ISLAMIC UNIVERSITY OF TECHNOLOGY

Development of a Prediction System To Enhance Fishing Activities

By

OUMAROU MOHAMADOU (191041040)

A Project Report submitted in partial fulfilment of the requirements for the degree of Master of Engineering in Computer Science and Engineering

Academic Year: 2019-2020

Department of Computer Science and Engineering

Islamic University of Technology.

A Subsidiary Organ of the Organization of Islamic Cooperation.

Dhaka, Bangladesh.

June 2022

Declaration of Authorship

I, Oumarou Mohamadou, declare that this thesis titled, 'Development of a Prediction System to Enhance Fishing Activities' and the work presented in it is my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Any part of this thesis has not been submitted for any other degree or qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

(Signature of the Candidate)
Oumarou Mohamadou- 191041040
June 2022

Development of a Prediction System to Enhance Fishing Activities

Approved By:

Dr. Md. Azam Hossain
Thesis Supervisor,
Assistant Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Md. Abu Raihan Mostofa Kamal
Head of the Department and Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Md Moniruzzaman
Assistant Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Mohammad Rezwanul Huq
Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh.

Abstract

The support to enhance the performance of fishing activities against the struggle to fight effectively the illegal fishing is a paramount to happy and successful life between fishermen and the government. Therefore, this project was aimed at designing and developing prediction application to ease the fishing activities works of the concerned people. Our project aims is the tracking, monitoring and prediction of fishing activities to prevent different fraudulent actions by exploring and interpretation the collected fishing data. Requirement gathering was achieved through the existing literature [Articles/Journals/research works] based on pre-established guiding questions designed by the project team members in order to attain those specific user problems that needed to be addressed. The analysis process was implemented using machine learning algorithms for prediction. The report of this project talked about the existing problem and proposed architecture of different research. We, then analyze their usability on tackling the issues that we are facing.

Keyword - NoSQL; IoT; Fishing Activities; Linear regression, Logarithmic Regression; Machine Learning

Acknowledgements

I would like to express my whole-hearted gratitude to Allah Subhanu Wata'ala for giving me strength to complete this study, and being with me when none was there beside me. I would also like to express my grateful appreciation to Dr. Md. Kamrul Hasan and Dr. Hasan Mahmud for their constant motivation and support throughout this study.

This work is partially supported by

Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Contents

List of Figures

List of Tables

Dedicated Specially to my late Father Prof. Dr. OUMAROU BOUBA (May the Almighty Allah forgives your shortcomings and grant you the high level of paradise. Ameen), to my lovely Mother and my amazing siblings for their lifelong dedicated support to my education ...

Chapter 1

Introduction

1.1 Overview

Chapter one of this project report entails the background to the problem, motivation and application domain, then the problem statement, goal of the study, general and specific objectives, project scope, significance of the study and finally proposal organization.

1.2 Brief Background of the Problem

Data collection and Analysis are critical components in the management of vital marine fish populations. Fishery-dependent data are critical for developing accurate fishery assessments and assessing the efficacy of state and federal fishery management policies. These data involve identifying the current or potential way of monitoring the fishes that have been caught by fishermen while keeping track of each of them. However, there are some conflicting views regarding on how best to implement a sustainable marine resources management policy for the better control of fishing activities. For instance, illegal fishing is a worldwide issue even though the legal fishing all alone is not capable of taking care of everything.

1.3 Motivation and Application Domain

Information and Communication Technology (ICT) has been used for a long time to diagnose, assess and bring solution to many kinds of daily life problems specially by the used of IoT devices and Cloud Technologies. For instance, the Fisheries Monitoring Project for assessing and monitoring fishes' life. Clearly, ICT Solutions have significantly greater impact on solving problems requiring specific needs. Since the government is fighting for Illegal fishing and that people are coming always with ideas to escape the surveillance of the government which makes it difficult to control. Therefore, the used of IoT devices and cloud technologies will definitely be suitable for the government to fight the illegal fishing IUU (Illegal fishing, Unreported fishing, Unregulated fishing).

1.4 Problem Statement

The amount of fishes caught by fishermen are increasing day by the day as well as the new technologies that are introduce. The data collected is without any doubt an asset because previously it is quite a big challenge to get information without using any data analysis method or algorithms. The conditions that have been developed including predictive algorithm or exploratory analytics algorithms for analysis of fishing for the purpose of stopping Illegal activities are now flexible in training session. The aforementioned studies suggest that, ICT is an effective approach to help the government with this issue. Therefore, this project aims at identifying and evaluating a dynamic way of fishing tracking activities best approaches and strategies for combatting Illegal fishing IUU (Illegal fishing, Unreported fishing, Unregulated fishing).

1.5 The goal of the study

The main goal of the project work is to track the fishing activities through the elaboration of strategies while monitoring the fishing zones[1]

1.6 General Objective

The general objective is to develop an application capable of analyzing the data that have been collecting through the IoT devices, then visualize those that onto the application's dashboard with different charts models.

1.7 Specific Objective

The specific objective of the study is to:

- Predict the amount of fishes that will be caught in a particular zone
- Compare, describe and rate each of the collected data using different charts models
- Present report to the user for decision making

1.8 The Scope of the Project

The application is web based and this project will define a use case model, collect the required data, develop the model of analyzing, then we execute the machine learning algorithm to provide the prediction on the fishing activities. Operationally, the new system was tested using fake data. It was basically designed to focus on exploring and interpreting the data that is to predict what is expected next time.

1.9 UI/UX Considerations

Input to the system was monitored and analyzed on a real-time basis, allowing for the application to dynamically change the user experience to optimize learning effectively information that the management team will required to take decisions.

1.10 Significance of the Study

The study was carried out with full consideration of the users and therefore, entirely helped to solve their core problems.

1.10.1 To the User

- Visualizing the collected data
- Getting to estimate what will happen in the next months

1.10.2 To the Researcher

- Implement skills acquired in class and gain more experience in the field of solving different or related problems.
- The group project helps us develop a sense of teamwork with others through interactive discussions.

1.11 Intended Audience

The target audience for this document includes project management stakeholders and system developers to help guide them during implementation and testing for evaluation accountability.

1.12 Project Outline

The document consists of six Chapters. Chapter 1 contains the background to the study, problem statement, general and specific objectives, scope and significance of the study. Chapter 2 gives a literature review on related projects done by previous researchers. Chapter 3 is about system architecture, system analysis use case, activity, sequence diagrams and requirement determination, sample interview question guides, Data flow diagram. Chapter 4 contains the system design and implementation that is, research findings, system inputs and outputs and so on. Chapter 5 entails testing and validation, tools and technologies used in system development. Chapter 7 draws a conclusion to the current study and discusses future directions. The final segment of this study contains all the references and credits used.

Chapter 2

Literature review

The purpose of this section is to provide a critical review of other previous research works been done which are related to this current one at hand. The previous studies will be analyzed and subjected to critics in order to justify the need for a new system depending on certain theories.

Fishing activities is a human-based activity that consist of catching fishes from the natural environment but it might also be caught from stocked bodies of water such as ponds or canals. Nowadays, tracking and monitoring of fishing activities is quite difficult since some issues and challenges are there. Studies have been conducted to improve predictions of fishing activities all around the world. The challenges generally fall into three categories:

- Communication issues
- Lack of Automation System
- Unavailability of data analysis in the domain

The current study is not exceptional in considering the above fishing activities challenges in the requirement and/or data gathering process. Many projects have been conducted to improve the monitoring, tracking and prediction of fishing activities. According to a study conducted by Buggey [2], (2005), videotaped self-modeling program was developed to allow participants to view themselves in a form of training while performing in a more advanced level than they typically function. The results indicated that participants exhibited immediate and significant gains and that those gains were maintained after cessation of treatment.

Advantage/Strength

• Improves fishermen's interest to learn

• Improves learning through interaction by responding to actions performed by the robot

• Improves attention towards learning what the robot is doing

Disadvantage/Weakness

• Where the robot does not move, the user's learning process is distracted

• This kind of training may lead to distraction to the real world since the trainer is entirely a machine.

According to University of Bristol, [3] (2019), with fishing activities Condition it is harder to recognize fishing zone without some equipment. However, the types of mistake made by young people from one region with high experience fishermen were very similar to the types of mistake made by young people without Experience.

Chapter 3

Proposed Approach

The procedure used to assess the severity of depression in this study is based on a well-established clinical assessment method known as the Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5) [4], and it was carried out under the supervision of two expert clinical psychologists. Twitter was chosen as the data collection platform for ease of accessibility of data and avalaibility of APIs.

3.1 Overview

This chapter presents the methods, data collection technique and system development strategy applied to attain the project objectives. These methods are the five phases of system development life cycle as follows: - Planning, system analysis and requirement gathering that has helped in developing Use Case scenarios, Designing and Implementation, Testing and Validating the new system and finally Maintaining it.

3.2 System Development Methodology

The methodology used to develop the new system is Agile. It's a method of project management characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans. Agile is a time boxed iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once.

3.3 Why Agile Method for the development of system

1. Revenue: The iterative nature of agile development means features are delivered incrementally, enabling some benefits to be realized early as the product continues to develop.

2. Speed-to-market: As well as the higher revenue from incremental delivery, agile development philosophy also supports the notion of early and regular releases, and 'perpetual beta'.

3. Quality: A key principle of agile development is that testing is integrated throughout the lifecycle, enabling regular inspection of the working product as it develops. This allows the product owner to make adjustments if necessary and gives the product team early sight of any quality issues.

4. Visibility: Agile development principles encourage active user involvement throughout the product's development and a very cooperative collaborative approach. This provides excellent visibility for key stakeholders, both of the project's progress and of the product itself, which in turn helps to ensure that expectations are effectively managed.

5. Risk Management: Small incremental releases made visible to the product owner and product team through its development help to identify any issues early and make it easier to respond to change.

6. Flexibility / Agility: We resist changes and put people through a change control committee to keep them to the essential minimum. Agile development principles are different. In agile development, change is accepted.

7. Cost Control: The above approach of fixed timescales and evolving requirements enable a fixed budget. The scope of the product and its features are variable, rather than the cost.

8. User Satisfaction: The active involvement of a user representative and/or product owner, the high visibility of the product and progress and the flexibility when change is needed, creates much better engagement and user satisfaction.

9. Right Product: Above all other points, the ability for agile development requirements to emerge and evolve, and the ability to embrace change (with the appropriate trade-offs), the team builds the right product.

10. More Enjoyable: The active involvement, cooperation and collaboration makes agile development teams a much more enjoyable place for most people.

3.4 Requirement Gathering Process

[5] There are basically two data sources for requirement gathering; primary and secondary as briefly explained below.

3.4.1 Primary Data

Primary data refers to data that is collected by a researcher from first-hand sources, using methods like surveys, interviews, or experiments. It is collected with the research project in mind, directly from primary sources.

3.4.2 Secondary Data

Secondary data is data gathered from studies, surveys, or experiments that have been run by other people or for other research. The data source for the current project was Secondary where we used the existing literature as a technique or a tool to obtain subjects' requirements by accessing published papers and other relevant web resources. These requirements were thoroughly analyzed in order to provide information that was able to justify the need for the new system.

However, the researchers formulated some questions as a guide in search of the specific user needs/requirements as they went through related articles [journals]. The purpose of these questions was to help the researchers to understand communication issues among the people as well as how to improve the activities of the fisheries industry that have not yet been answered.

3.5 Dataset Preparation

In this phase of system development [2], in order to build our system, we need to acquire a large set of diverse datasets to present the insights of the collected data. Additionally, we must pre-process our dataset to highlight attribute names, aggregation functions, and complexity through data exploration and Validation, then we clean the data before processing to the visualization part. It consists of the following tasks:

- Data Exploration
- Data validation
- Data Cleaning

3.6 Prediction of Optimal Strategies

3.6.1 Outline of the Prediction

We should first develop a model that is capable of presenting the included dataset via its attribute names and underlying values, as well as analyzing and deriving any derived data from the charts model. We then need to construct a model that will generate visualizations based on the prior model's extracted features. After we will design a framework that tells us what is happening with the collected data.

In this step, we need to create a new model to generate an explanation for the future fish to be caught in a particular zone before the descriptive analysis framework, taking into account the sensitive significance of features for analytical tasks, attributes, and visualizations on both a local and global level. To do that, we will first train our model by using 80%of the data and the remaining 20% is for testing. The next step is to repeat the process several times before going for interpretation and final testing.

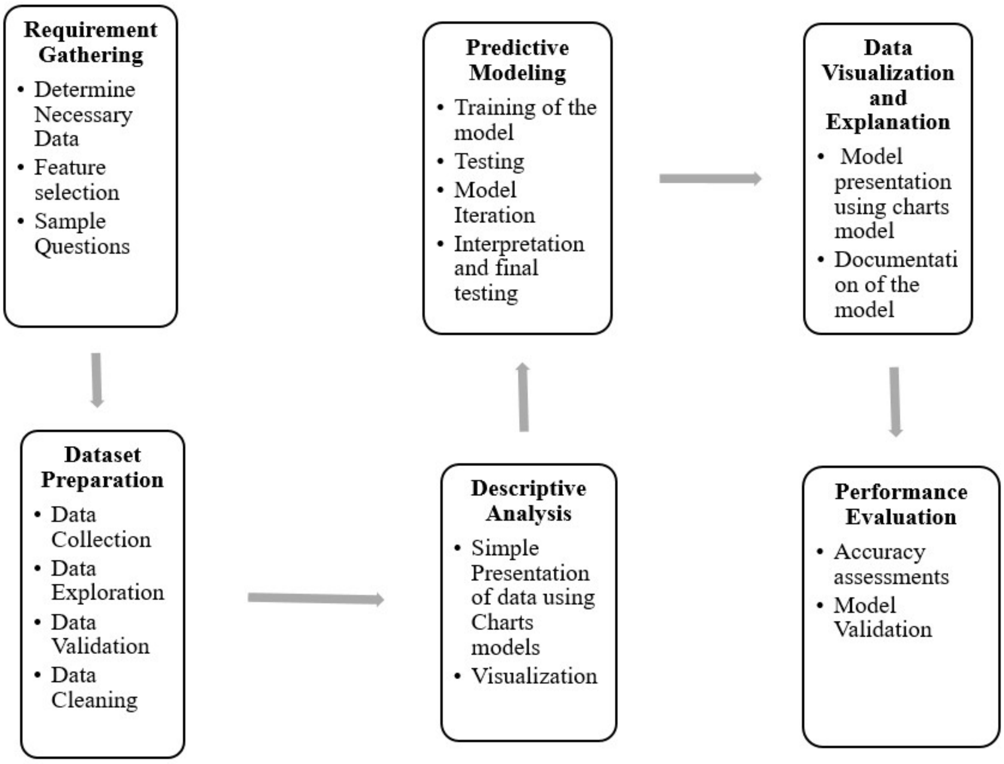

FIGURE 3.1: Outline of our proposed work's methodology

3.6.2 Linear Regression Algorithm Prediction Method

3.7 Data Visualization and Explanation

Once the aforementioned models are operational, we will combine them to create an explainable fish activities framework that conforms to our desired model. The final model's visualizations and explanations will be fine-tuned using a variety of model validation procedures.

To assess the proposed model's performance, additional analysis [2] may be conducted by calculating the generalization error of feature extractions for the prior models using diverse datasets and comparing them to their counterparts. Additionally, the entire framework should be assessed by humans to ascertain their level of satisfaction.

To summarize the methodology process, we presented all the required steps to be followed in order to have the expected outcome.

3.8 System requirements

The system under development needs to perform certain specific activities identified to help meet the user needs discussed above.

3.8.1 Activities the system need to perform

- System receives input from fishermen
- Processing of the collected data
- Display those data into charts models
- System analyzes the data
- System predict the expected information

Basing on the information gathered, the system's functional and non-functional requirements were as follows:

3.8.2 Functional System Requirements

- System shall take input from the user
- System shall be able to do prediction
- System shall compare points of features
- System shall analyze data
- System displays an explanation of the analyzed data

3.8.3 Non-Functional System Requirements

- System errors shall be handled appropriately to avoid system crash so as to enhance data security.
- Provision of passwords as security measure to authenticate users to access information.

Chapter 4

System Design

Predicting the number of fishes that will be caught in a particular zone next month using linear regression algorithm follow this step:

1. Dependent variable is Y which is the Total fish caught
2. Independent variable is X which is the number of boats that have been used
3. Formula of the linear regression equation is: Y =a + bX

4. We used 80 % of the data to train the model and 20% for testing

The proportion of classes shown in Figure 4.2 indicates that the *non-depressed* samples outnumber the other classes by a wide margin. Though all the data samples are scraped based on the keywords related to different severity levels of

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

FIGURE 4.1: Equation of the Linear Regression Model

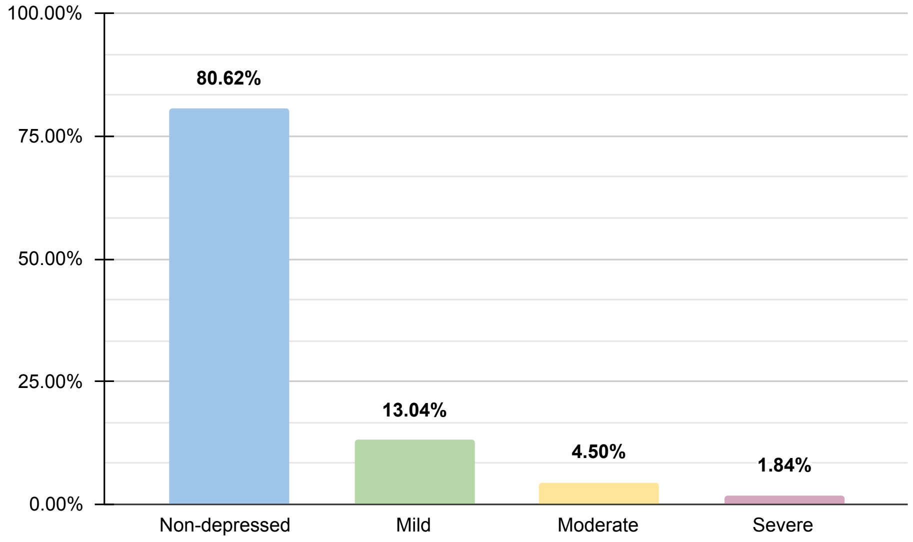

Figure 4.2: Percentage of Data Samples for Each Class

depression and the control samples were removed prior to the final preparation of the dataset, the number of data samples for different severities of depression is inevitably low. This class imbalance represents an important characteristic in the identification of various depressive disorders on social media. To discover this, manual analysis was done in two stages of this study: (i) while randomly choosing data samples for annotation, and (ii) during the initial iterations of the annotation job. The analysis indicated that the final class proportions roughly represent the percentage of similar attributes in similar live contexts.

Generally, the overall positive content shared in social media outnumbers the negative content. This is because people usually show their positive, friendly side over social media and tend to talk less about their struggles [6]. To mitigate this problem, previous studies depended on self-labeled data for collating large and balanced datasets on different mental disorders [7, 8]. However, depending only on self-labeled data to understand mental health from personal levels and measure the severity of the condition is not feasible without the intervention from expert psychologists. But considering the lack of resources in the mental health sector, only relying on psychologists can be time-consuming and expensive. As a result, in this study, crowdsourcing supervised by psychologists was opted to obtain high-quality data on different depression severities.

Despite the measures undertaken to ensure the quality of the dataset, the method of annotation warrants a certain level of noise in the dataset. This results in

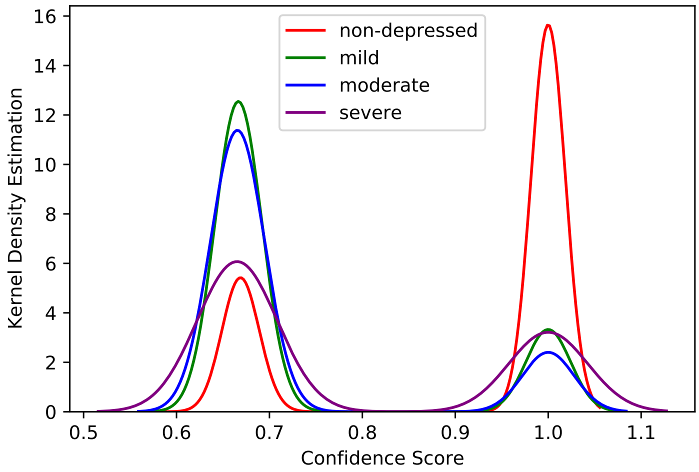

FIGURE 4.3: Kernel Density Estimation of Confidence Scores for Each Class

different yet rational interpretations of the same tweet. The kernel density estimation of the confidence scores portrayed in Figure 4.3 indicates that there is reasonable agreement among the annotators on deciding the class label of the *non-depressed* and *severe* classes. While these two classes lies on two different polarities of attributes, the subtle nuances of the *mild* and *moderate* classes allowed for rational disagreement among the annotators, which is evident from the high concentration of probability density for *mild* and *moderate* classes between 0.6 and 0.7 in Figure 4.3. This may be attributed not only to the lack of apprehension or awareness of the annotator, but also on the subjectivity of the topic at hand. It highlights the difficulty of using typical reliability metrics such as Inter-Rater Reliability (IRR), which calculates the level of agreement between two or more annotators. More sophisticated metrics like Fleiss' Kappa [9] can be applied in this scenario since the sample tweets were distributed randomly among the annotators and each annotator chose from one of the four mutually exclusive labels to indicate the severity of depression per tweet [10, 11]. However, Fleiss' Kappa assumes that the disagreement among the annotators on the same sample reduces the reliability of the dataset. Considering the subjective nature of the severity of depression detected by different annotators, that might not be the case [12]. In spite of that, Fleiss' Kappa is calculated to get an understanding of the overall agreement of the annotators in this study. The value of Fleiss' Kappa ranges from -1 (indicating no observed agreement) to +1 (indicating a perfect agreement) [11]. Here, a value less than 0.20 indicates a poor agreement, 0.21 to 0.40 indicates a fair agreement, 0.41 to 0.60 indicates moderate agreement, 0.61 to 0.80 indicates substantial agreement and 0.81 to 1 indicates a near perfect agreement among the annotators.

TABLE 4.1: Fleiss' Kappa per Class

Class	**Fleiss' Kappa**
Non-depressed	0.44
Mild	0.27
Moderate	0.30
Severe	0.45
Overall	0.36

As reported in Table 4.1, the Fleiss' Kappa for the *non-depressed* and *severe* classes show a moderate agreement among the annotators. This can be explained considering the extreme nature of these two classes as they tend to be the polar opposite of each other. On the other hand, a fair agreement in *mild* and *moderate* classes highlight the intricate relationship among these two classes and the difficulty in identifying the subtle cues to differentiate them, even for the humans. However, despite the subjective nature of the severity of depression, an overall fair agreement provides indication of the quality of the annotation, and the dataset in general.

4.1 Visualizing the Dataset

To visualize and discover the hidden themes in the four classes of the dataset, some unsupervised topic modeling techniques are applied. Topic modeling identifies topics present in a text object and to derive hidden patterns exhibited by a text corpus. The following sections contain various techniques and visualizations to represent topics and themes of the classes to understand underlying pattern of the linguistic use.

4.1.1 Wordcloud

A word cloud is a simple visual representation object for text processing, which shows the most frequent word with bigger and bolder letters, and with different colors. The smaller the the size of the word the lesser it's important. figurename 4.8 represents the wordclouds for four classes. The wordclouds might look similar, but the difference of the words are in the context of their usages. *Non-depressed* class can contain terms like 'tired', 'exhausted', 'depressed', etc. but they might casually indicate to a temporary tiredness or exhaustion because of a work, not a permanent trait of the users' daily lives. The Non-depressed wordcloud containing 'work', 'love', 'today', etc. confirm this assumption. On the other hand, frequent words like 'suicide', 'hate', 'self-destruction', etc. in the wordcloud of the *severe* class provide an idea of the calamitous mental state of users potentially suffering from severe depression.

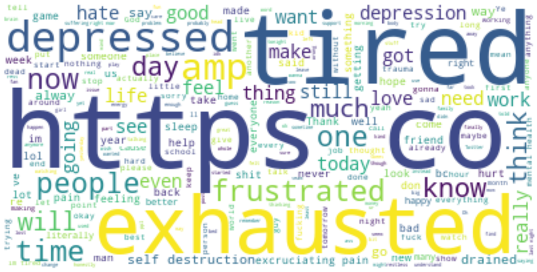

FIGURE 4.4: *Non-depressed* Word-cloud

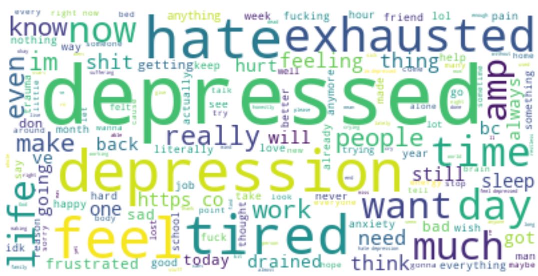

FIGURE 4.5: *Mild* Wordcloud

FIGURE 4.6: *Moderate* Wordcloud

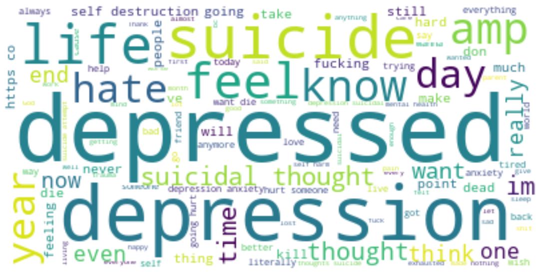

FIGURE 4.7: *Severe* Wordcloud

FIGURE 4.8: Wordclouds of Different Classes

4.1.2 Topic Modeling with Local Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a matrix factorization technique and one of the most popular topic modeling methods used by data analytics. LDA produces a generative probabilistic model that assumes each topic is a mixture over an

underlying set of words, and each document is a mixture of over a set of topic probabilities.

Table 4.2 shows top 10 words in the top 5 topics of all the classes. Along with the words, the probability distribution of the words over that topic is provided. This provides great insight into the dataset classes. The top topic of the *Non-depressed* class contains words like 'depress','frustrate', etc. which might look similar to the other classes. But term like 'http' says this might be just a song/entertainment source provided by the users on their feed with captions containing depressive words. Topics in other classes, specially in 'Severe' and 'Moderate' classes lay out

TABLE 4.2: Prevalent Topics of 4 Classes Discovered by LDA

Depression Levels	**Non-Depressed**	**Mildly Depressed**	**Moderately Depressed**	**Severely Depressed**
Topic 1	0.042*like+ 0.035*depress+ 0.033*http+ 0.025*feel+ 0.024*look+ 0.016*hate+ 0.015*say+ 0.015*love+ 0.010*frustrate+ 0.010*tire	0.035*tire+ 0.034*life+ 0.030*like+ 0.025*drain+ 0.023*mental+ 0.017*physic+ 0.016*feel+ 0.015*work+ 0.014*sick+ 0.013*know	0.035*know+ 0.031*go+ 0.030*feel+ 0.030*hurt+ 0.028*want+ 0.023*time+ 0.020*sick+ 0.020*suffer+ 0.019*fuck+ 0.018*month	0.137*feel+ 0.090*suicide+ 0.062*like+ 0.054*want+ 0.049*know+ 0.029*thing+ 0.027*thought+ 0.021*lose+ 0.020*live+ 0.020*anymore
Topic 2	0.034*depress+ 0.027*fuck+ 0.024*exhaust+ 0.022*http+ 0.020*mental+ 0.018*watch+ 0.018*need+ 0.018*health+ 0.012*care+ 0.011*life	0.058*exhaust+ 0.044*time+ 0.041*like+ 0.035*tire+ 0.030*know+ 0.019*work+ 0.017*feel+ 0.017*mental+ 0.014*hate+ 0.013*start	0.039*time+ 0.037*trauma+ 0.037*like+ 0.032*exhaust+ 0.023*sleep+ 0.022*anxieties+ 0.020*think+ 0.019*come+ 0.017*fuck+ 0.016*life	0.128*suicide+ 0.075*help+ 0.069*thought+ 0.048*time+ 0.037*life+ 0.034*heal+ 0.029*take+ 0.026*anxieties+ 0.024*know+ 0.021*day
Topic 3	0.027*tire+ 0.022*depress+ 0.019*http+ 0.016*frustrate+ 0.016*go+ 0.013*get+ 0.012*work+ 0.010*live+ 0.010*exhaust+ 0.009*time	0.049*exhaust+ 0.043*go+ 0.021*want+ 0.020*sleep+ 0.018*thing+ 0.017*work+ 0.016*tire+ 0.014*feel+ 0.013*little+ 0.013*life	0.038*tire+ 0.037*anxieties+ 0.034*feel+ 0.034*year+ 0.032*like+ 0.027*hate+ 0.027*trauma+ 0.020*life+ 0.019*exhaust+ 0.018*pain	0.145*suicide+ 0.059*life+ 0.048*tri+ 0.041*go+ 0.034*attempt+ 0.030*http+ 0.023*anxieties+ 0.023*year+ 0.022*hate+ 0.021*thing
Topic 4	0.151*tire+ 0.028*sleep+ 0.026*feel+ 0.025*exhaust+ 0.024*today+ 0.022*work+ 0.016*good+ 0.015*drain+ 0.014*http+ 0.014*go	0.053*fuck+ 0.050*feel+ 0.035*exhaust+ 0.032*hate+ 0.024*shit+ 0.023*tire+ 0.021*like+ 0.020*hurt+ 0.017*year+ 0.016*go	0.031*hate+ 0.031*anxieties+ 0.029*self+ 0.023*thing+ 0.018*time+ 0.018*fuck+ 0.018*shit+ 0.017*feel+ 0.017*think+ 0.016*break	0.069*suicide+ 0.057*hurt+ 0.052*want+ 0.051*thought+ 0.050*go+ 0.040*love+ 0.034*know+ 0.026*like+ 0.026*worst+ 0.026*think
Topic 5	0.060*depress+ 0.030*peopl+ 0.020*suffer+ 0.019*suicide+ 0.019*like+ 0.014*help+ 0.013*year+ 0.013*thing+ 0.012*think+ 0.012*trauma	0.042*like+ 0.042*hurt+ 0.034*tire+ 0.029*want+ 0.027*shit+ 0.027*feel+ 0.022*today+ 0.019*know+ 0.014*get+ 0.012*see	0.059*feel+ 0.030*like+ 0.030*people+ 0.024*today+ 0.023*anxieties+ 0.017*go+ 0.017*want+ 0.016*tell+ 0.015*suffer+ 0.014*know	0.092*suicide+ 0.092*self+ 0.078*like+ 0.058*destruct+ 0.041*hate+ 0.035*feel+ 0.034*fuck+ 0.031*think+ 0.030*sleep+ 0.027*harm

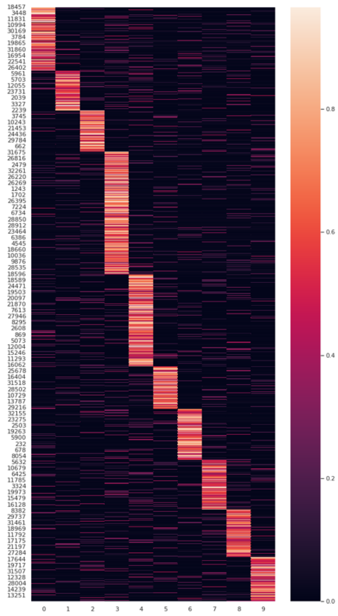

FIGURE 4.9: *Non-depressed* Class Topic Distribution

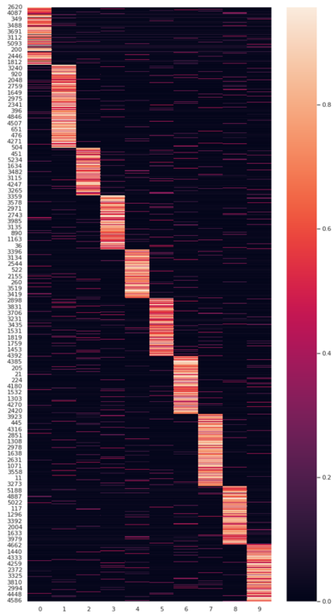

FIGURE 4.10: *Mild* Class Topic Distribution

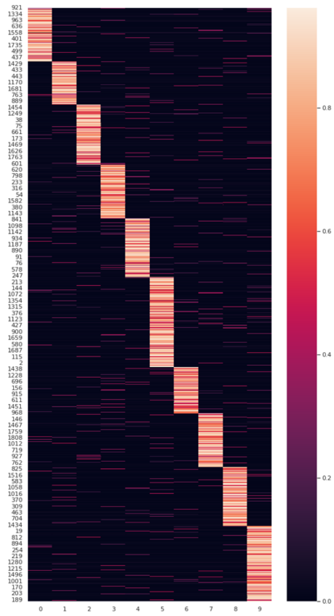

FIGURE 4.11: *Moderate* Class Topic Distribution

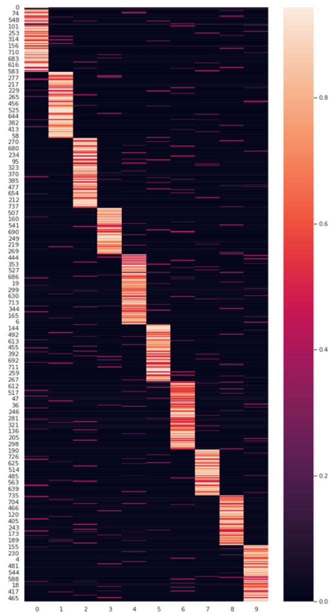

FIGURE 4.12: *Severe* Class Topic Distribution

FIGURE 4.13: Topic Distribution over Documents for all Classes

valuable insights about the lives of people suffering from depression. The prevalent terms in ‘Severely Depressed’ class, such as ‘suicide’, ‘thought’, ‘live’, ‘anymore’, etc., display the untold sufferings and mental struggles of a terminally depressed individual. The topic distribution over the documents of all the classes can be found in Figure 4.13.

Figure 4.13 reveals the prevalence of LDA topics in the data samples. For examples, topic 4 is prevalent among all the documents in *Non-depressed* class, while on the other hand, *severe* class has mostly uniform distribution of topics over its documents.

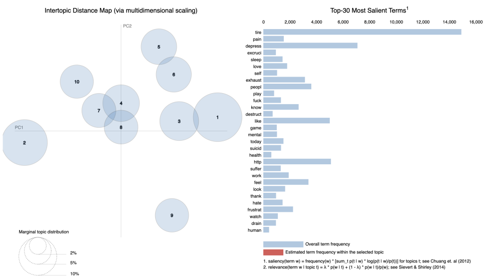

FIGURE 4.14: *Non-depressed* Salient terms and Intertopic distance Map

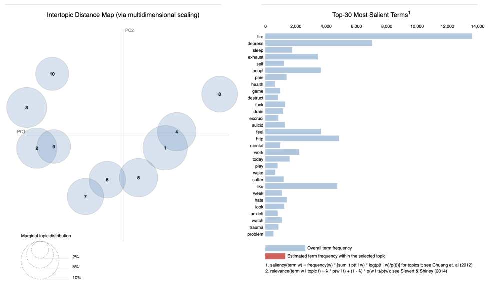

FIGURE 4.15: *Mild*Salient terms and Intertopic distance Map

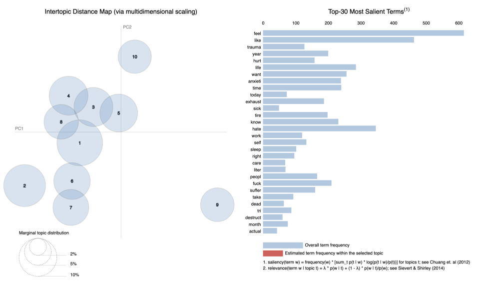

FIGURE 4.16: *Moderate* Salient terms and Intertopic distance Map

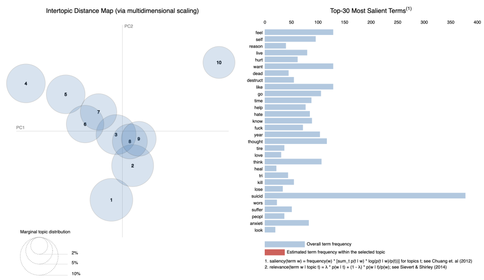

FIGURE 4.17: *Severe* Salient terms and Intertopic distance Map

FIGURE 4.18: Most Salient Terms and Inter-topic distance Map of all Classes

Finally, Figure 4.18 exhibits most salient terms and the intertopic distance map of all the classes. It is worth noticing that the topics in *Moderate* and *Severe* are classes are very well adjacent and connected, whereas the topics in *Non-depressed* class are rather sparse and have lesser inter-connection.

Chapter 5

System Design and Implementation

This chapter discusses about the available solutions to the potential problems identified through system modules to show the new system information flow, processes, tools to be used in its implementation.

5.1 Research Findings

Findings of the current study in terms of data input/output and processing are explained below.

5.1.1 System input/output

The system captures the input from the user as an input to the system. Those data input then undergoes analysis process to display system re-generated data insights based on probability of the machine learning algorithm model we have used.

5.1.2 System Processing

The new system captures the facial image of the parent/caregiver/trainer, then preprocesses by detecting facial features such as the eye, nose and mouth with their appropriate landmark points. These landmark points are analyzed and implemented using an API based models with different datasets of different facial expressions for display in a Google glass screen.

5.1.3 The Salient Thing

Children with ASD of age six and above were able to identify caregiver's emotions based on system re-generated facial expressions. They were also able to demonstratively read and display the appropriate caregiver's emotions based on text and system re-generated facial expressions and lastly they were able to relate system generated probability of facial expression to facial emotion of the caregiver successively.

5.2 System Design

This phase constitutes of system modeling which refers to the process of developing abstract models of a system with each model presenting a different view or perspective of the new system. It basically shows the flow of system functions through different Unified Modeling Language (UML) diagrams below. System modeling is categorized into two; Behavioral and Structural System Models.

5.3 Behavioral System Models(Dynamic)

This model shows the dynamic behavior of the objects in the new system which can be described as a series of changes to the system over time.

5.3.1 Use Case

As the most known diagram type of the behavioral UML diagrams, Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions are interacted. It consists of the following:

- Actors
- Use Case

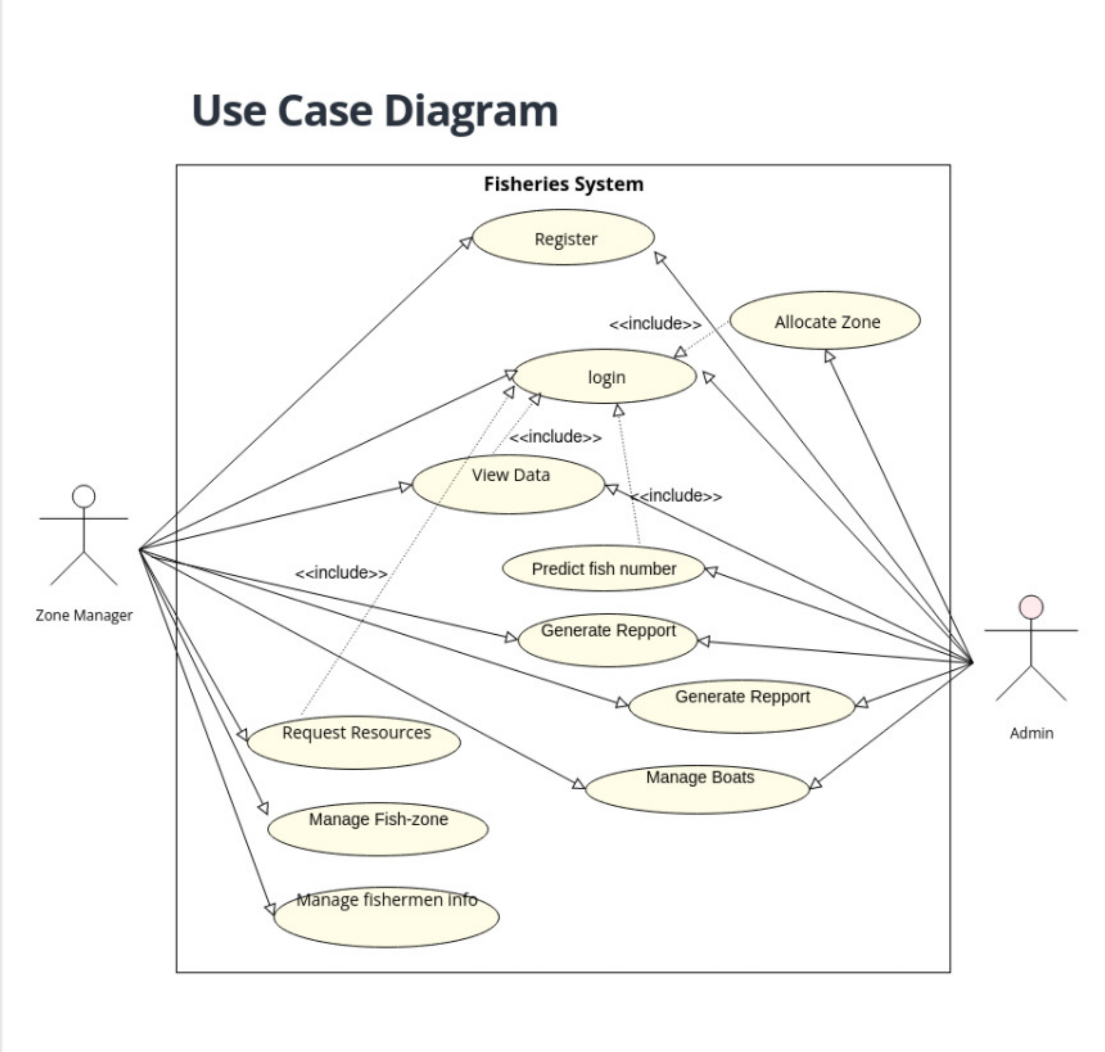

FIGURE 5.1: Snapshot of the use case diagram

- Relationships and System Boundary

Use Case:

Actor: Fishermen Pre-condition: Launch the application Post-condition: Input captured, analyzed and predict successfully

Input Representation

Before being fed into the pre-trained models for embedding, each tweet text are converted into an acceptable format. A single vector representing the entire input sentence is required to be passed to a classifier in order to complete the classification operation. BERT-based models use WordPiece tokenizer [13], which works by splitting the input sequence into full forms or word pieces. In case of full form, a word is represented by one token string, whereas, for word pieces, a word is represented by multiple token strings. Using word pieces helps the models to

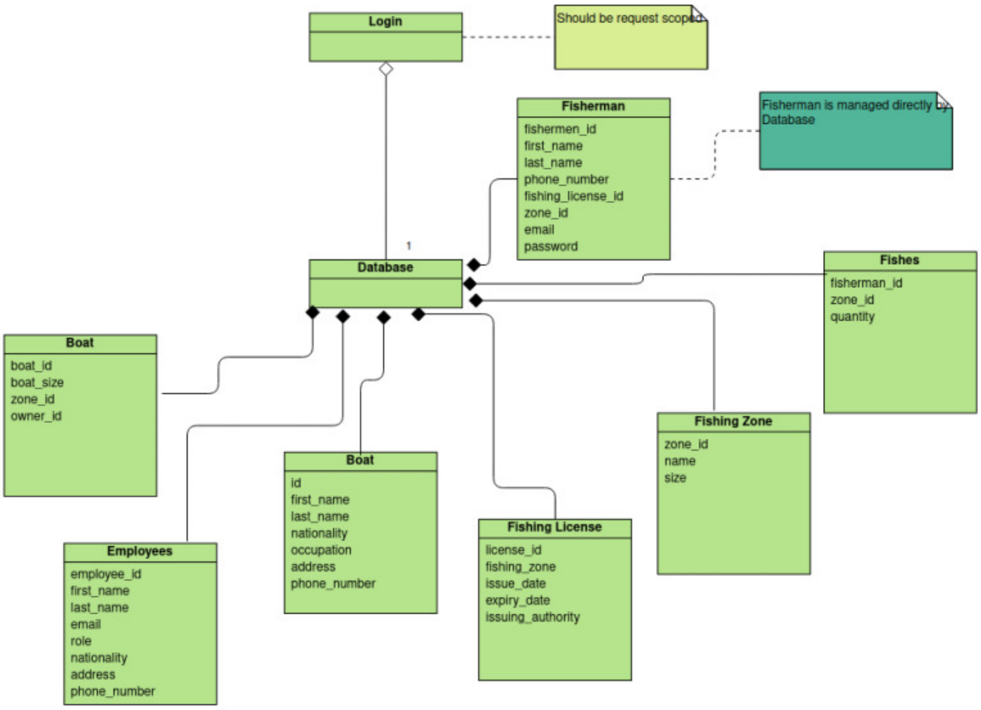

FIGURE 5.2: UML Diagram

identify related words as they share similar token strings, which is crucial for context understanding. Some special token strings are generated during tokenization to indicate the task type, beginning of input sequence, mask, etc., e.g.,

5.3.2 UML Diagram

UML diagrams represent workflows in a graphical way. They can be used to describe workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams

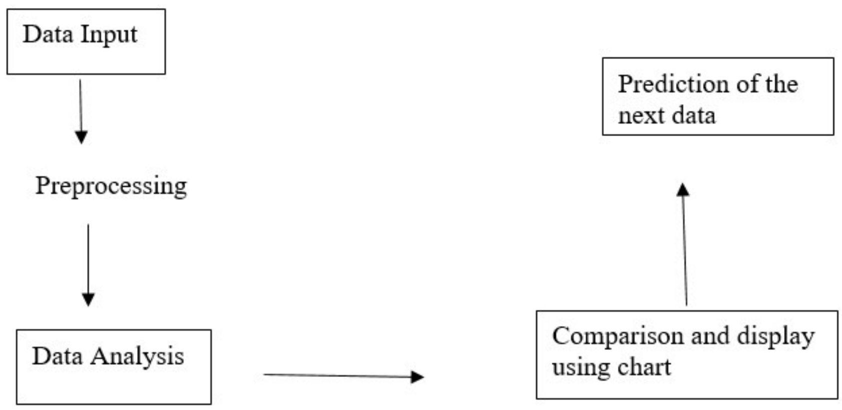

FIGURE 5.3: Data Flow Diagram

5.3.3 Data Flow Diagram

5.4 Interaction Design

The interaction design model applied was user-centered through full time [member of the design team] interaction where the user needs were identified for system requirement establishment. Once requirements were established the system was designed and implemented then tested for evaluation. Based on the system methodology used that is Agile, the process was repeated several times to meet the final user needs through the system.

5.5 System Implementation

The system is implemented using MERN Technology. It is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library used for machine learning applications such as neural networks.

5.5.1 Application Interfaces

Here are a few images of the user interface of the application including the frameworks that have been used. In figure **??** we have an overview of what the Home page look like for both users (i.e. Admin user and Zone Manager user).

FIGURE 5.4: Snapshot of the Home Page of The application

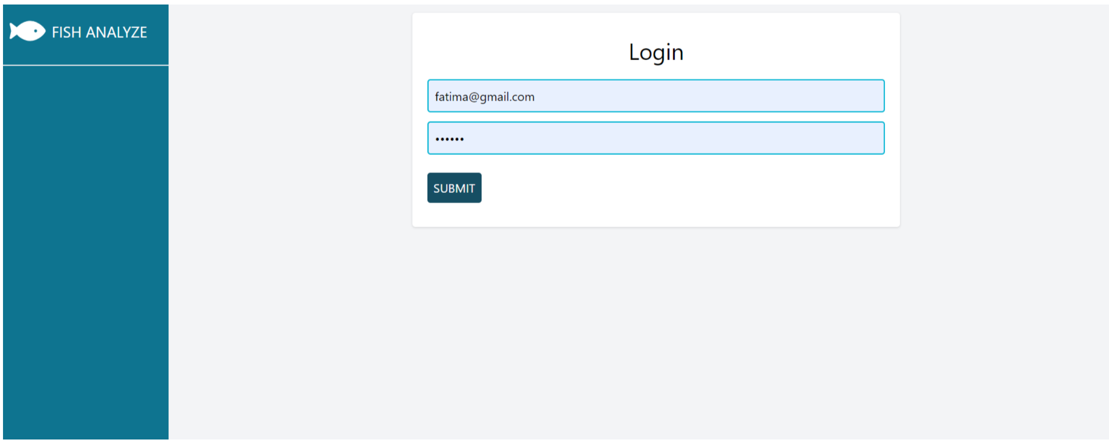

FIGURE 5.5: Snapshot of the Login Page of The application

First the login page of all the roles direct to the dashboard of all the regions or available zones for the admin user while the zone manager is having the same dashboard but with data which are only restricted to the zone he/she is responsible for. This is done based on the user credentials inserted during login and redirect the user to the corresponding dashboard of roles.

In the above figure, we can see the dashboard of the employee management where the admin user initiates the role for the zone manager user in the application. Here as an admin user, he can add user either as another admin or add a zone manager and allocate a particular zone for him. Figure 4.x shows us the registered zone manager user along with their information and permissions which can be manage by the admin user.

Figure 4.x and figurer 4.x show us the dashboard of the details regarding both the

FIGURE 5.6: Snapshot of the Employee dashboard (adding and removing users) of the application

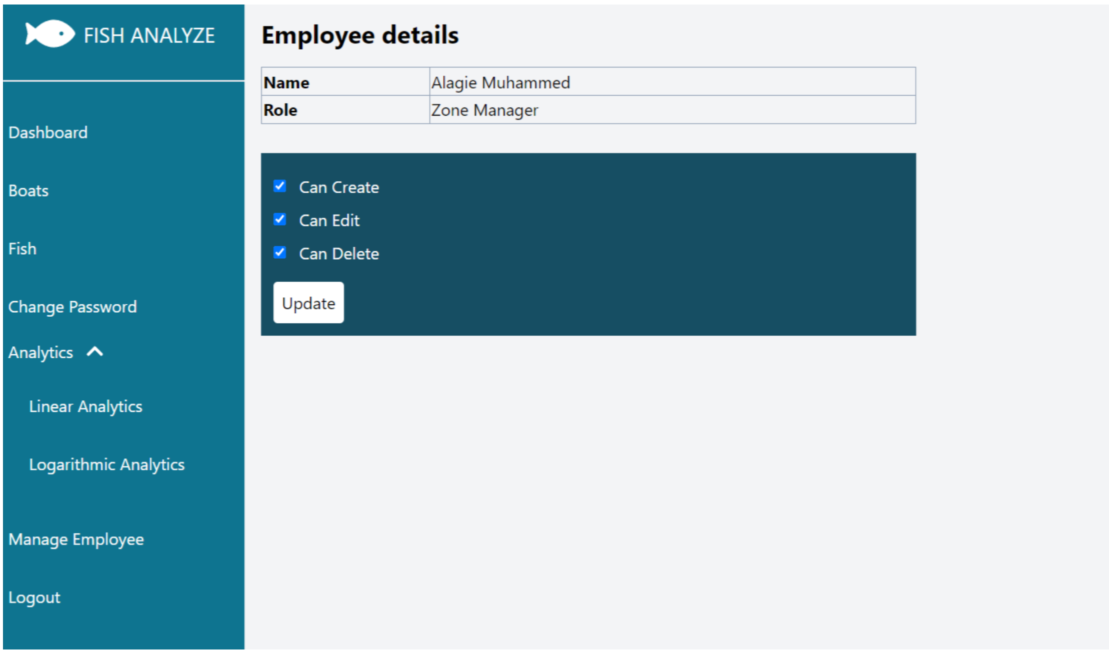

FIGURE 5.7: Snapshot of Permissions management of the application

fish activities as well as boats information respectively. Here, we can allocate new boats to be used as well as the number of fish that have been caught so that the application can process the information. For the fish dashboard, we can see when the fish has been added to the system and the quantity too. Coming to the boat details, we can see the number of registered boats in the current month, the date of registration and the size too are part of the information.

Now that we have the required data we can start doing what the application is supposed to do which predicting the next number of fishes to be caught in each

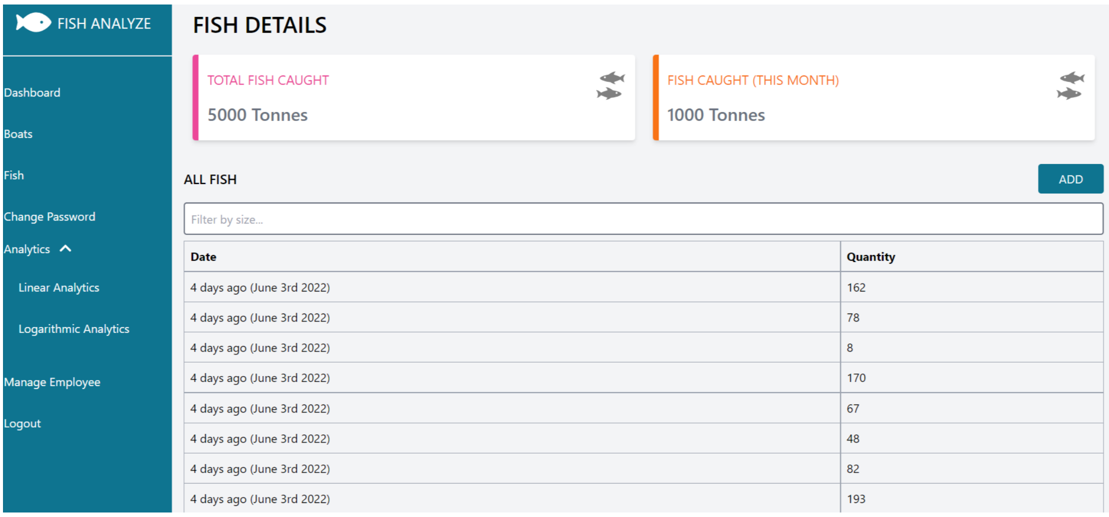

FIGURE 5.8: Snapshot of the fish details Dashboard of the application

FIGURE 5.9: Snapshot of the Boats details Dashboard of the application

and every available and registered zone in the system. In our case, [14] we followed two types of machine learning model (i.e. Linear Regression and Logarithmic). In figure 4.x we have the prediction based on Linear regression and in figure 4.x it is based on Logarithmic approach.

Now that we have the required data we can start doing what the application is supposed to do which predicting the next number of fishes to be caught in each and every available and registered zone in the system. In our case, we followed two types of machine learning model (i.e. Linear Regression and Logarithmic). In figure 4.x we have the prediction based on Linear regression and in figure 4.x it is based on Logarithmic approach.

FIGURE 5.10: Snapshot of the Boats details Dashboard of the application

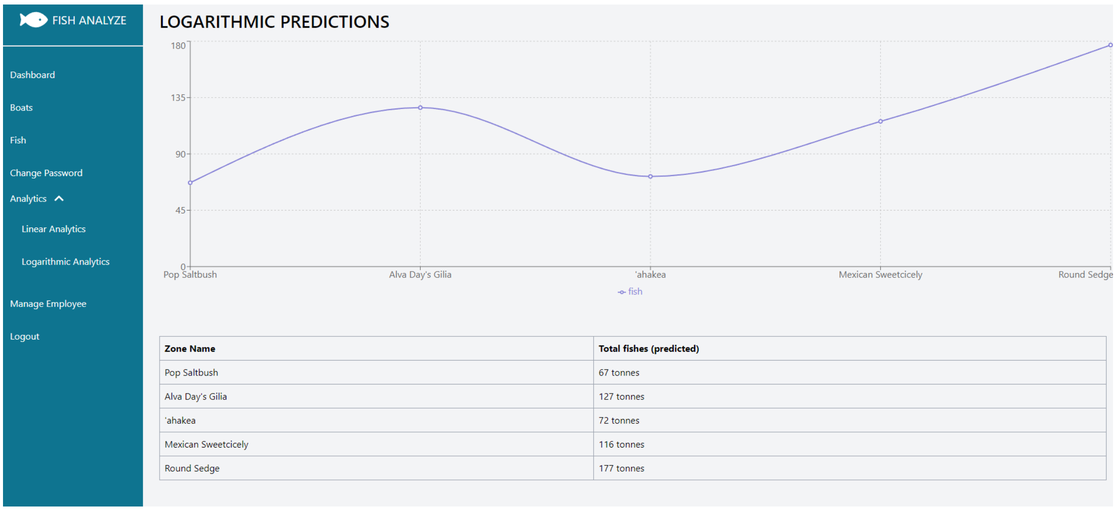

FIGURE 5.11: Snapshot of the Boats details Dashboard of the application

Chapter 6

System Testing and Validation

This chapter describes how testing and validation tasks were performed. The test tools used and test procedures for testing the functionalities of the application. Since the system was web-based, the testing was conducted by use of two different machine learning algorithm. The prototype was designed using MERN technologies which are MongoDB for the database, Express as a framework and React and NodeJS for frontend and backend respectively. Verification process was conducted through checking documents, design, code, and program in order to see if the system was built according to the requirements or not for system quality assurance.

6.1 System validation

System Validation was achieved through activities like unit testing, integration testing, system testing and user acceptance testing. This helped to check if the system product actually met the exact needs of the fishing activities through having fulfilled the desired possession of information and use in the appropriate environment.

6.2 System Evaluation

The system was implemented using MERN Technologies and tested using machine learning algorithm models several times as user needs continued to be identified through evaluating design against requirements. The system was evaluated using

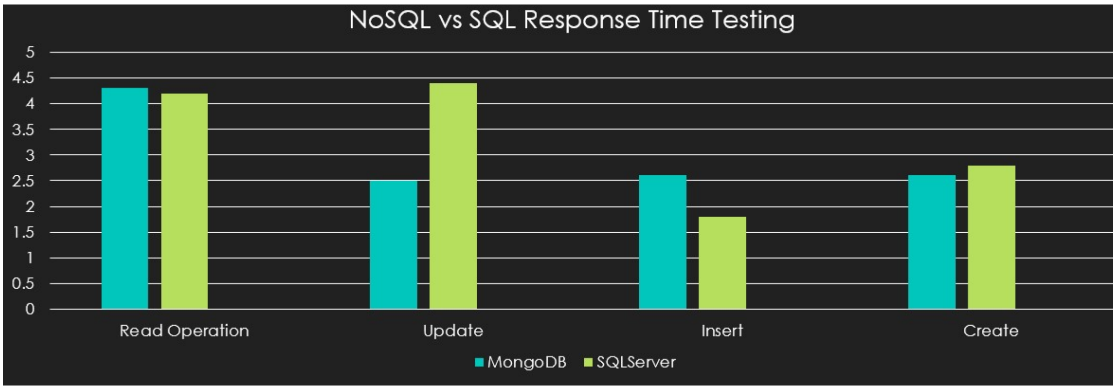

FIGURE 6.1: Comparison of CRUD operation between SQL and NoSQL Database.

the star model and also through the accuracy measurement using mathematical calculations. This iterative procedure was done through task analysis and re-adjustment of the functional requirements to meet user needs of the system. New user needs continuously kept affecting requirement specifications, conceptual designs, their implementation and prototypes until the goal was met.

6.3 Discussion

The study outcomes of [15] were basically classified into three; firstly, a prediction system to enhance fishing activities. This was achieved through use of high and several iteration of training the system to automatically generate the appropriate output. shape and weight that revealed the main target features with better visualization of the facial expressions compared to the typical [original facial expression] of the trainer. Secondly, the explanation and interpretation of fishing activities with respect to zone. In this way, it will definitely provide features with better visualizations of the fishing activities compared to the typical users' expectations. Thirdly, the ability to process the data in real time. This is because we will have a large volume of data and we need to balance the load so that the system can be very scalable. Also, we will have the application running in multiple regions, which therefore justify the scalability of our system. This way will help the users which are the fishermen to have the exact and required trusted information regarding the fishing activities all around where the system is implemented which in turn will guarantee the food security.

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

The prototype we presented here in this paper was designed to track, monitor and predict the fishing activities for the purpose of fighting illegal fishing as well as to have security of food. It was a wonderful learning experience for us while working on this project. It took us through the various phases of project development and gave us real insight into the world of software engineering. The prototype results revealed successful improvements in fishing related activities information which was unknown to the government and the fishermen. However, the system was able to analyze and predict data insight captured from the training session for easier information gathering about a fishing zone based on system output. This output presented by the system was able to address different issues and challenges which are encountered in those environment. Therefore, the system was able to support a range of communication related issues and enhanced prediction output though it might still be difficult to create a design to apply to all types of data. The real effect of the application is not yet estimated and is to be validated in future when executed in the field itself. Some degree of flexibility such as personalizing the designs for instance a login system for the fishermen user is yet to be added.

7.2 Recommendations

Since fishing activities [16] include a wide spectrum of issues and challenges, future researchers and designers should consider both the policy of where the system should be implemented and the knowledge of the fishermen which is in most of the case illiterate people into their design for more specific results.

Appendix A

Appendix

A.1 Source Code

A.1.1 FrontDashboard

import React, useEffect from "react"; import useDispatch, useSelector from "react-redux"; import useHistory from "react-router-dom"; import fetchBoatStats, fetchFishStats, fetchZoneStats, from "../actions/statsAction"; import AuthError from "../components/AuthError"; import Charts from "../components/dashboard/Charts"; import Stats from "../components/dashboard/Stats"; import Loading from "../components/Loading"; import FETCH$_{B}OAT_{S}TATS_{R}ESET, FETCH_{F}ISH_{S}$

const DashboardScreen = () =¿ const userInfo = useSelector((state) =¿ state.userLogin); const loading, error, fishStats = useSelector((state) =¿ state.fishStats); const loading: boatLoading, error: boatError, boatStats, = useSelector((state) =¿ state.boatStats); const loading: zoneLoading, error: zoneError, zoneStats, = useSelector((state) =¿ state.zoneStats); const dispatch = useDispatch(); const history = useHistory(); useEffect(() =¿ if (!userInfo) history.push("/"); else if(!fishStats) dispatch(fetchFishStats()); if(!boatStats) dispatch(fetchBoatStats()); if(!zoneStats) dispatch(fetchZoneStats()); , [userInfo, history, dispatch, fishStats, boatStats, zoneStats]); return (¡div¿ (loading —— boatLoading —— zoneLoading) (¡Loading text="Analyzing... Please wait" /¿)

error ¡AuthError error=error handleClose=() =¿ dispatch(type: FETCH$_{F}ISH_{S}TATS_{R}ESET)/$
$boatError < AuthErrorerror = boatErrorhandleClose = () => dispatch(type : FETCH_{B}OA$

¡div className="m-3 flex items-center justify-between"¿ ¡h2 className="text-2xl"¿DASHBOARD¡/h2¿ /* ¡button onClick=() =¿ history.push("/report") disabled className="p-2 bg-cyan-700 text-white border-2 rounded shadow-md hover:bg-white hover:text-black" ¿ GENERATE REPORT ¡/button¿ */ ¡/div¿ ¡Stats fishStats=fishStats boatStats=boatStats zoneStats=zoneStats /¿ ¡Charts fishStats=fishStats ? fishStats.overveiw : null boatStats=boatStats ? boatStats.overveiw : null zoneStats=zoneStats ? zoneStats.overveiw : null /¿ ¡/div¿); ;

export default DashboardScreen;

A.1.2 Backend Dashboard

const router = require("express").Router(); const Fish = require("../models/Fish"); const Zone = require("../models/Zone");

router.get("/", async (req, res) =¿ try const fishes = await Fish.find(); let stats = ; fishes.forEach(async (fish, i) =¿ const data = await Zone.find($_id : fish.zoneId); if(data.length > 0) const newData = fish : fish.quantity, zone : data[0].name, si$

if (stats[newData.zone]) stats[newData.zone] = fish: newData.fish, zone: newData.zone, size: newData.size, ; else stats[newData.zone] = newData; if (i === fishes.length - 1) res.json(msg: stats);); catch (err) console.log(err); res.status(500).json(msg: "Server Error"););

module.exports = router;

A.1.3 Employee Dashboard

import React, useEffect from "react"; import FaPlus from "react-icons/fa"; import useDispatch, useSelector from "react-redux"; import useHistory from "react-router-dom"; import deleteEmployeeAction, getEmployeeAction from "../actions/employeeAction"; import AuthError from "../components/AuthError"; import DELETE$_EMPLOYEE_RESET, GET_EMPLOYEE_RESET, GET_SINGLE_EMPLOYEE$
$() => const userInfo = useSelector((state) => state.userLogin); const loading, employees, err$

¡div className="p-3"¿ getEmployeeError ¡AuthError error=getEmployeeError handleClose=() =¿ dispatch(type: GET$_EMPLOYEE_RESET)/ > deleteEmployeeError < Au$
$br/ > /* < div className = "mb - 2" >< input type = "text" placeholder = "Filter by name..."$
$table className = "table - fixed w - full border - collapse border border - gray -$

400” $>< thead >< trclassName = ”mb - 3” >< thclassName = ”text - leftborder-collapseborderborder-gray-400p-2” > Name < /th >< thclassName = ”text - leftborder - collapseborderborder - gray - 400p - 2” > Role < /th >< thclassName = ”text - leftborder - collapseborderborder - gray - 400p - 2” > Zone < /th >< thclassName = ”text - leftborder - collapseborderborder - gray - 400p - 2” > Action < /th >< /tr >< /thead >< tbody > loading? < h2 > Loading... < /$

¡td className=”my-3 border-collapse border border-gray-400 p-2”¿ employee.firstName + employee.lastName ¡/td¿ ¡td className=”my-3 border-collapse border border-gray-400 p-2”¿ employee.role ¡/td¿ ¡td className=”my-3 border-collapse border border-gray-400 p-2”¿ employee.zone ¡/td¿ ¡td className=”my-3 border-collapse border border-gray-400 p-2 ”¿ ¡div className=”flex items-center justify-end”¿ ¡button className=”p-2 border-2 bg-cyan-900 mx-1 text-white text-sm” onClick=() =¿ history.push(‘/manage-permission/$employee._id‘) > MANAGEPERMISSIONS < /button >< buttononClick = () => dispatch(deleteEmployeeAction(employee._id))className$ $”p - 2border - 2bg - red - 600mx - 1text - whitetext - sm” > DELETE < /button >< /div >< /td >< /tr >) :< h2 > NoEmployees < /h2 >< /tbody >< /table >$

¡div className=”my-5 mx-auto flex items-center justify-center py-5”¿ pagination pagination.prev ¡button onClick=() =¿ dispatch(getEmployeeAction(” , pagination.prev.page)) className=”p-2 px-5 rounded bg-cyan-700 text-white mx-2”¿Prev¡/button¿ pagination pagination.next ¡button onClick=() =¿ dispatch(getEmployeeAct pagination.next.page)) className=”p-2 px-5 rounded bg-cyan-700 text-white mx-2”¿Next¡/button¿ ¡/div¿ ¡/div¿ ¡/div¿); ;

export default EmployeeScreen;

A.1.4 Analytics

const router = require(”express”).Router(); const Fish = require(”../models/Fish”); const Zone = require(”../models/Zone”);

router.get(”/”, async (req, res) =¿ try const fishes = await Fish.find(); let stats = ; fishes.forEach(async (fish, i) =¿ const data = await Zone.find($_id : fish.zoneId); if(data.length > 0)constnewData = fish : fish.quantity, zone : data[0].name, si$

if (stats[newData.zone]) stats[newData.zone] = fish: newData.fish, zone: newData.zone, size: newData.size, ; else stats[newData.zone] = newData; if (i

=== fishes.length - 1) res.json(msg: stats);); catch (err) console.log(err); res.status(500).json(msg: ”Server Error”););

module.exports = router;

Bibliography

[1] J. Gladju, B. S. Kamalam, and A. Kanagaraj, "Applications of data mining and machine learning framework in aquaculture and fisheries: A review," *Smart Agricultural Technology*, vol. 2, p. 100061, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375522000260

[2] J. Nidzwetzki and R. Güting, "Distributed secondo: an extensible and scalable database management system," *Distributed and Parallel Databases*, vol. 35, 12 2017.

[3] W. N. Probst, "How emerging data technologies can increase trust and transparency in fisheries," *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1286–1294, 03 2019. [Online]. Available: https://doi.org/10.1093/icesjms/fsz036

[4] G. Arbanas, "Diagnostic and Statistical Manual of Mental Disorders (DSM-5)," *Alcoholism and psychiatry research*, vol. 51, pp. 61–64, 2015. [Online]. Available: https://www.amberton.edu/media/Syllabi/Spring%202022/Graduate/CSL6798_E1.pdf

[5] C. Pala, "Tracking f shy behavior, from space," *The Atlantic*, vol. 16, 2014.

[6] A. Vermeulen, H. Vandebosch, and W. Heirman, "#Smiling, #venting, or both? Adolescents' social sharing of emotions on social media," *Computers in Human Behavior*, vol. 84, pp. 211–219, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563218300803

[7] J. Kim, J. Lee, E. Park, and J. Han, "A deep learning model for detecting mental illness from user content on social media," *Scientific reports*, vol. 10, no. 1, pp. 1–6, 2020.

[8] D. M. Low, L. Rumker, T. Talkar, J. B. Torous, G. A. Cecchi, and S. S. Ghosh, "Natural language processing reveals vulnerable mental health support groups and heightened health anxiety on reddit during covid-19: Observational study," *Journal of Medical Internet Research*, vol. 22, 2020.

[9] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical Methods for Rates and Proportions*, 3rd ed., ser. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0471445428

[10] K. L. Gwet, *Handbook of Inter-rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters.* Advanced Analytics, LLC, 2014.

[11] Leard Statistics, "Fleiss' kappa in SPSS statistics," 2019. [Online]. Available: https://statistics.laerd.com/spss-tutorials/fleiss-kappa-in-spss-statistics.php

[12] J. O. Salminen, H. A. Al-Merekhi, P. Dey, and B. J. Jansen, "Inter-Rater Agreement for Social Computing Studies," in *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS).* IEEE, 2018, pp. 80–87. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8554744

[13] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[14] N. B. V. Le and J.-H. Huh, "A design of smart aquaculture recommender system applying big data analytics," 07 2021.

[15] S. B. Saila, M. Wigbout, and R. J. Lermit, "Comparison of some time series models for the analysis of fisheries data," *ICES Journal of Marine Science*, vol. 39, no. 1, pp. 44–52, 04 1980. [Online]. Available: https://doi.org/10.1093/icesjms/39.1.44

[16] H. Li, Z. Li, S. Peng, J. Li, and C. E. Tungom, "Mining the frequency of time-constrained serial episodes over massive data sequences

and streams," *Future Generation Computer Systems*, vol. 110, pp. 849–863, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18332138