positions += [li]
    elif x_offset_value <= 0 and y_offset_value <= 0:
        if item[0] > (fov[0] + x_offset_value + 15) or item[1] > (fov[1] + y_offset_value + 15) or item[0] < (
                x_offset_value - 15) or item[1] < (y_offset_value - 15):
            outlier_positions += [li]
    elif x_offset_value <= 0 and y_offset_value >= 0:
        if item[0] > (fov[0] + x_offset_value + 15) or item[1] > (fov[1] - y_offset_value + 15) or item[0] < (
                x_offset_value - 15) or item[1] < (y_offset_value - 15):
            outlier_positions += [li]
    else:
        if item[0] > (fov[0] - x_offset_value + 15) or item[1] > (fov[1] - y_offset_value + 15) or item[0] < (
                x_offset_value - 15) or item[1] < (y_offset_value - 15):
            outlier_positions += [li]
```

Figure 5.10: Code showing how offset values and border points are added

5.3.2.2 Outliers inside the image area

This section consists of two major parts the first is the clustering algorithm, and the second is the limits. The implementation for both is very easy. Any of the algorithms can easily be replaced using others in the code as only one or two lines need to be changed.

Clustering algorithms

For all the code snippets in this section, li1 indicates the trajectory, and a is the number of outlier points present, b is the position of outliers. DBScan algorithm can be used from the sklearn.clusters module [115]. As mentioned before, two parameters need to be set: *eps* and *min_samples*. A few other parameters which can also be configured are mentioned in [115]. For training, a complete trajectory is given, and the algorithm returns the positions with -1 where outliers are present. Therefore the positions need to be computed for the values with -1. The algorithm is quite simple to implement, and its snippet can be seen in Figure 5.11. The code implementation of OPTICS is as same as DBScan and can be seen in Figure 5.12. The only difference is that here only one parameter needs to be feed, that is, the *min_samples* as the *eps* is constant. The implementation of LOF is also the same, just the parameter here to be set is *n_neighbors*.

```
outlier_detection = DBSCAN(min_samples=3, eps=2)
clusters = outlier_detection.fit_predict(li1)
a = list(clusters).count(-1)
b = np.where(clusters == -1)
b = list(b)
```

Figure 5.11: Code snipet explaining DBScan algorithm

```
outlier_detection = OPTICS(min_samples=3)
clusters = outlier_detection.fit_predict(li1)
a = list(clusters).count(-1)
b = np.where(clusters == -1)
b = list(b)
```

Figure 5.12: Code snippet for implementation of OPTICS

Limits

For limits, 4 values are needed, two for the minimum and maximum of the x-axis and two for the minimum and maximum of the y axis. The code for this is also quite

```
outlier_detection =LocalOutlierFactor(n_neighbors=3)
clusters = outlier_detection.fit_predict(li1)
a = list(clusters).count(-1)
b = np.where(clusters == -1)
b = list(b)
```

Figure 5.13: Code snippet for implementation of LOF

simple. For manual limits, directly the values are added. The standard deviation is usually taken around the mean. The formula for mean and standard deviation can be seen in the Equation 5.1,Equation 5.2. The python implementation of the same can be seen in Figure 5.14.

$$\text{Standard deviation} = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{5.1}$$

$$\text{Mean} = \frac{\text{Sum of all elements}}{\text{Number of elements}} \tag{5.2}$$

The next is the Median absolute deviation, and steps for calculating the same are

```
value5=np.mean(trajectories_outlier[:,0])
value6=np.std(trajectories_outlier[:,0])
value7=np.mean(trajectories_outlier[:,1])
value8=np.std(trajectories_outlier[:,1])
```

Figure 5.14: Code snipet for calculating mean and standard deviation

given below:

1. Sort the dataset and find the median
2. Find the median from each data point
3. Find the absolute value of each number
4. Sort the numbers
5. Find the median for the new dataset

The python implementation can be seen in Figure 5.15. The box plot methods usually requires the calculation of the quartiles for the dataset. So 25th and 75th quartile

```
def mad(data, axis=None):
    return np.median(np.absolute(data - np.mean(data, axis)), axis)
value11=mad(trajectories_outlier[:,0])
value12 = mad(trajectories_outlier[:, 1])
```

Figure 5.15: Code snipet for median absolute deviation

```
value13=np.quantile(trajectories_outlier[:,0],0.75)-np.quantile(trajectories_outlier[:, 0], 0.25)
value14 = np.quantile(trajectories_outlier[:, 1], 0.75)-np.quantile(trajectories_outlier[:, 1], 0.25)
```

Figure 5.16: Code snipet for box plot method

needs to be calculated and the python implementation for the same is given in the Figure 5.16. The calculation of percentile is also easy and 98th and 2nd percentile of the data is calculated to set the range. Therefore values of 0.02 and 0.98 are used. The python implementation can be seen in Figure 5.17. The nanpercentile is used to ignore the nan values.

```
value1=np.nanpercentile(trajectories_outlier[:,0],2)
value2=np.nanpercentile(trajectories_outlier[:,0],98)
value3 = np.nanpercentile(trajectories_outlier[:,1],2)
value4 = np.nanpercentile(trajectories_outlier[:,1],98)
```

Figure 5.17: Code snipet for percentile

5.4 Data augmentation

Data augmentation is a method in which original data can be altered or modified to increase or decrease the size of the dataset. The problem with the current pipeline is that there may be users for whom there are no outliers. In this case, there will be no input given to the outlier classifier. This will result in an error being produced at the output. To avoid this problem, data augmentation must be done. For users with no outliers, a dummy vector needs to be created to give some value to the classifier. This value could be zero, mean value, or any other random value. More complex methods are also available to deal with such cases, but in this case, two basic methods that are giving zero value or giving mean value are studied.

Along with this, after adding a dummy vector, the data is still imbalanced and thus,

to balance the data, some technique should be used. There could also be cases where outliers are present in training and absent in testing and vice versa. A technique to solve this problem should also be implemented. All this can be seen in the sections below.

5.4.1 Dummy vector with zero

To deal with this problem of users with missing outlier values, a dummy vector of zeros is added. This is the most basic method. This dummy vector is added after the feature extraction stage. If the feature list is missing for any classifier, then a dummy vector is added for that user. By doing this, successful training and evaluation are done. The algorithm for the same is given below:

1. Check for available sample types for a user, which can be a fixation, saccade, or outlier. If a user has no outlier, then the sample type would be just fixation and saccade.
2. Check if an outlier is present in sample type. If yes, get out of the loop, and if no, go to the next step
3. Check for the feature values one by one
4. If feature name is sample type then this values cannot be zero, then add the value to be an outlier, and for all other features add values to zero
5. Add the new values to the feature list

5.4.2 Dummy vector with mean

Another approach to solving this problem is to add mean values [116]. Mean has the disadvantage that it reduces the variability of data, but again using a more complicated algorithm is not required here until and unless the results are quite bad. To implement this, nan values need to be added for the missing users, which is done by default. Then finally, when all features are gathered for all the users, the algorithm mentioned below needs to be executed.

1. Consider the first feature to be checked
2. Check if any value in the list for the feature name is missing
3. If there are no nan values, then just go to the next feature
4. If nan values are present, calculate the mean of values and just replace the nan values with the mean for that particular feature
5. Repeat the same for all features

5.4.3 Data balancing

It has been already discussed previously that the data presented to the outlier classifier is highly imbalanced. A few users have no outlier points, and a few users have many data points considered as outliers. Therefore data balancing could be the way to improve the accuracy. A few of the most common ways are to oversample or under-sample the data. In undersampling, the amount of data size for the majority class is reduced. In this case, undersampling will reduce the data points for users with more outliers. The other possible way is to resample the data. In this case, random points are added to the data to increase the size of the data. Adding random data points for users with fewer outliers may increase the accuracy. One more way is to do both undersampling and oversampling. This can be done by adding the oversampling for users with less number of outliers and undersampling for users with more number of outliers. This is the most effective way that can be used, but it has the same disadvantages as mentioned for under and oversampling. Data balancing could be done before presenting the data to the classifiers.

5.5 Evaluation

The final step in the pipeline is to combine the results from all the classifiers. All the classifiers at their output give a probability for each user. These probabilities are then multiplied by some weights, and then the final predictions are made. The two methods that can be used are averaging and weighting. In averaging, equal weights are given to all the classifiers, and in weighting, weights are given depending on the performance of each classifier.

5.5.1 Averaging

This method is originally used in SMIDA and is inspired by the work of [12]. In this paper, they have mentioned the formula for calculating the weights. The formula depends on the number of features used for fixation and saccades, and this decides the ratio of weighting for fixation and saccade. But as the number of features are the same length, the formula can be given as

$$\mathsf{weight} = \frac{1}{\mathsf{Number\ of\ classifiers}} \tag{5.3}$$

The final result is, therefore, an average of the single results

5.5.2 Weighting

The main disadvantage of averaging is the assumption that all classifiers perform in the same manner and predictions of all classifiers are equal and the same. But this

is not the case. After taking into consideration the probability estimation of each classifier, it proves that the prediction accuracy is different for each classifier, and therefore same weights cannot be given to all the classifiers. As the data size given to the outlier classifier is very low compared to that of fixation and saccade, therefore the prediction accuracy for these classifiers is also low. Therefore less weight should be given to this classifier. Experiments with different weights can be seen in Chapter 6.

6 Results and Discussion

In Chapter 5, all the different methodologies used in the implementation of the pipeline are explained. In this chapter, results produced by all the methods are compared and evaluated. All the results are discussed, and the final comparison of the results is presented in this chapter. All the results are the representation of accuracy in percentage. The results are for user identification. All the experiments are performed for five seeds with a user limit of 100. The reason for selecting these parameters was the time limitation and memory limitations. All the experiments are performed on the Bio-Tex dataset. In the final results, a comparison for the Bio-Ran dataset is available. Initially, the results are compared for the original pipeline with the outliers removed from the data in Section 6.1. In the next section, results for two proposed pipelines are compared in Section 6.2. In the next section, results for different clustering algorithms and different limiting algorithms are compared in Section 6.3, and the best method is then selected. Then the results for different data augmentation methods are compared in Section 6.4. In the next step, results for different classifiers are compared in Section 6.5. Then the results for averaging and different weights are seen and studied in Section 6.6. Finally, the achieved results by the pipeline are analysed and compared with the original results in Section 6.7.

6.1 Result for the original pipeline with outliers removed

Initially, the first task for the thesis was to detect the outliers in the data using the algorithms discussed in Chapter 4. These detected outliers were then removed from the data, and the accuracies were calculated for the pipeline with two classifiers (fixation and saccade). The results for the Bio-tex dataset can be seen in Table 6.2. The accuracies after removing the outliers got reduced for RF by 0.4% but increased for RBFN by 1.2% approximately. But for getting a more clear idea, experiments need to be performed on more number of seeds. The results for the same are present in Section 6.7 The reason for the decrease in accuracy is that training data was reduced considerably compared to the previous one. This can be seen from Table 6.1, where the count of fixations and saccades have reduced after removing the outliers. It can be seen from Table 6.1 that there are many outliers even in the image area because of some extreme trajectory values in the image itself. Table 6.1 are the count of fixation, saccade and outlier for 100 users. The results show that the maximum of outliers are within the image area, and mostly the outliers outside the image area

	Original	Outliers outside the image removed	All Outliers removed
Fixation	21825	21119	20001
Saccade	16330	15051	13356
Outliers	-	1985	4798

Table 6.1: Number of outliers for 100 users

are saccades. So we cannot conclude here whether it is an ideal choice to remove the outliers or not. Therefore in the next steps, we will add the new classifier which works with outliers, and in the end of the thesis, again, the results will be compared by removing the outliers.

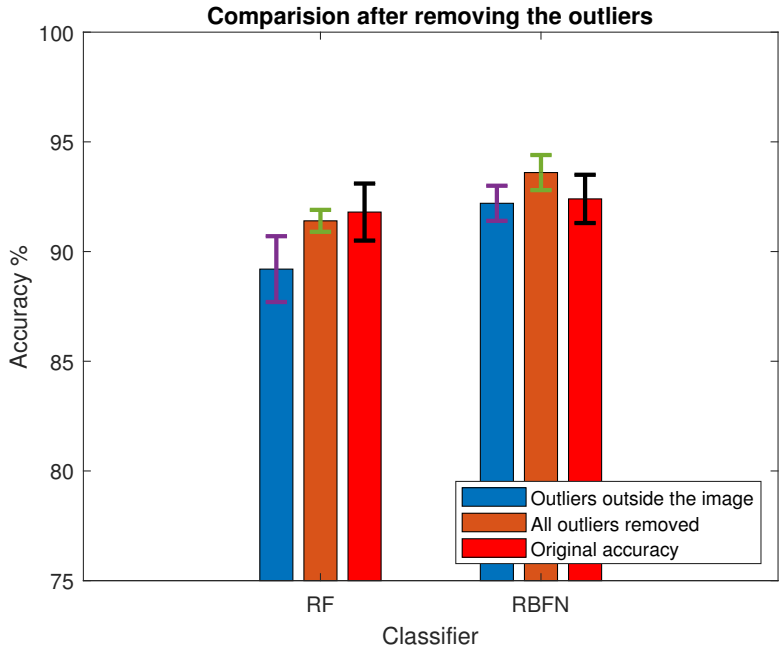


Figure 6.1: Comparision of accuracies after removing the outliers from training. Number of users used:100. Dataset: Bio-Tex

6.2 Results for newly introduced pipeline

In this section, a comparison of results for the two new pipelines that is one with three classifiers and the other with four classifiers, is presented. These classifiers are to deal with the outlier data present in the trajectory. As it can be seen from the Table 6.3

	Outliers outside the image removed	All Outliers removed	Original accuracy
RF	89.2% ± 0.7%	91.4% ± 0.2%	91.8% ± 0.6%
RBFN	92.2% ± 0.3%	93.6% ± 0.4%	92.4% ± 0.5%

Table 6.2: Comparision after removing the outliers from the pipeline. Number of users used:100. Dataset: Bio-Tex

	RF	RBFN
Pipeline with 3 classifiers	67% ± 0.5%	87% ± 0.8%
Pipeline with 4 classifiers	45% ± 0%	54.6% ± 0.4%

Table 6.3: Bio-tex accuracies for pipeline with 3 and 4 classifiers. 3 classifiers include fixation,saccade and outliers. 4 classifiers contain fixation,saccade, outlier fixation and outlier saccade. The results are for 100 users.

both the pipelines, don't seem to perform very well in the initial stages. But after comparing the results of both the pipelines, the pipeline with three classifiers seems to perform better. The results have such a low accuracy because of equal weights used for all classifiers. The concept of weighting is discussed in Section 6.6. Outlier classifier for RF does not perform well and decreases the accuracy. The reasons for the failure of pipeline with 4 classifiers are given in below:

- It has been observed that for most of the users, the outliers are in the form of saccades, and therefore the classifier which works on fixations has a less amount of input data. For 100 users, there are 1824 outlier fixations, while there are 2974 outlier saccades.
- Addition of dummy vector for missing values is reducing the variability of data, and mean values are present for many users.
- The data is highly imbalanced, for some users, there are many outliers present, and for some users, there are no outliers present.

The pipeline with 3 classifier also doesn't seem to perform very well, but it performs better than the one with 4 classifiers.

6.3 Results for Outlier IVT

All the methods used for Outlier detection are mentioned in Chapter 4. A comparative study of the clustering methods used and the limiting methods used can be seen in this section. All the outliers and detected and considered for experiments in this section. The clustering methods are compared, and then one is finalized to be used

	RF	RBFN
DBScan	93.6% ± 0.2%	93.2% ± 0.5%
Optics	89.2% ± 0.6%	92.8% ± 0.2%
LOF	92.4% ± 0.2%	92.6% ± 0.2%

Table 6.4: Accuracies for clustering algorithms on Bio-Tex dataset with a user limit of 100

for further experiments. Once the clustering method is selected, different limiting algorithms are used in combination with the clustering method, and one is selected.

6.3.1 Different methods for outlier detection

The main methods compared are DBScan, LOF, and Optics. As mentioned above, Optics is an improvement to the DBScan clustering algorithm. The main advantage of Optics is that only one parameter needs to be set, that is, the *min_samples*, while for the DBScan, two parameters need to be set that is the *min_samples* and the *eps*. *Eps* is automatically calculated in the Optics algorithm. A comparison of all of these algorithms can be seen in the Figure 6.2. Table 6.4 shows the accuracies achieved for all of these algorithms.

The results for all the three clustering algorithms have been discussed below:

- **DBScan**: DBScan considers the outliers in its calculation as mentioned in Subsubsection 4.3.2.1, it does a better prediction and classification of points to be outliers. So even though Optics may be a better clustering algorithm, DBScan performs better for outliers.

- **OPTICS**: It can be seen that accuracies decrease significantly for Optics. These results are concluded after doing a comparative study for different values of *min_samples* from very small values of 1 to larger values of even 100. The main reason here for the decrease in accuracy is that Optics does not consider the noise as mentioned in Subsubsection 4.3.2.1. It just clusters the incoming data based on the density. Therefore not all outliers are detected, or it may happen that some false outliers are detected.

- **LOF**: While in the case of LOF, some points close to the cluster of fixation and saccade are also considered as outliers. Due to this, there are some false predictions for outliers, and this decreases the accuracy.

Hence for further experiments, the DBScan algorithm is used. To further select the parameters of the DBScan algorithm, experiments were performed by changing the values of *min_samples* and *eps*. The Table 6.5 shows that the value of 3 for

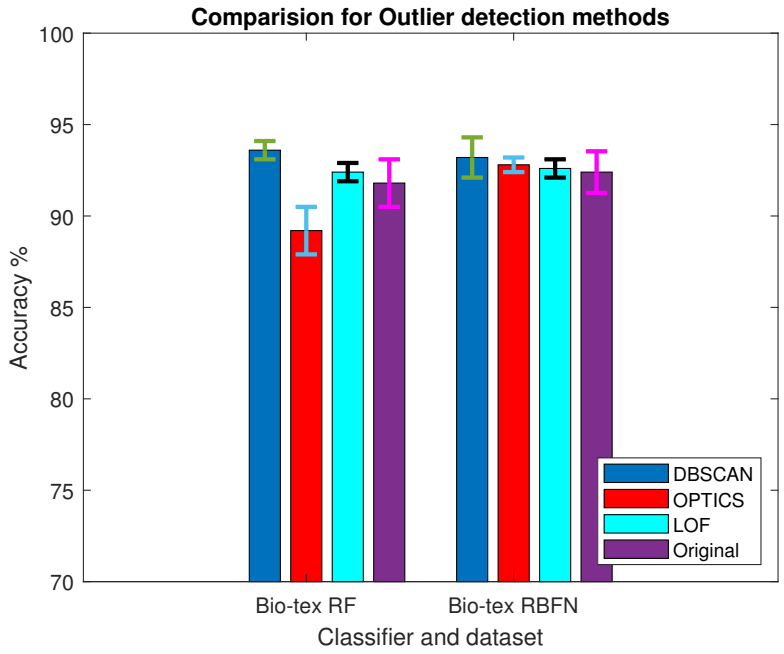


Figure 6.2: Comparision of accuracies for DBScan,OPTICS, LOF and original accuracies for 100 users

min_samples and 2 for *eps* gives best results. The values for *min_samples* and *eps* are decided by trial and error method. A value for epsilon region more than this may lead to some distinct points getting detected, and values less than this may lead to some points not getting detected and points getting falsely labelled as outliers. So these values were used for further experiments.

6.3.2 Different methods for adding limits

As discussed in Subsubsection 4.3.2.2 limits can be added to where exactly the DBScan algorithm should be applied. Various methods have been discussed along with their pros and cons in Subsubsection 4.3.2.2. Results for all these algorithms have been discussed and compared in this section. The different values tried for different parameters of each of the algorithm are given below:

- **Manual limits**: For manual limits, limits of between (450,1250), (500,1300), (400,1200), (400,1300), (500,1200) were tried and the results shown in Table 6.6 are for the range of (450,1250).
- **Standard Deviation**: For Standard deviation ranges of (Mean ± sd), (Mean

DBScan parameters		Classifier	
eps	**min_pts**	**RF**	**RBFN**
2	**3**	93.6% ± 0.2%	93.2% ± 0.5%
4	**4**	87.8% ± 0.7%	92% ± 0.5%
2	**2**	88.8% ± 0.6%	93.2% ± 0.4%
3	**2**	87.8% ± 0.4%	93.2% ± 0.4%

Table 6.5: Accuracies for different DBScan parameters that is *min_samples* and *eps* for 100 users on Bio-Tex dataset

	RF	**RBFN**
Manual limits	89.6% ± 0.2%	92.6% ± 0.2%
Standard deviation	91.2% ± 0.6%	92.2% ± 0.4%
Absolute deviation	92.4% ± 0.2%	93.4% ± 0.5%
Box plot	92.8% ± 0.2%	92.6% ± 0.4%
Percentile	93.6% ± 0.2%	93.2% ± 0.4%

Table 6.6: Accuracies for RF and RBFN for different limiting methods after using DBScan algorithm with them. Results are for Bio-Tex dataset for 100 users.

± 1.5×sd), (Mean ± 2×sd) were experimented, and the best results were given by Mean ± 1.5×sd.

- **Absolute Deviation**: The same limits of standard deviation were used for absolute deviation.
- **Box plot**: For the box plot, the interquartile range(IQR) was calculated by taking the difference between 25 and 75 quartile and then (Median ± IQR), (Median ± 1.5×IQR), (Median ± 2×IQR) and the best results were received while multiplying the IQR with 1.5.
- **Percentile**: For percentile also ranges of (2,98),(1,99),(3,97), and few other random values were tried, but (2,98) provided the best results.

Table 6.6 shows the comparison of the different limiting algorithms. Here the results are for RF and RBFN based on the DBScan clustering algorithm. A graph of the comparison can be seen in Figure 6.3. It can be seen from the results mentioned in Table 6.6 that the Percentile and median absolute deviation tends to perform much better than the rest. Results for all the methods have been individually discussed below:

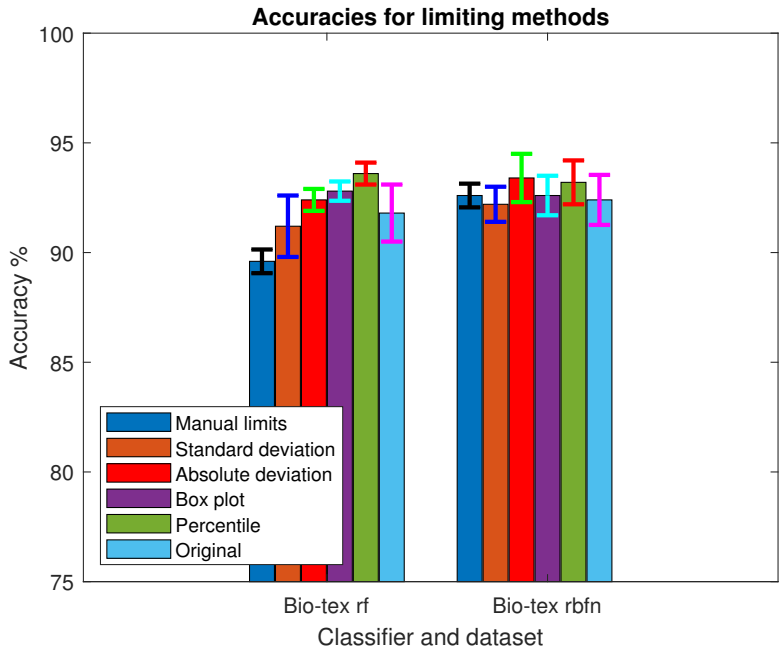


Figure 6.3: Comparision for different limiting algorithm with original accuracy for 100 users

- **Manual limits**: The manual limits have the biggest disadvantage that it does not take into the consideration the distribution of data. So it may happen that there are many data points outside the limits, and they all may be considered for outlier detection for DBScan, which is not necessary. Therefore the limits may need to vary for each user. Another problem with using this method is that the limits can be used only for this dataset. For all other datasets, these limits may be different. So this is not the right approach to go forward with.

- **Standard Deviation**: For the standard deviation method and absolute deviation method, the mean is considered, and as already discussed, the value of the mean is largely affected by the presence of outliers. For the scenario where the number of outliers is less, this may still work well, but when outliers are more, this method tends to fail.

- **Absolute Deviation**: The absolute deviation performs better for the RBFN than all other methods, but for RF, it fails to perform better than the rest.

- **Box plot**: The box plot method uses the quartile range as discussed previously. It performs again well but still is less good than the percentile method.

	RF	RBFN
Dummy vector with zero	92.6% ± 0.2%	92% ± 0.4%
Dummy vector with mean	93.6% ± 0.2%	93.2% ± 0.5%

Table 6.7: Comparision for dummy vector with mean and zero values for 100 users for Bio-Tex dataset

- **Percentile**: The percentile method shows very good results, and also the standard error of the mean of results for this method is not so high, and this is thus the best method among all. The reason for good performance is that only the data points outside the range of interest are considered.

6.4 Data augmentation

In this section, various methods to deal with missing users are discussed. One way of dealing with missing users is to add a dummy vector. The next thing is the imbalance in the dataset. One way to deal with this imbalance is balancing the data using resampling or undersampling. A few other experiments were also performed to deal with missing users or to deal with users with fewer outliers. All these things are explained in the section below.

6.4.1 Dummy vectors

Dummy vectors are needed to compensate for the users with no outliers. Because a user with missing features cannot be given to the classifier. Dummy vectors first with zero value were added, and then with mean value were added to the pipeline. Table 6.7 shows the difference in values for both these dummy vectors. The dummy vector with the mean value tends to perform better. The results are for both classifiers RF and RBFN. In Table 6.6 also dummy vector with mean value is used.

6.4.2 Data balancing

Data balancing was performed using different methods like resampling, downsampling, or a combination of both. But none of these worked. The most probable reason for this type of performance is that resampling the data for users whose outlier is absent is just like resampling the mean value, and thus, it does not work, and in the same way, downsampling will reduce the already fewer data available for training. And because of these two reasons, the combination of both also didn't work. It has been stated and proven that resampling fails when there is very low data to resample, which is true in this case [117]. Therefore data balancing was not added into the pipeline.

	Accuracy for IVT after removing users	Accuracy for Outlier IVT after removing the users
RF	84.8% ± 0.2%	86% ± 0.6%
RBFN	88.2% ± 0.2%	89% ± 0.3%

Table 6.8: Accuracies after removing the users with fewer outliers for 100 users for Bio-Tex dataset

6.4.3 Removing users with less/no outliers

In the initial stages, when the new pipeline didn't perform well, one obvious guess for the reduction in the accuracy was that users with fewer outliers reduced the efficiency of training, and thus, one possible approach to confirm this assumption was to remove these users from the pipeline. There were few users with no outliers in training but outliers present in testing data or the users with outliers in testing but no outliers in training. These users lead to the false prediction of other users. The assumption was proven to be correct that when these users were removed, the performance of the pipeline was improved. The Table 6.8 shows a comparison for a new set of users (users with fewer outliers replaced with the new one). The comparison is for IVT and Outlier_IVT for this new set of users.

6.4.4 Training users with no outliers on only two classifiers

After it is proven that the users with fewer/no outliers affect the training, the next possible step could have been to train these users on only two classifiers that is fixation and saccade. The assumption was that as these users are not considered in the outlier classifier group, the performance of the outlier classifier will improve. But surprisingly, this didn't happen. The performance remained the same. Therefore this step was not added to the pipeline.

6.5 Classifiers

There are two classifiers used in the initial pipeline, one for fixation and one for saccade. A third classifier is added to the pipeline for outliers. The parameter rbfn_k, which defines the cluster size, should have the inputs equal to or more than the cluster size. That is, if the cluster size is set to the default value of 32, each user should have a minimum of 32 samples for each classifier. A cluster size of 32 will divide the whole set of data into different cluster groups, and each group will have a similar type of data. But this is not the case; there may be some users with no outliers or fewer outliers. So, it is not possible to train the classifier with a cluster size of 32. So the solution for this problem is that cluster size could be set to 1.

	RF	RBFN	RF+RBFN	SVM	Naive-bayes
Accuracies	93.6% ± 0.2%	94.4% ± 0.2%	93.6% ± 0.2%	18.8% ± 0.6%	13% ± 0.0%

Table 6.9: Accuracies for different classifiers for 100 users for Bio-Tex dataset

Clusters are important to place similar data in a single cluster group but setting the cluster size to 1 lead to all types of data into a single group which reduces the accuracy. Therefore cluster size of 1 is used for the outlier classifiers, and for fixation and saccade classifier, the default value of 32 will be maintained.

It has been observed in the initial stages that the RBFN classifier performs better on the outliers than that of RF as seen in Table 6.3. RF produced very bad results while working with outliers. Due to this, an experiment was performed in which RF was used to train fixation, and saccade classifier and RBFN were used to train the outlier classifier.

Along with this, to see the working of different classifiers on the current pipeline, few other classifiers like Naive Bayes and SVM were also tried. A comparison of all the results can be seen in the Figure 6.4.

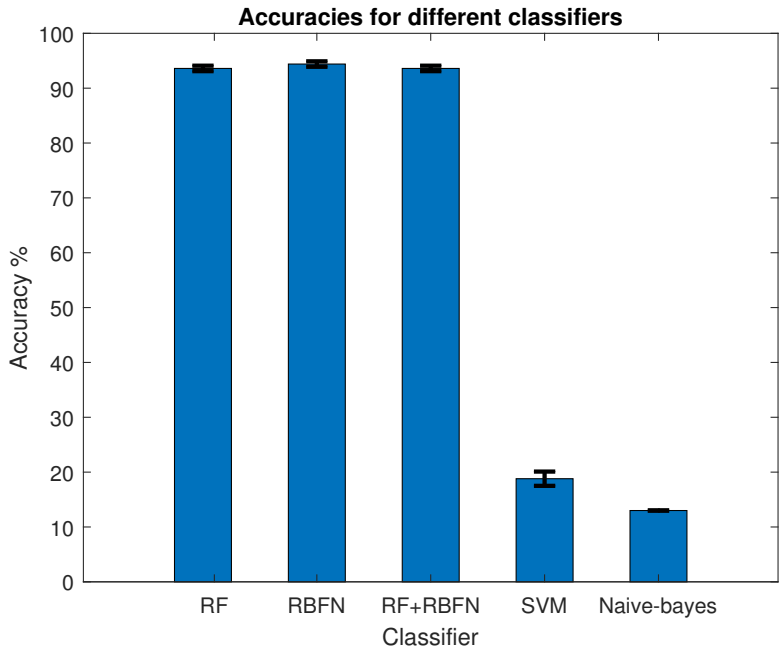


Figure 6.4: Comparision of different classifiers for 100 users for Bio-Tex dataset

It can be seen in Table 6.9 that RBFN performs best among all the classifiers. The reason for the poor performance of classifiers are discussed below:

- **SVM**: As it can be seen, SVM performs very badly. The data present in the dataset is highly imbalanced in some cases. In these cases, the ratio between positive and negative support vectors also becomes imbalanced and thus leads to false predictions. Also, as mentioned by the authors of [118], SVM is highly sensitive to the presence of outliers, and thus it performs badly. SVM also suffers when the training size is small, as mentioned by [119]. SVM is mainly used for binary classification, that is when output can be only two classes, but in this case, SVM is used for multiclass classification, and thus it fails. This may be some of the reasons for really poor performance.

- **Naive-Bayes**: On the other hand, Naive-Bayes is also producing very poor results. This may be due to the fact that it considers each feature independently and assumes that features are independent of each other, which is not the case [120]. It is also observed that it is a bad estimator, and thus, probability output is not accurate for consideration.

- **RF+RBFN**: RF+RBFN was considered in the initial stages of experiments when both RF and RBFN didn't perform considerably well. At this time, RF+RBFN performed better than the rest. But in the later stages, bot RF and RBFN performed really well, so there is no more need to consider this classifier because both RF and RBFN individually perform better than this combination.

RF and RBFN are highly accurate, and thus both of these classifiers are considered for most of the experiments.

6.6 Evaluation

In an evaluation in the first case, weights for all classifiers are kept the same, and in the second case, weights are varied for each classifier. When weights were varied for classifiers, the accuracy was highly improved, the reason being that all classifiers don't perform in the same manner.

It is very difficult to decide which weights should be assigned to each classifier. To solve this problem, the weights of fixation, saccade, and outliers were varied over a range, and the best weights were calculated. The addition of fixation weight and saccade weight at each point gives the outlier weight, as can be seen on the graph. The addition of fixation weight of 40 and saccade weight of 40 gave an outlier weight of 20. In the plots in Figure 6.5,Figure 6.6 the performance of RF and RBFN for different weights can be seen. It can be seen that for RBFN, saccade weight in the range of 50-60 and fixation weight in the range of 30-40 performs better. Similarly, for RF, saccade weight in the range of 30-40 and fixation weight in the range of 45-55 shows better performance than others. So this range must be explored more. It can

be seen that the performance of RF is better on fixation than on a saccade, and for RBFN, performance is better on saccade than fixation. Therefore weights should be calculated accordingly.

A calculation was done for over 20 seeds for this range, and it was found that for RF, the weights of 35 for saccade, 52 for fixation, and 13 for outliers achieved the best results. On the other hand, for RBFN, the weights of 52 for saccade, 36 for fixation, and 12 for outliers gave the best results. It can be seen that less weights are assigned to the outlier classifier, but their presence is significant and improves the performance.

A comparison of values by giving equal weights to all classifiers and giving the above-mentioned weights for all classifiers is made in Table 6.10. A comparison of the results for both of these with the original accuracy can also be seen in Figure 6.7.

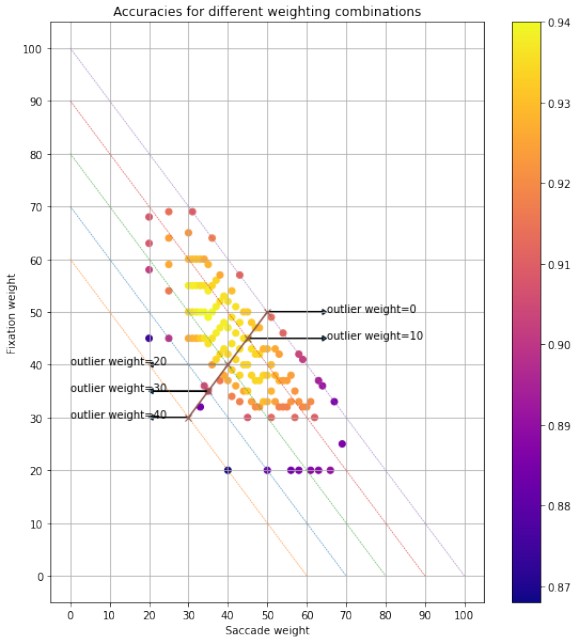


Figure 6.5: Results for different weights used for three classifiers for Bio-Tex for 100 users for RF. Color map indicates the accuracies. Fixation weight + Saccade weight + Outlier weight = 1

6.7 Final results

Initially, the experiments were performed by removing the outliers from the calculation. Then in the next step, the two proposed pipelines were used, one with 3

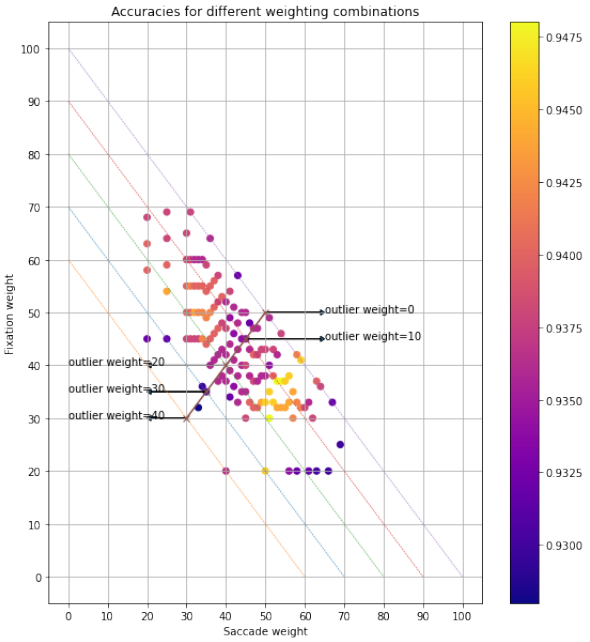


Figure 6.6: Results for different weights used for three classifiers for Bio-Tex for 100 users for RBFN. Color map indicates the accuracies. Fixation weight + Saccade weight + Outlier weight = 1

classifiers and one with 4 classifiers. The pipeline with 3 classifiers was seen to outperform the pipeline with 4 classifiers. This shows that dividing outliers into fixation and saccade doesn't provide any added valuable information. Instead, using them together is a better option. Then for detecting the outliers, different methods were tried and compared. Among the methods used for clustering, DBScan provided the best results due to its ability to consider outliers. Among the limiting algorithms, percentile was the best algorithm as it is very less affected by the presence of noise. In the data augmentation part, various methods to deal with users with no outliers were discussed. Among the methods for data augmentation, adding a dummy vector with a mean value for users with no outliers was used. Then different classifiers were

	RF	**RBFN**
Same weights to all	89.4% ± 0.2%	92.8%± 0.2%
Different weights to all	94% ± 0%	95% ± 0.3%

Table 6.10: Comparison of accuracies for averaging and weighting method for Bio-Tex dataset for 100 users

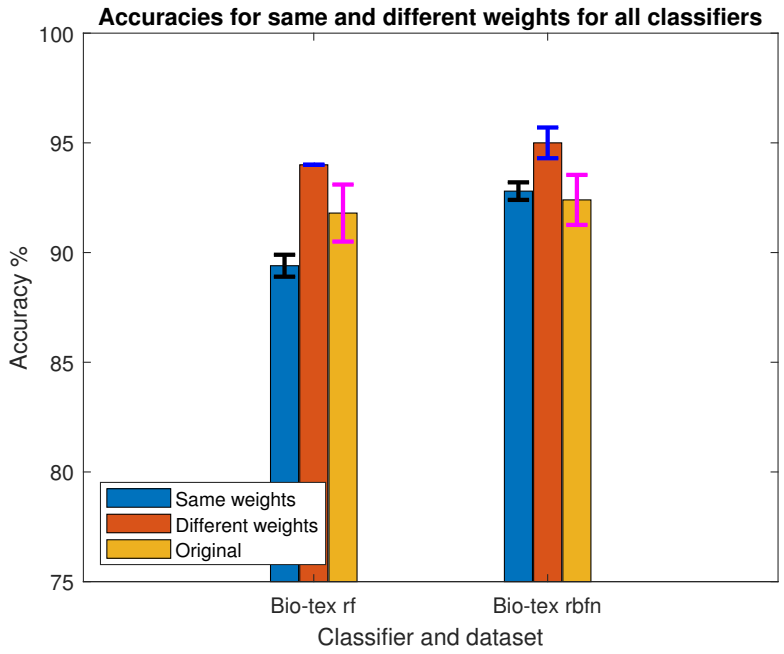


Figure 6.7: Comparision of accuracies for same and different weights. Same weights: 0.33 for all. Different weights(Saccade, Fixation, Outlier): Bio-Tex RF-35,52,13, Bio-Tex RBFN-52,36,12, Bio-Ran RF-37,41,22,Bio-Ran RBFN:50,44,06

compared for their performance on the outliers, and it was found that RF and RBFN perform better than the others. Therefore the final results presented are for both of these classifiers. Then in the final step, weights were calculated for all the three classifiers that give the best results. The received weights were then used in the final results.

The comparison of results for this final pipeline with the original pipeline and the pipeline with outliers removed can be seen in Table 6.11 and Figure 6.8. It can be seen in the results that for Bio-Tex, there is an improvement in accuracy by 1.75% for RF and for RBFN, the accuracy is improved by 1% after considering the outliers. This proves that outliers play a vital role in the identification process. Even when outliers are removed, the accuracy decreases slightly by 0.5% for RF but increases for RBFN. This shows that outliers are affecting the prediction for fixation and saccade classifier for RBFN. But in the case of Bio-Ran, the accuracies have not improved and are approximately the same as the original accuracies. In fact, for RF, the accuracies have decreased by approximately 6%. So for both the datasets, removing the outliers is decreasing the accuracy for RF. The reason may be that in Bio-Tex,

	RF			RBFN		
	Original	**Outlier removed**	**New**	**Original**	**Outlier removed**	**New**
Bio-Tex	92% ± 0.2%	91.45% ± 0.2%	93.75% ± 0.1%	92.4% ± 0.1%	93.35% ± 0.2%	93.4% ± 0.2%
Bio-Ran	93.8% ± 0.2%	87.7% ± 0.2%	93.6% ± 0.1%	95.9% ± 0.2%	94.95% ± 0.2%	95.1% ± 0.2%

Table 6.11: Final results for RF and RBFN for Bio-Tex and Bio-Ran dataset

there is comparatively a relatively stable path followed by each user of following the text. But in Bio-Ran, the path is very random and thus the position of an outlier. So probably, chances of false outliers being detected or outliers not getting detected may happen. So after adding outliers, there is a slight decrease in the accuracy. It can also be seen that RBFN performs better than RF. This is due to the factor that RBFN can generalize well for patterns not present in training.

The same weight calculation technique as mentioned in Section 6.6 has been used for the Bio-Ran dataset. The best weights for RF are 37 for saccade, 41 for fixation, and 22 for outliers. Similarly, for RBFN, the best weights are 50 for saccade, 44 for fixation, and 6 for outliers. The result for the original accuracy and the results using the new pipeline can be seen in Table 6.11,Figure 6.8.

Results were also calculated for cross-evaluation. That is training the classifier on Bio-Tex and testing on Bio-Ran, but this doesn't provide good results. The results received were in the range of 15-20%. This shows that the classifier training is very much task-specific, and thus classifier trained on one stimulus will not perform well on other stimuli.

So, the final pipeline for Bio-Tex and Bio-Ran is completely the same except for the weights. The pipeline with 3 classifiers is used for both. Outlier detection is done using clustering algorithm DBScan with values of *min_samples* is 3 and *eps* is 2. The percentile values for adding limits are 0.2 and 0.98. The best accuracies for Bio-Tex are obtained using RF with weights of 35,52,13 for saccade, fixation and outliers. The best accuracies for Bio-Ran are obtained for the new pipeline using RBFN with weights of 50,44,6 even though this accuracy is less than the original one.

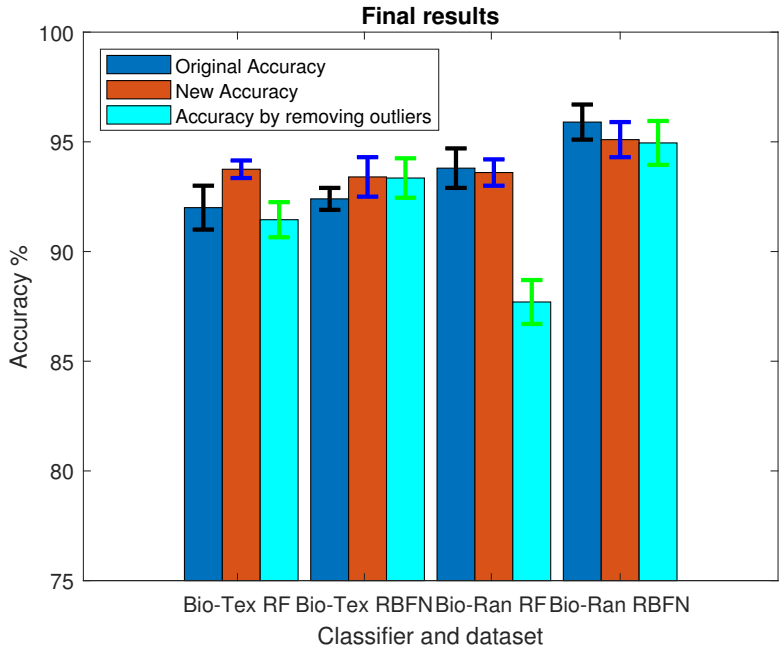


Figure 6.8: Comparision of final accuracies for Bio-Tex and Bio-Ran dataset for 20 seeds with original accuracies

7 Conclusion

In this thesis, the importance of outliers in eye moment trajectories is discussed. Along with fixations and saccades, a new classifier of outliers was added to the pre-existing pipeline. Outliers can be present outside the image area as well as inside the image area due to various reasons mentioned in this thesis. This new classifier is introduced to deal with outliers present outside the stimulus and also within the stimulus. A final comparison of accuracies is done for three different scenarios. The first one is the original pipeline, the second one is after removing the outliers, and the third one is after adding a new classifier into the pipeline. All the experiments were mainly performed on the Bio-Tex dataset, and then the final pipeline was tried on the Bio-Ran dataset also. As expected, a pipeline with three classifiers tends to improve the accuracy of the Bio-Tex dataset. Also, there is no conclusive result for the Bio-Ran dataset as the results after including a new classifier for outliers are approximately the same as the original accuracies. More experiments need to be performed to see the effects of outliers on the Bio-Ran dataset.

One of the objectives of the thesis is to detect outliers. It is very difficult to treat which points to be treated as outliers and which as fixation/saccade. An observation is that outliers outside the image area are mostly saccades, while outliers inside the image area have an equal ratio of fixation and saccades. Along with these outliers is random data that is it does not appear at a fixed interval or a fixed duration. The only thing is that the pattern of outliers that is the trajectory of outliers for a user is quite similar. Thus the inclusion of outliers into a separate classifier is beneficial. More research can be done on the pattern and occurrence of outliers in the data.

The next objective of the thesis was to remove the outliers from the pipeline and check the accuracy The results signify that removing the outliers does not improve the accuracy for RF for both of the datasets. For RBFN, the accuracy is improved for the Bio-Tex dataset but not for the Bio-Ran dataset. Therefore, it is not a good idea to work by removing the outliers.

Therefore the other way to deal with outliers is to include the outliers. This can be done by adding one classifier for all outliers or two classifiers, one for outlier fixation and the other for outlier saccades. One significant observation is that segmenting outliers into fixation and saccade leads to decreased accuracy. This is because of the small data size present after segmenting the outliers and the imbalance present in the data. Working with outliers into a single group is much more useful and gives better results.

A few of the points can be improved and worked on and are mentioned below. In

Bio-Tex, the eye movement is directed by the text. So there is a relatively stable and directed movement of the eye and here, adding outliers into the pipeline improves the results. In Bio-Ran, where the path is completely random, there is no improvement in accuracy. Experiments need to be performed on more datasets to see the performance of outliers on the data. This may help to understand more the pattern in which outliers occur. It can also help to understand does the outlier frequency remain the same for users irrespective of the stimuli.

One more approach which can be used is by classifying further the outliers group based on their reasons. As discussed before, blinking may occur at approximately the same interval and in a similar pattern. So if it is possible to classify outliers based on blinking, it will be helpful to classify users, and each user has a different blinking pattern. But currently, it is very difficult to separate blinking points from the data as there is no information about the blinking rate for the users. The new eye trackers usually compensate for most of the blinking points. One more approach that could be used is to classify outliers into outlier fixation and outlier saccade and then to use a classifier that can deal with small data size and highly imbalanced data. Or a method to appropriately balance the outlier data for all users can be used.

In the future, experiments may be tried to perform on more users and for more seeds. Due to hardware and time restrictions, it was not possible in this thesis. But increasing the users and runs will definitely give a clearer idea about the results, mainly for the Bio-Ran dataset where there are no conclusive results obtained.

Thus, it can be stated that outliers play a significant role in eye movement. The improvement in accuracy for one dataset proves that with more work and research on outliers, accuracy can be further increased. The uniqueness of outliers from user to user is an important observation, and it can be further explored for improving the results. More research in this direction would definitely be helpful in the field of user identification.

Bibliography

[1] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends in cognitive sciences*, vol. 4, no. 1, pp. 6–14, 2000.

[2] A. Bayat and M. Pomplun, "Biometric identification through eye-movement patterns," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2017, pp. 583–594.

[3] C. Schröder, S. M. K. Al Zaidawi, M. H. Prinzler, S. Maneth, and G. Zachmann, "Robustness of eye movement biometrics against varying stimuli and varying trajectory length," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–7.

[4] D. L. Silver and A. Biggs, "Keystroke and eye-tracking biometrics for user identification." in *IC-AI*. Citeseer, 2006, pp. 344–348.

[5] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to biometrics*. Springer Science & Business Media, 2013.

[6] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: a survey and classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[7] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of biometrics*. Springer Science & Business Media, 2007.

[8] R. S. Mohammed, N. J. Habeeb, and Z. M. Abood, "Iris matching using surf algorithm," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 12, pp. 91–102, 2016.

[9] A. Babich, "Biometric authentication. types of biometric identifiers," 2012, accessed on 2021-04-18. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/44684/Babich_Aleksandra.pdf

[10] "Ocular Motor Control (Section 3, Chapter 8) Neuroscience Online: An Electronic Textbook for the Neurosciences | Department of Neurobiology and Anatomy - The University of Texas Medical School at Houston," accessed on 2021-02-10. [Online]. Available: https://nba.uth.tmc.edu/neuroscience/m/s3/chapter08.html

[11] P. Kasprowski and J. Ober, "Eye movements in biometrics," in *International Workshop on Biometric Authentication*. Springer, 2004, pp. 248–258.

[12] A. George and A. Routray, "A score level fusion method for eye movement biometrics," *Pattern Recognition Letters*, vol. 82, pp. 207–215, 2016.

[13] M. A. Hossain and B. Assiri, "An enhanced eye-tracking approach using pipeline computation," *Arabian Journal for Science and Engineering*, pp. 1–14, 2020.

[14] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.

[15] A. Ahrens, *Studies on the movement of the eyes when writing*. Carl Boldt'sche Hof-Buchdruckerei, 1891.

[16] H. E. Blanchard, G. W. McConkie, D. Zola, and G. S. Wolverton, "Time course of visual information utilization during fixations in reading." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 10, no. 1, p. 75, 1984.

[17] D. Noton and L. Stark, "Scanpaths in eye movements during pattern perception," *Science*, vol. 171, no. 3968, pp. 308–311, 1971.

[18] S. Josephson and M. E. Holmes, "Visual attention to repeated internet images: testing the scanpath theory on the world wide web," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, 2002, pp. 43–49.

[19] G. Bargary, J. M. Bosten, P. T. Goodbourn, A. J. Lawrance-Owen, R. E. Hogg, and J. Mollon, "Individual differences in human eye movements: An oculomotor signature?" *Vision research*, vol. 141, pp. 157–169, 2017.

[20] S. Makowski, L. A. Jäger, A. Abdelwahab, N. Landwehr, and T. Scheffer, "A discriminative model for identifying readers and assessing text comprehension from eye movements," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 209–225.

[21] L. A. Jäger, S. Makowski, P. Prasse, S. Liehr, M. Seidler, and T. Scheffer, "Deep eyedentification: Biometric identification using micro-movements of the eye," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 299–314.

[22] H.-k. Ko, D. M. Snodderly, and M. Poletti, "Eye movements between saccades: Measuring ocular drift and tremor," *Vision research*, vol. 122, pp. 93–104, 2016.

[23] M. Nyström, D. W. Hansen, R. Andersson, and I. Hooge, "Why have microsaccades become larger? investigating eye deformations and detection algorithms," *Vision research*, vol. 118, pp. 17–24, 2016.

[24] J. Otero-Millan, X. G. Troncoso, S. L. Macknik, I. Serrano-Pedraza, and S. Martinez-Conde, "Saccades and microsaccades during visual fixation, exploration, and search: foundations for a common saccadic generator," *Journal of vision*, vol. 8, no. 14, pp. 21–21, 2008.

[25] I. Rigas and O. V. Komogortsev, "Current research in eye movement biometrics: An analysis based on bioeye 2015 competition," *Image and Vision Computing*, vol. 58, pp. 129–141, 2017.

[26] R. Bednarik, T. Kinnunen, A. Mihaila, and P. Fränti, "Eye-movements as a biometric," in *Scandinavian conference on image analysis*. Springer, 2005, pp. 780–789.

[27] T. Kinnunen, F. Sedlak, and R. Bednarik, "Towards task-independent person authentication using eye movement signals," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, pp. 187–190.

[28] O. V. Komogortsev, S. Jayarathna, C. R. Aragon, and M. Mahmoud, "Biometric identification via an oculomotor plant mathematical model," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, pp. 57–60.

[29] O. V. Komogortsev, A. Karpov, L. R. Price, and C. Aragon, "Biometric authentication via oculomotor plant characteristics," in *2012 5th IAPR International Conference on Biometrics (ICB)*. IEEE, 2012, pp. 413–420.

[30] C. Holland and O. V. Komogortsev, "Biometric identification via eye movement scanpaths in reading," in *2011 International joint conference on biometrics (IJCB)*. IEEE, 2011, pp. 1–8.

[31] C. D. Holland and O. V. Komogortsev, "Complex eye movement pattern biometrics: the effects of environment and stimulus," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2115–2126, 2013.

[32] N. V. Cuong, V. Dinh, and L. S. T. Ho, "Mel-frequency cepstral coefficients for eye movement identification," in *2012 ieee 24th international conference on tools with artificial intelligence*, vol. 1. IEEE, 2012, pp. 253–260.

[33] I. Rigas, G. Economou, and S. Fotopoulos, "Biometric identification based on the eye movements and graph matching techniques," *Pattern Recognition Letters*, vol. 33, no. 6, pp. 786–792, 2012.

[34] ——, "Human eye movements as a trait for biometrical identification," in *2012 IEEE fifth international conference on biometrics: theory, applications and systems (BTAS)*. IEEE, 2012, pp. 217–222.

[35] Z. Liang, F. Tan, and Z. Chi, "Video-based biometric identification using eye tracking technique," in *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*. IEEE, 2012, pp. 728–733.

[36] Y. Zhang and M. Juhola, "On biometric verification of a user by means of eye movement data mining," in *Proceedings of the 2nd international conference on advances in information mining and management*, 2012, pp. 85–90.

[37] O. V. Komogortsev and C. D. Holland, "Biometric authentication via complex oculomotor behavior," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2013, pp. 1–8.

[38] I. Rigas and O. V. Komogortsev, "Biometric recognition via probabilistic spatial projection of eye movement trajectories in dynamic visual environments," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1743–1754, 2014.

[39] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," 2012.

[40] V. Cantoni, C. Galdi, M. Nappi, M. Porta, and D. Riccio, "Gant: Gaze analysis technique for human identification," *Pattern Recognition*, vol. 48, no. 4, pp. 1027–1038, 2015.

[41] I. Rigas, E. Abdulin, and O. Komogortsev, "Towards a multi-source fusion approach for eye movement-driven recognition," *Information Fusion*, vol. 32, pp. 13–25, 2016.

[42] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, "An overview on trajectory outlier detection," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2437–2456, 2019.

[43] V. A. Tran, O. Hirose, T. Saethang, L. A. T. Nguyen, X. T. Dang, T. K. T. Le, D. L. Ngo, G. Sergey, M. Kubo, Y. Yamada *et al.*, "D-impact: A data preprocessing algorithm to improve the performance of clustering," *Journal of Software Engineering and Applications*, vol. 2014, 2014.

[44] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107 964–108 000, 2019.

[45] V. Ilango, R. Subramanian, and V. Vasudevan, "A five step procedure for outlier analysis in data mining," *European Journal of Scientific Research*, vol. 75, no. 3, pp. 327–339, 2012.

[46] E. Achtert, H.-P. Kriegel, L. Reichert, E. Schubert, R. Wojdanowski, and A. Zimek, "Visual evaluation of outlier detection models," in *International Conference on Database Systems for Advanced Applications*. Springer, 2010, pp. 396–399.

[47] S. Seo, "A review and comparison of methods for detecting outliers in univariate data sets," Ph.D. dissertation, University of Pittsburgh, 2006.

[48] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," *Computer Networks*, vol. 129, pp. 319–333, 2017.

[49] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

[50] H.-P. Kriegel, P. Kröger, and A. Zimek, "Outlier detection techniques," *Tutorial at KDD*, vol. 10, pp. 1–76, 2010.

[51] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[52] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[53] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1649–1652.

[54] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based gmm," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 145–154.

[55] J. Zhang, "Advancements of outlier detection: A survey," *ICST Transactions on Scalable Information Systems*, vol. 13, no. 1, pp. 1–26, 2013.

[56] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 61–75.

[57] T. T. Dang, H. Y. Ngan, and W. Liu, "Distance-based k-nearest neighbors outlier detection method in large-scale traffic data," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2015, pp. 507–510.

[58] H. Huang, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 3, pp. 518–531, 2013.

[59] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 2, pp. 145–160, 2005.

[60] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[61] R. T. Ng and J. Han, "E cient and e ective clustering methods for spatial data mining," in *Proceedings of VLDB*. Citeseer, 1994, pp. 144–155.

[62] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[63] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 68–86, 1971.

[64] G. Karypis, E. Han, and V. Kumar, "A hierarchical clustering algorithm using dynamic modeling," 1999.

[65] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[66] A. Hinneburg, D. A. Keim *et al.*, "An efficient approach to clustering in large multimedia databases with noise," in *KDD*, vol. 98, 1998, pp. 58–65.

[67] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[68] P. Kasprowski and J. Ober, "Enhancing eye-movement-based biometric identification method by using voting classifiers," in *Biometric Technology for Human Identification II*, vol. 5779. International Society for Optics and Photonics, 2005, pp. 314–323.

[69] B. V. Ehinger, K. Groß, I. Ibs, and P. König, "A new comprehensive eye-tracking test battery concurrently evaluating the pupil labs glasses and the eyelink 1000," *PeerJ*, vol. 7, p. e7086, 2019.

[70] "Eyelink 1000 user manual," accessed on 2021-03-10. [Online]. Available: http://sr-research.jp/support/EyeLink%201000%20User%20Manual%201.5.0.pdf

[71] E. Dalmaijer, "Is the low-cost eyetribe eye tracker any good for research?" PeerJ PrePrints, Tech. Rep., 2014.

[72] D. J. Graham and R. W. Jeffery, "Location, location, location: eye-tracking evidence that consumers preferentially view prominently positioned nutrition information," *Journal of the American Dietetic Association*, vol. 111, no. 11, pp. 1704–1711, 2011.

[73] G. E. Raney, S. J. Campbell, and J. C. Bovee, "Using eye movements to evaluate the cognitive processes involved in text comprehension," *Journal of visualized experiments: JoVE*, no. 83, 2014.

[74] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[75] T. Sen and T. Megaw, "The effects of task variables and prolonged performance on saccadic eye movement parameters," in *Advances in Psychology*. Elsevier, 1984, vol. 22, pp. 103–111.

[76] D. D. Salvucci, "Interpreting eye movements with process models," in *CHI 98 Conference Summary on Human Factors in Computing Systems*, 1998, pp. 66–67.

[77] D. Sauter, B. Martin, N. Di Renzo, and C. Vomscheid, "Analysis of eye tracking movements using innovations generated by a kalman filter," *Medical and biological Engineering and Computing*, vol. 29, no. 1, pp. 63–69, 1991.

[78] L. Stark, "Scanpaths revisited: Cognitive models, direct active looking," *Eye movements: Cognition and visual perception*, pp. 193–226, 1981.

[79] J. H. Goldberg and J. C. Schryver, "Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection," *Behavior research methods, instruments, & computers*, vol. 27, no. 3, pp. 338–350, 1995.

[80] M. R. Harwood and J. P. Herman, "Optimally straight and optimally curved saccades," *Journal of Neuroscience*, vol. 28, no. 30, pp. 7455–7457, 2008.

[81] L. Breiman, "Random forests," *UC Berkeley TR567*, 1999.

[82] M. J. Orr *et al.*, "Introduction to radial basis function networks," 1996.

[83] "Radial Basis Function Networks," May 2019, accessed on 2021-13-04. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/radial-basis-function-network

[84] J. Joyce, "Bayes' theorem," 2003.

[85] D. Berrar, "Bayes' theorem and naive bayes classifier," *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands*, pp. 403–412, 2018.

[86] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[87] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[88] "SVM | Support Vector Machine Algorithm in Machine Learning," accessed on 2021-04-01. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[89] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.

[90] D. Ghosh and A. Vogt, "Outliers: An evaluation of methodologies," in *Joint statistical meetings*, vol. 2012, 2012.

[91] A. J. Hornof and T. Halverson, "Cleaning up systematic error in eye-tracking data by using required fixation locations," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 592–604, 2002.

[92] A. Aaltonen, A. Hyrskykari, and K.-J. Räihä, "101 spots, or how do users read menus?" in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1998, pp. 132–139.

[93] J. M. Findlay, D. Brogan, and M. G. Wenban-Smith, "The spatial signal for saccadic eye movements emphasizes visual boundaries," *Perception & Psychophysics*, vol. 53, no. 6, pp. 633–641, 1993.

[94] B. A. Smith, J. Ho, W. Ark, and S. Zhai, "Hand eye coordination patterns in target selection," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, 2000, pp. 117–122.

[95] A. Pollatsek, K. Rayner, and J. M. Henderson, "Role of spatial location in integration of pictorial information across saccades." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 16, no. 1, p. 199, 1990.

[96] K. Mukuno, S. Aoki, S. Ishikawa, S. Tachibana, H. Harada, G. Hozumi, and E. Saito, "Three types of blink reflex evoked by supraorbital nerve, light flash and corneal stimulations." *Japanese journal of ophthalmology*, vol. 27, no. 1, pp. 261–270, 1983.

[97] L. A. Riggs, J. P. Kelly, K. A. Manning, and R. K. Moore, "Blink-related eye movements." *Investigative ophthalmology & visual science*, vol. 28, no. 2, pp. 334–342, 1987.

[98] A. Ruetsche, A. Baumann, X. Jiang, and D. S. Mojon, "Automated analysis of eye tracking movements," *Ophthalmologica*, vol. 217, no. 5, pp. 320–324, 2003.

[99] R. S. Hessels and I. T. Hooge, "Eye tracking in developmental cognitive neuroscience–the good, the bad and the ugly," *Developmental cognitive neuroscience*, vol. 40, p. 100710, 2019.

[100] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[101] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[102] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[103] M. Breuig, H. Kriegel, R. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings ACM SIGMOD Conference, New York*, 2000, pp. 427–438.

[104] J. Miller, "Reaction time analysis with outlier exclusion: Bias varies with sample size," *The quarterly journal of experimental psychology*, vol. 43, no. 4, pp. 907–912, 1991.

[105] D. Howell, M. Rogier, V. Yzerbyt, and Y. Bestgen, "Statistical methods in human sciences," *New York: Wadsworth*, vol. 721, 1998.

[106] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of experimental social psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[107] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the american statistical association*, vol. 69, no. 346, pp. 383–393, 1974.

[108] P. J. Huber, *Robust statistics*. John Wiley & Sons, 2004, vol. 523.

[109] H. Konno and T. Koshizuka, "Mean-absolute deviation model," *Iie Transactions*, vol. 37, no. 10, pp. 893–900, 2005.

[110] J. W. Tukey *et al.*, *Exploratory data analysis*. Reading, Mass., 1977, vol. 2.

[111] M. Hubert and E. Vandervieren, "An adjusted boxplot for skewed distributions," *Computational statistics & data analysis*, vol. 52, no. 12, pp. 5186–5201, 2008.

[112] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, *Understanding robust and exploratory data analysis*, 2000, no. Sirsi) i9780471384915.

[113] G. Barbato, E. Barini, G. Genta, and R. Levi, "Features and performance of some outlier detection methods," *Journal of Applied Statistics*, vol. 38, no. 10, pp. 2133–2149, 2011.

[114] "scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation," accessed on 2021-03-18. [Online]. Available: https://scikit-learn.org/stable/

[115] "sklearn.cluster.DBSCAN — scikit-learn 0.24.1 documentation," accessed on 2021-04-18. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN

[116] A. C. Acock, "Working with missing values," *Journal of Marriage and family*, vol. 67, no. 4, pp. 1012–1028, 2005.

[117] R. Dangl and F. Leisch, "Effects of resampling in determining the number of clusters in a data set," *Journal of Classification*, pp. 1–26, 2019.

[118] T. Kanamori, S. Fujiwara, and A. Takeda, "Breakdown point of robust support vector machine," *arXiv preprint arXiv:1409.0934*, 2014.

[119] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Semisupervised svm batch mode active learning with applications to image retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 3, pp. 1–29, 2009.

[120] M. Martinez-Arroyo and L. E. Sucar, "Learning an optimal naive bayes classifier," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 1236–1239.