**HIFAZAT ~ Saving Lives**

**Capstone Project Report**

**MID SEMESTER EVALUATION**

**Submitted by:**

**(101917014) Ritik Rajdev**

**(101917023) Akshat Dogra**

**(101917125) Harshita Gupta**

**(101917179) Kavya Jaswani**

**BE Third Year, COPC**

**CPG No: 69**

Under the Mentorship of: **Dr. Ashima Anand**

Assistant Professor, CSED, TIET, Patiala



**Computer Science and Engineering Department**

**Thapar Institute of Engineering and Technology, Patiala**

**ABSTRACT**

Our primary goal is to develop a software that can decrease the rescue time by employing machine learning models to identify patterns in the victim's screams.

It has been noted that the victim in these cases typically lets out a scream or other loud noise. If they are taken hostage in any way, generally their position coordinates substantially shift. Therefore, even if a scream is not heard, it is usually possible to determine from their location whether they are in danger. In the event of an accident, their stationary position and any sounds can provide a general indication as to whether or not an accident has occurred. These occurrences can be used as indicators to establish whether or not someone is in danger. These occurrences can be used as clues to identify whether someone is in danger or not, and we can then alert the necessary contacts to what we have seen so they can at least confirm whether assistance or rescue is actually needed.

By alerting the people the user specifies, this software aims to save lives by dramatically increasing the likelihood that quick action will be taken, resulting in the timely rescue of accident victims, the detection of rape cases and domestic violence, among other things, depending on the situation.

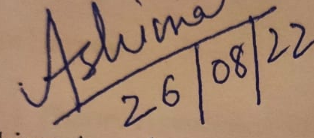
**DECLARATION**

We hereby declare that the design principles and working prototype model of the project entitled ***Hifazat ~ Saving Lives*** is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of **Dr. Ashima Anand** during 6th semester (2020).

Date: 26-08-2022

|  |  |  |
| --- | --- | --- |
| **Roll Number** | **Name** | **Signature** |
| 101917014 | Ritik Rajdev |  |
| 101917023 | Akshat Dogra |  |
| 101917125 | Harshita Gupta |  |
| 101917179 | Kavya Jaswani |  |

*Counter Signed By:*



**Faculty Mentor:** Dr. Ashima Anand **Designation:** Assistant Professor, CSED, TIET, Patiala

**ACKNOWLEDGEMENT**

We would like to express our thanks to our mentor **Dr. Ashima Anand**. She has been of great help in our venture, and an indispensable resource of technical knowledge. She is truly an amazing mentor to have.

We are also thankful to **Dr. Shalini Batra**, Head, the Computer Science and Engineering Department, the entire faculty and staff of the Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards the successful completion of this project. We thank all those who have contributed either directly or indirectly to this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement.

They always wanted the best for us and we admire their determination and sacrifice.

Date: 26-08-2022

| **Roll Number** | **Name** | **Signature** |
| --- | --- | --- |
| 101917014 | Ritik Rajdev |  |
| 101917023 | Akshat Dogra |  |
| 101917125 | Harshita Gupta |  |
| 101917179 | Kavya Jaswani |  |

**TABLE OF CONTENTS**

**ABSTRACT……………………………………………………………………i**

**DECLARATION………………………………………………………………ii**

**ACKNOWLEDGEMENT…………………………………………………….iii**

**LIST OF FIGURES…………….……………………………………………...iv**

**LIST OF TABLES……………………………………………………………..v**

**LIST OF ABBREVIATIONS.………………………………………………...vi**

**CHAPTER…..…………………………………………………………**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** | **Introduction** | |  |
|  | 1.1 | Project Overview |  |
|  | 1.2 | Need Analysis |  |
|  | 1.3 | Research Gaps |  |
|  | 1.4 | Problem Definition and Scope |  |
|  | 1.5 | Assumptions and Constraints |  |
|  | 1.6 | Standards |  |
|  | 1.7 | Approved Objectives |  |
|  | 1.8 | Methodology |  |
|  | 1.9 | Project Outcomes and Deliverables |  |

1.10 Novelty of Work

**2. Requirement Analysis**

2.1 Literature Survey

2.1.1 Theory Associated With Problem Area

2.1.2 Existing Systems and Solutions

2.1.3 Research Findings for Existing Literature

2.1.4 Problem Identified

2.1.5 Survey of Tools and Technologies Used

2.2 Software Requirement Specification

2.2.1 Introduction

2.2.1.1 Purpose

2.2.1.2 Intended Audience and Reading Suggestions

2.2.1.3 Project Scope

2.2.2 Overall Description

2.2.2.1 Product Perspective

2.2.2.2 Product Features

2.2.3 External Interface Requirements

2.2.3.1 User Interfaces

2.2.3.2 Hardware Interfaces

2.2.3.3 Software Interfaces

2.2.4 Other Non-functional Requirements

2.2.4.1 Performance Requirements

2.2.4.2 Safety Requirements

2.2.4.3 Security Requirements

2.3 Cost Analysis

2.4 Risk Analysis

* 1. **Methodology Adopted**

3.1 Investigative Techniques

3.2 Proposed Solution

3.3 Work Breakdown Structure

3.4 Tools and Technology

1. **Design Specifications**

4.1 System Architecture

4.2 Design Level Diagrams

4.3 User Interface Diagrams

4.4 Snapshots of Working Prototype

1. **Conclusions and Future Scope**

5.1 Work Accomplished

5.2 Conclusions

5.3 Social Benefits

5.4 Future Work Plan

**APPENDIX A: References**

**APPENDIX B: Plagiarism Report**

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Table No.** | **Caption** | **Page No.** |
|  |  |  |  |
|  | Table 1. | Assumptions and Constraints | 14 |
|  |  |  |  |
|  | Table 2 | Research Findings for Existing Literature | 18 |
|  |  |  |  |
|  | Table 3 | Investigative Techniques | 22 |
|  | Table 4 | Student Outcomes Description and Performance Indicators | 41 |
|  | Table 5 | Peer Assessment Matrix | 39 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Figure No.** | **Caption** | **Page No.** |
|  |  |  |  |
|  | Figure 1 | Neural network of our model | 12 |
|  |  |  |  |
|  | Figure 2 | Work Breakdown Structure | 24 |
|  |  |  |  |
|  | Figure 3 | Design Level Diagram of Hifazat (level-0) | 25 |
|  | Figure4 | Design Level Diagram of Hifazat (level-1) | 26 |
|  | Figure 5 | Sequence Diagram. | 26 |
|  | Figure 6 | Screenshot of Prototype. | 27 |
|  | Figure7 | Loss value and accuracy | 32 |
|  | Figure 8 | Precision, recall and F1 score | 32 |
| F | Figure 9 | Table depicting Precision and Accuracy | 33 |
|  | Figure 10 | Block Diagram | 35 |
|  | Figure 11 | Startup Page | 35 |
|  | Figure 12 | Adding SOS Contacts | 36 |
|  | Figure 13 | Manually deactivating the noise detection | 37 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Name** |
| NCRB | National Crime Records Bureau |
| SNR | Signal to noise ratio |
| GPS | Global policing system |
| SOS | Save our souls |
| SVM | Support Vector Machine |

**INTRODUCTION**

**1.1 Project Overview**

Hifazat is an application with the purpose to decrease rescue time and save lives.

Hifazat uses a noise detection technology that detects your scream and informs your trusted contacts, as well as the Police optionally depending on the results it draws by analyzing the type of noise.

We aim to create an application that is open source, lightweight, cross-platform, easy to use, and is backward compatible till android version 6.1.1 and IOS 11.

We had to use deep learning as there is very limited support for SKlearn and other libraries in android. Moreover, around 95.84% of the Indian population uses android, and 31.31% of users among them use android 9. Provided the current android 13, we were severely limited with options but deep learning provided us with an accuracy of around 95-98% which is on par with its competitors.

The working of our software is mentioned below:

● On startup, a daemon will observe the environment.

● Whenever a noise is detected, daemon will contact the application.

● Our application will categorize the scream on the basis of certain emotions.

● If the scream is classified as of the one in danger or hurt, this application will record a

voice clip of 10 seconds.

● This voice clip is then sent to contacts appropriated by the user.

● These users can contact the Police or Ambulance or can cancel the action after listening to that voice clip using provided buttons.

Location support will also be available to further decrease the rescue time.

The following is the diagram of our neural network:

**Figure 1 : Neural Network of our model**



**1.2 Need Analysis**

Road traffic injuries are currently estimated to be the 8th[1] leading cause of death across all age groups globally, and are predicted to become the seventh leading cause of death by 2030.Care of injuries after a crash has occurred is extremely time-sensitive: delays of minutes can make the difference between life and death.

We aim to provide software that can help detect an accident and report it immediately in order to decrease the rescue time.

Rape is the fourth most common crime against women in India. According to the 2019 annual report[2] of the National Crime Records Bureau, 32033 rape cases were registered across the country.

An opinion survey among experts carried out by the Thomson Reuters Foundation ranked India as the most dangerous country in the world for women.[5]

**1.3 Research Gaps**

1. Fairuz Samiha Saeed, Abdullah Al Bashit 2, Vishu Viswanathan and Damian Valles:

Since SVM is better suited for noise detection rather than scream detection and in our case we are tuning our data for scream detection as it is relatively more sensitive than noises such as one made in road accidents, so we should rather not use SVM. SVM only provided an accuracy of around 90% . When used deep learning , the accuracy is further increased to 95%.

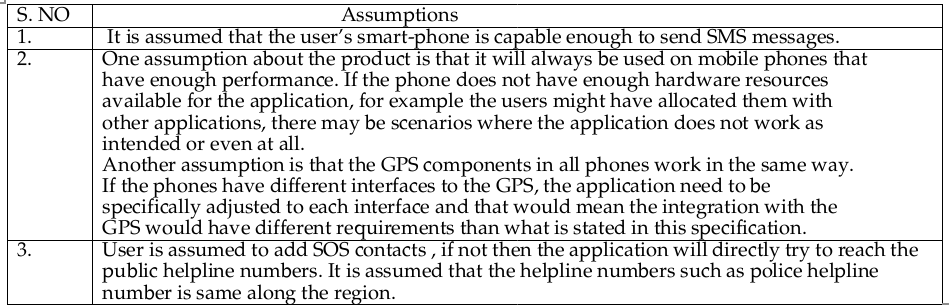
1. While using SNR , the number of events missed by the classifier grows as the noise level of tested samples increases and since, to train the model adequately we have to use a large dataset as screams can vary greatly as it depends on various factors such as pitch, intensity, etc, so a large dataset is required to train the model. As a result, this model also only gives an accuracy around 90%.

**1.4 Problem Definition and Scope**

The rescue time in various accidents such as accident , rape and domestic violence is extremely time sensitive , so in order to minimize casualties , we need to provide a framework that can help detect the incident and hence report it immidiately to the designated people and police.

Scope :

* Currently , there are many SOS applications in play store and IOS store , but most of them require manual button press to alert contacts and the rest only provide automatic detection by GPS tracking
* **1.5 Assumptions and Constraints**



**Table 1.** Assumptions and Constraints

**1.6 Standards**

* android version >= 6.1.1 or IOS version >= 12

* **IEEE 830-1998:** There are several examples of software requirements specifications (SRS) outlined and the content and qualities of a good SRS are described. In addition to specifying the requirements of the software to be developed, this recommended practice can assist in selecting in-house and commercial software.

* **IEEE/ISO/IEC P23026:** This document defines system engineering and management requirements for the life cycle of websites. This document applies to those using web technology to present information and communications technology (ICT) information, such as information for users of systems and services, plans and reports for systems and software engineering projects, and documentation of policies, plans, and procedures for IT service management.

* **IEEE SA - P3123:** The standard defines specific terminology utilized in artificial intelligence and machine learning (AI/ML). The standard provides clear definitions for relevant terms in AI/ML. Furthermore, the standard defines requirements for data formats.

**1.7 Approved Objectives**

* Train a model to detect noise and classify it efficiently.
* Make a lightweight, cross-platform and easy-to-use application.
* Format data and send messages to SOS contacts.

**1.8 Methodology**

We are classifying noise and using the detected noise as a cue that the user might be in danger. To classify noise , we are using a model trained on deep learning. When the noise is detected , the software will ping the location and will send it to the designated SOS contacts.

**1.9 Project Outcomes and Deliverables**

* An application with minimal and easy to user interface.
* A system that can classify noise based on a variety of criteria, determine whether or not a person is in danger, and record that specific noise.
* An automated SOS message with an audio file that is the smallest size feasible and maintains audio quality will be sent to the contacts who have been notified.
* Location sharing to keep an eye out for any odd movements in the event that no cries are heard.
* User interface cross-platform optimization.

**1.10 Novelty of Work**

1. Currently, SOS applications have limited features and mostly require the user to manually ping the SOS contacts. We propose to make software that optionally needs the user to ping SOS contacts.

**REQUIREMENT ANALYSIS**

**2.1 Literature Survey**

**2.1.1 Theory Associated With Problem Area**

**Rape In India**

Rape is the fourth most common crime against women in India. According to the 2019 annual report of the NCRB, 32033 rape cases were registered across the country, or an average of 88 cases daily. The share of victims who were minors or below 18 the legal age of consent in India stood at 15.4%.Among these, victims who do report the assaults are alleged to suffer mistreatment and humiliation from the police.In 2018, official data showed that 1 rape was reported every 15 minutes in India. Of the 34,000 cases reported, just over 85 per cent led to charges, and 27 per cent ultimately led to convictions. This was then increased to 16 cases per minute in 2019. Estimation Of Unreported Rapes Most rapes go unreported because the rape victims fear retaliation and humiliation, both in India and throughout the world.Indian parliamentarians have stated that the rape problem in India is being underestimated because many cases are not reported, even though more victims are increasingly coming out and reporting rape and sexual assaults.The National Crime Records Bureau report of 2006 mentions that about 71% rape crimes go unreported. Some of the notable incidents are :

1. Ajmer rape case In 1992, the Ajmer rape case was one of India's biggest cases of coerced sexual exploitation, with more than a hundred underage schoolgirls estimated to have been sexually molested and raped.
2. 2012 Delhi Gang rape case The gang rape of a 23-year-old student on a public bus, on 16 December 2012, sparked large protests across the capital Delhi.She was with a male friend who was severely beaten with an iron rod during the incident.This same rod was used to penetrate her so severely that the victim's intestines had to be surgically removed, before her death thirteen days after the attack.
3. 2013 Mumbai gang rape In August 2013, a 22-year-old photojournalist, who was interning with an English-language magazine in Mumbai, was gang-raped by five persons, including a juvenile, when she had gone to the deserted Shakti Mills compound, near Mahalaxmi in South Mumbai, with a male colleague on an assignment. This caused protests throughout the country since Mumbai with its very active nightlife was previously considered a safe haven for women. The city sessions court found the accused guilty and sentenced death penalty to the three repeat offenders in the Shakti Mills gang rape case, making them the first in the country to get the death sentence stipulated under the newly enacted Section 376E of the Indian Penal Code.

**Traffic Collisions**

Traffic collisions in India[2] are a major source of deaths, injuries and property damage every year. The NCRB 2016 report states there were 496,762 roads, railways and railway crossing-related traffic collisions in 2015.Of these, road collisions accounted for 464,674 collisions which caused 148,707 traffic-related deaths in India.According to the 2013 global survey of traffic collisions by the UN World Health Organization, India suffered a road fatality rate of 16.6 per 100,000 people in 2013. Traffic collision-related deaths increased from 13 per hour in 2008 to 14 per hour in 2009. In 2015, 15 people per hour died due to road accidents according to NGO 'Indians for Road Safety'.According to road traffic safety experts, the actual number of casualties may be higher than what is documented, as many traffic collisions go unreported. Moreover, victims who die some time after the collision, a span of time which may vary from a few hours to several days, are not counted as car crash victims.

**Kidnappings In India**

In 2020, there were over 84 thousand kidnapping and abduction cases reported in India[3][4]. The leading reason for this crime was to compel women to get married with over 24 thousand cases that year. Although the threat of kidnapping exists throughout the country, two states and one territory in northern India -- Bihar, Uttar Pradesh and Delhi -- accounted for 50 percent of all abductions from 2004 through 2006. Bihar was the site of one-quarter of all kidnappings reported during that period. Following Bihar was Delhi, the national capital, with 16 percent of all abductions, and Uttar Pradesh, with 9 percent.

**Domestic Violence**

In India According to a National Family and Health Survey in 2005[5], total lifetime prevalence of domestic violence was 33.5% and 8.5% for sexual violence among women aged 15–49.A 2014 study in The Lancet reports that although the reported sexual violence rate in India is among the lowest in the world, the large population of India means that the violence affects 27.5 million women over their lifetimes.However, an opinion survey among experts carried out by the Thomson Reuters Foundation ranked India as the most dangerous country in the world for women.

**2.1.2 Existing Systems and Solutions**

* Currently , SOS applications have limited features and mostly require the user to manually ping the SOS contacts.
* There is no android application that supports audio detection integrated with SOS applications.

**2.1.3 Research Findings for Existing Literature**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Roll. No.** | **Name** | **Paper Title** | **Tools and Technology** | **Findings** | **Citations** |
| 1 | 101917023  101917179 | Akshat Dogra  Kavya Jaswani | Scream And Gunshot Detection In Noisy Environments | SVM , LSTM ,AESV , features , Machine Learning | This proposed model achieved a detection rate of 93.16% and a FAR of 4.76% at a SNR of  20 dB.  Results show that the CPU  kky. | Saeed, F.S.; Bashit, A.A.;  Viswanathan, V.; Valles, D. An Initial  Machine Learning-Based Victim’s  Scream Detection Analysis for  Burning Sites. Appl. Sci. 2021, 11,  8425. https://doi.org/10.3390/  app11188425 |
| 2 | 101917014  101917125 | Ritik Rajdev  Harshita Gupta | Human Scream Detection and Analysis for Controlling Crime Rate | SVM,  Multilayer perceptrons model of Deep learning,Kivy framework | **t**he system can generate a 90% accuracy. | Ashutosh Shankhdhar,Rachit,Vinay Kumar & Yash Mathur |

**Table 2.** Research Findings for Existing Literature

**2.1.4 Problem Identified**

Currently , there exists no application that can automatically detect if the user is in danger or not. These applications require the user to manually ping the application. This approach is inconsistent as most of the time the user will not be provided the luxury to open his/her phone and then open the application to press a button.

**2.1.5 Survey of Tools and Technologies Used**

1. Machine learning.
2. Deep learning
3. Neural Network
4. SVM
5. Flutter

**2.2 Software Requirement Specification**

**2.2.1 Introduction**

**2.2.1.1 Purpose**

The purpose of this application is to decrease the rescue time by detecting noise and hence classify whether the user is in danger or not.

**2.2.1.2 Intended Audience and Reading Suggestions**

Anyone who wishes to do research in sound detection or women safety can read this article.

**2.2.1.3 Project Scope**

The project is supposed to provide software that can automatically detect if a person is in danger or not by detecting various noises and classifying them.

**2.2.2 Overall Description**

**2.2.2.1 Product Perspective**

The product is supposed to be open source , under the GNU General Public License and is based on the explicit invocation architectural style. Hifazat aims to decrease the rescue time after various incidents such as road accidents , physical violence etc by using various methods such as scream detection and location sharing.

**2.2.2.2 Product Features**

The following are the main features of Hifazat :

* Cross platform support : Offers support on widely used operating systems.
* Incident detection : Detects if a person needs help by various methods such as scream/noise detection , panic button pressing etc.
* SOS message support : If detected that the user might be in danger , the software will send SOS messages to the designated contacts on user defined platforms which will include the method of detection and location if possible.

**2.2.3 External Interface Requirements**

Not Applicable

**2.2.3.1 User Interfaces**

A graphical user interface is provided following the guidelines provided by Material UI[4]. The user is required to give the application some permissions such as to send/receive SMS messages , access the contacts list , access location etc. The user is then asked to add SOS contacts.

**2.2.3.2 Hardware Interfaces**

* Microphone
* GPS
* Cellular card

**2.2.4 Other Non-functional Requirements**

* Easy Deployment
* Efficiency
* Data Privacy
* Standard Compliance
* Security

**2.2.4.1 Performance Requirements**

Should run in background to sustain location tracking**.**

**2.2.4.2 Safety Requirements**

Not applicable

**2.2.4.3 Security Requirements**

Since we take no information from the user , there are no extra measures required for privacy.

**2.3 Cost Analysis**

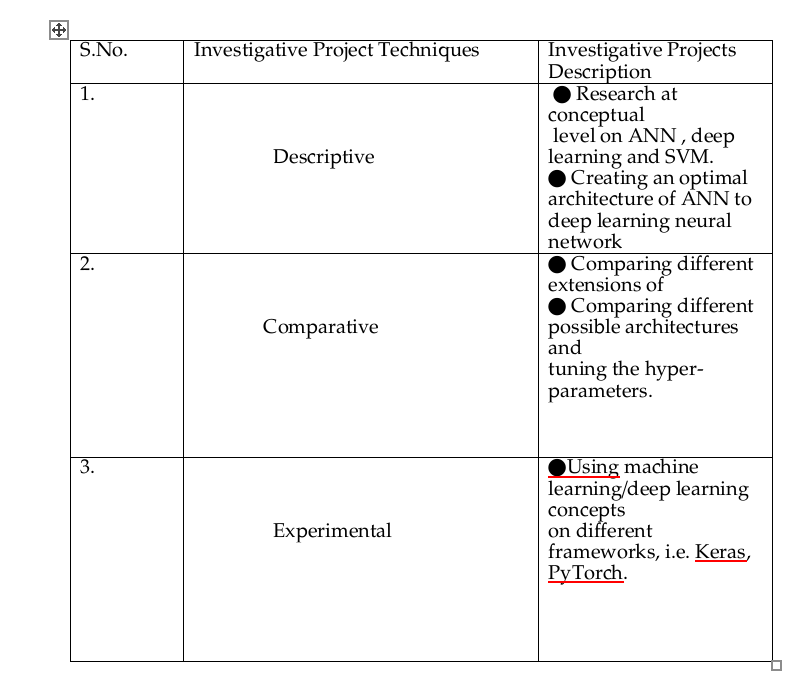
Since we are using fat client architecture we will not be needing any server and our plugins are open source so we will not be needing any monetary resources.

**2.4 Risk Analysis**

Our major goal will be to decrease the reporting of false positives , as it can create panic among the SOS contacts and will be a gesture of mistrust towards our software. Sending a SOS message on time can also pose a problem as it might take some time to send the message from the client side due to various problems with the cell provider.

**METHODOLOGY ADOPTED**

**3.1 Investigative Techniques**



**Table 3.** Investigative Techniques

**3.2 Proposed Solution**

We propose a cross-platform application that can detect noise . Since we can seldom prevent road accidents from occuring , we aim to decrease the rescue time by detecting the accident and hence decreasing the rescue time. In the case of rape or domestic violence we cannot always stop it from happening from a cell phone , but perhaps we can aware the SOS contacts or police and that can increase the chances of rescue dramatically. We can also provide the last known location to further decrease the rescue time. To detect screams , we will be using a model trained by ANN i.e artificial neural network[3] , it not only provides more accuracy than other models such as RNN or SVM , but is also more efficient. We aim to make our software as light as possible , since in India 95.84% of the smartphone users have android , majority having android version 9(31.3%) , so it is necessary to provide backwards compatibility. It is rather difficult to go below android 6 provided android 13 is about to be released in a matter of weeks , so we will be aiming to provide support till android 6. Every smartphone has a decent enough microphone to detect noise in the user's pocket or purse , so we will not be focusing much on upscaling the output efficiently. Tracking continuously with GPS is battery consuming especially in the older hardware , so many users prefer to turn off their GPS. This can be a serious problem as it can dampen the chances of rescuing on time , so rather than always keeping GPS on , we will be pinging the current location in certain intervals so as to provide a rather approximate trail , but when the scream is detected , we will surely ping the location of the user.

The working of our software is mentioned below:

● On startup, a daemon will observe the environment.

● Whenever a scream is detected, daemon will contact the app.

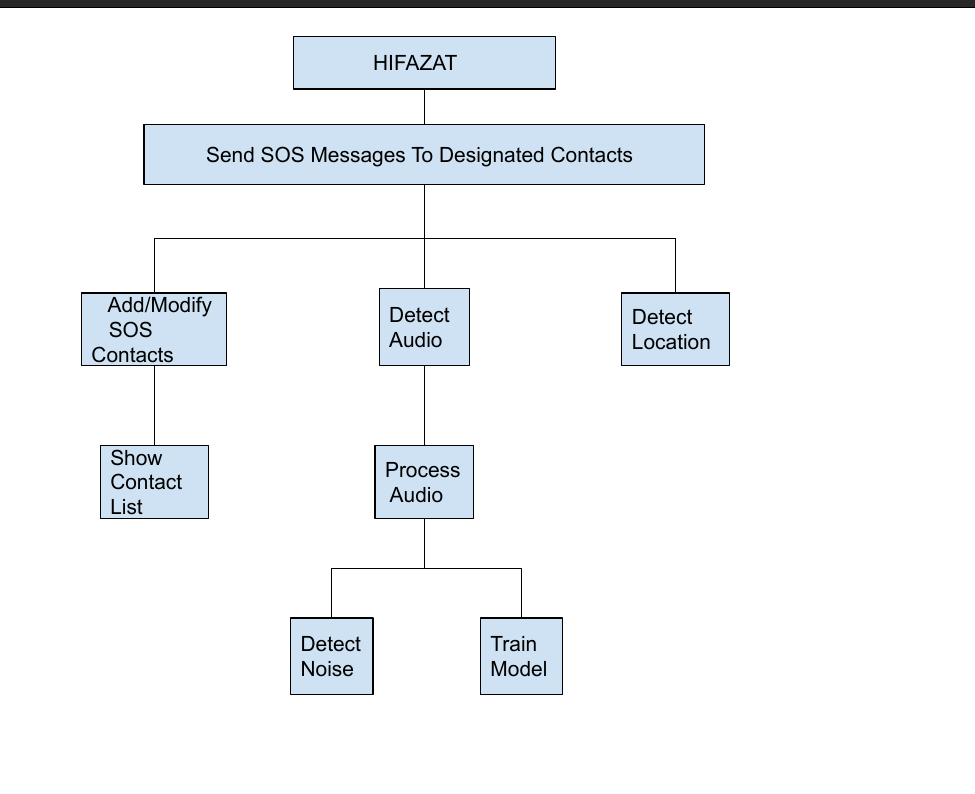
● Our app will categorize the scream on the basis of certain emotions.

● If the scream is classified as of the one in danger or hurt, this app will record a voice clip of 10 seconds.

● This voice clip is then sent to contacts appropriated by the user.

● These users can contact the Police or Ambulance or cancel the action, after listening to that voice clip, using provided buttons.

**3.3 Work Breakdown Structure**



**Figure 2 : Work Breakdown Structure**

**3.4 Tools and Technology**

1.Flutter - for Cross Platform Application building

2.Python - for building Scream Detection Model.

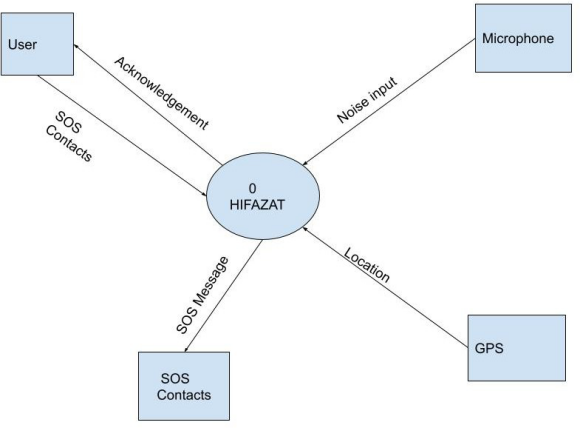
3.Neural Networks, Machine Learning for Scream Detection and Classification.

**DESIGN SPECIFICATION**

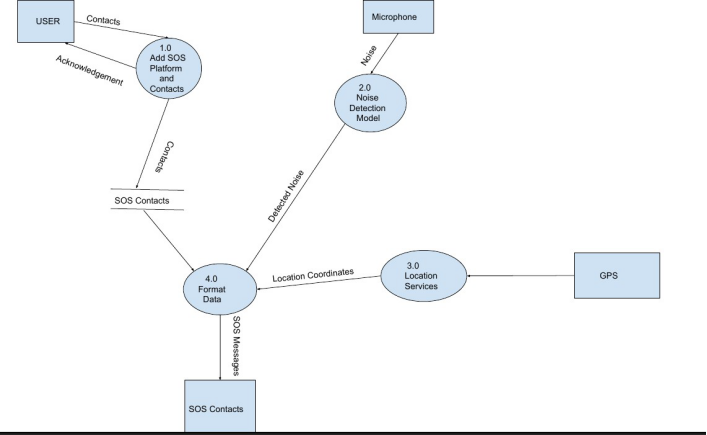
**4.1 System Architecture**

We are using Fat Client Architecture, the Server here is only used for hosting the model file which can get updated in near future. Our app will contact this server on every run and will check whether the model is updated or not, if updated it'll get the latest model. Client is Responsible for most of the work, hence Fat Client Architecture is a suitable choice for this job. Moreover, this provides additional privacy since no personal information is stored in the server. One Tier Architecture is used as well, because a File Server is used to host model file and our app contacts it to get the latest model as well.

**4.2 Design Level Diagrams**



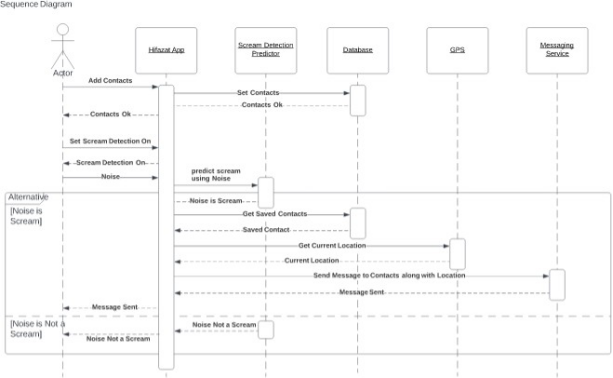
**Figure 3. Design Level Diagram of Hifazat**



**Figure 4. Design Level Diagram of Hifazat**

**4.3 User Interface Diagrams**

**Figure 5 Sequence Diagram.**



# 

# 

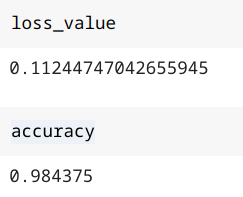
**5.5 -IMPLEMENTATION AND EXPERIMENTAL RESULTS**

**5.1 Experimental Setup**

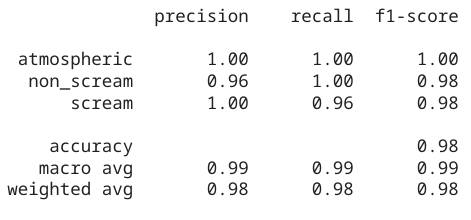
The experimental setup for this project is mostly software intensive with a little focus on various key hardware components necessary for our software to work. The study was performed in google collab and the code was written in python language as it has a wide range of library support and is excellent for scripting.The datasets were pre-processed and later divided into training and testing subsets in a similar experimental setup.

**5.2 Experimental Analysis  
 5.2.1 Data**1.Custom Model made using 3 hidden layers

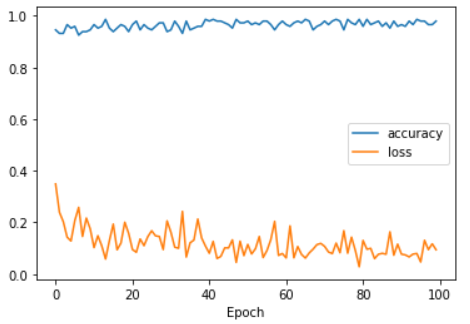
**Fig.7**



The following figure shows the precision , recall and the f1-score of our data categorized:

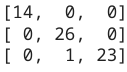
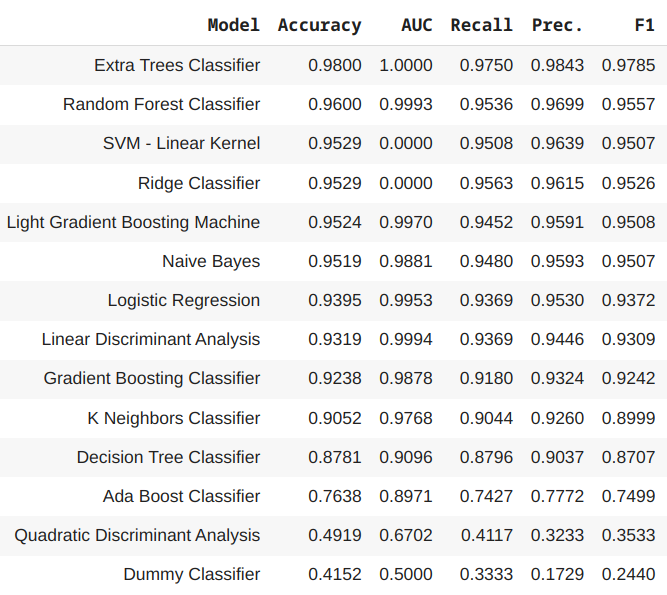


**Fig8. Precision , recall and f1- score**



**Fig.10 Epoch curve**

**Fig. 11 Different parameters tested on various Machine Learning Models**



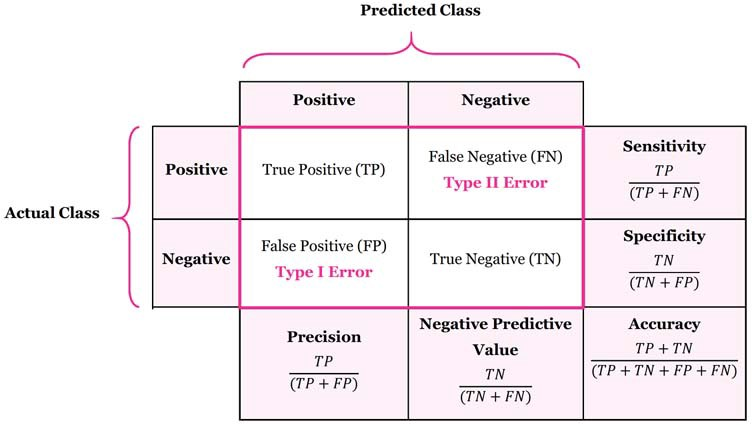
**Fig.12 Confusion Matrix  
  
 5.2.2 Performance Parameters**We have used various common and often reliable parameters such as -   
1. Accuracy: The accuracy of a ML model is a metric for determining which model is the best at distinguishing associations and trends between variables in a dataset based on the input, or training data. Overall accuracy depends on the true positive and false negative, the better the value of these two parameters, the best the results will be.   
Accuracy = Total correct predictions / Total predictions  
  
2.Precision: It is the ability of a measurement to be consistently reproduced. Basically,it is the ratio of true positives and a total number of predicted positives.  
Precision= no. of true positives/ total no. of predicted positives

Where,

Total number of predicted positives = Number of true positives + Number of false positives.

3. Recall: It is the fraction of relevant instances that were retrieved i.e the portion of positives that were identified as positive.  
Recall=TP/(TP+FN)

4.F1-score: The weighted average of precision and recall is known as the F1 score. This way, it takes both false positive and false negative into account.  
  
F1 = 2\*(Precision\*Recall)/(Precision+Recall).



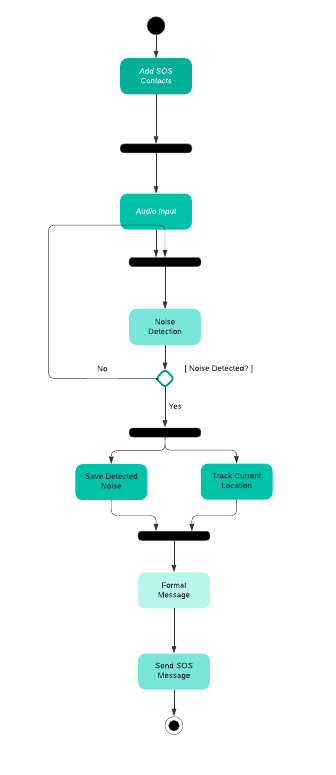
5. AUC-ROC:The Area Under the Curve (AUC) –Receiver Operating Characteristic (ROC) curve is a performance metric for classification problem at various levels. Here, the AUC represents the degree of separability, whereas ROC is a probability curve.

It basically shows how well the model can distinguish between classes, i.e., the higher the AUC, the better the model predicts if a person is in danger or not

**5.3 Working of the project**

**5.3.1 Procedural Workflow**

1. On startup, a daemon will observe the environment.
2. Whenever a scream is detected, the daemon will contact the application.
3. Our app will categorize the scream on the basis of certain emotions.
4. If the scream is classified as the one in danger or hurt, this app will record a voice clip of some seconds
5. The SOS message along the current location is then sent to contacts appropriated by the user.  
     
    **Fig 10. Block Diagram**



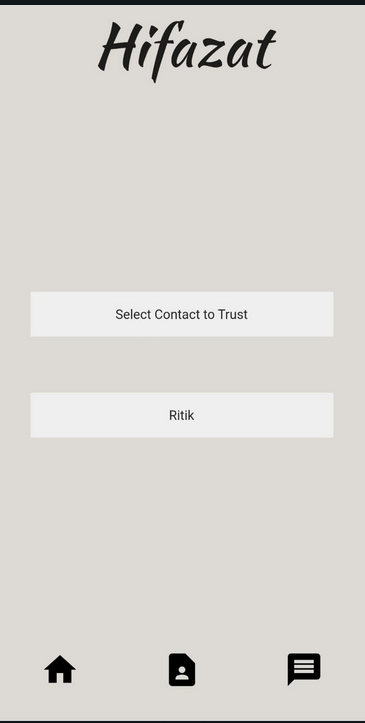
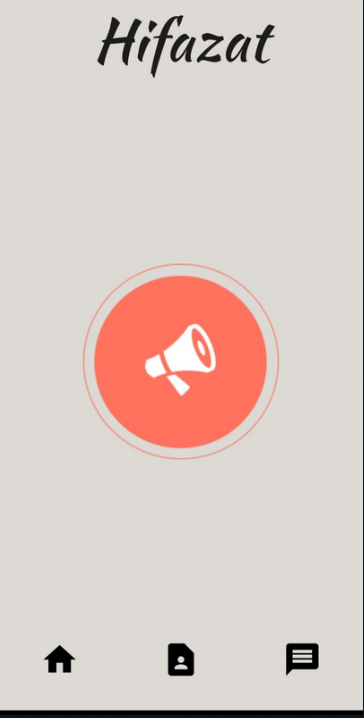
**5.3.2 Algorithmic Approaches Used**

We have used Deep Learning, Neural Networks to train our model as the accuracy we’ve got using this is astonishingly good. **5.3.3 Project Deployment**

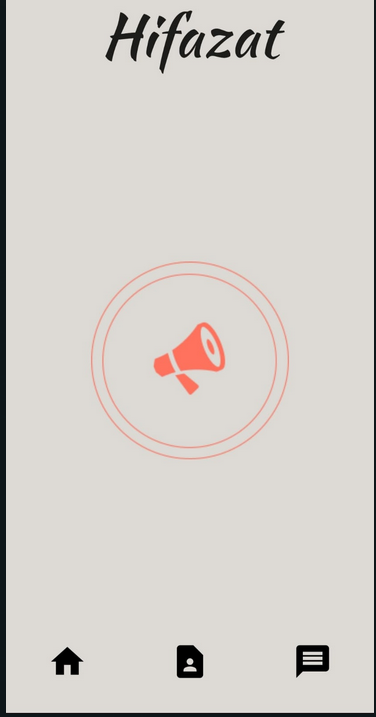
We’ve deployed our project’s backend on AWS EC2 instance for now which can be expanded on demand for larger requests as well.

**5.3.4 System Screenshots**

**Fig.11 Startup Page  
  
Fig12. Adding SOS Contacts**



**Fig.13 Manually deactivating the noise detection**



**5.4 Testing Process**

**5.4.1 Test plan**

The major testing we are required to do was server’s testing & the model’s testing. We’ve used major open source packages for flutter which are already unit tested hence there’s no requirement for unit testing functions provided by those packages. We required testing mainly of web api & our model which has been thoroughly tested.

**5.4.2 Features to be tested**

The following features were identified to be tested to make a robust software system:

1. Add Contacts

2. Send SOS message

3. Detect Scream

**5.4.3 Test strategy**

The test strategy is to validate each required feature one by one manually and thus validate the project deliverables**.** If the testing produces the desired result for each feature without freezing or taking longer times, it is considered a pass, otherwise, fail.

**5.4.4 Test techniques  
 Unit Testing:** Individual components of the software are tested in unit testing. The goal is to ensure that each component of the software works as intended. A unit is the tiniest piece of software that can be tested. It usually only has one or a few inputs and one output.

**Acceptance Testing:** Acceptance Testing is a type of software testing that examines the acceptability of a system. The goal of this test is to determine whether the system meets the functional requirements and whether it is suitable for delivery. **5.4.5 Test Cases**

Test Case#: T001  
 Subsystem: Add Contacts  
 Short Description: The user adds SOS contacts  
 Pre-Condition: The application has been given permission to access the contact list

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Action | Expected System Response | Result |
| 1. | Click the Add SOS Contacts button | Contacts show up | PASS |
| 2. | The user adds the SOS contacts | Contacts are added | PASS |
|  |  |  |  |

Test Case#: T002  
Subsystem: Send SOS message  
Short Description: SOS message is sent to the designated contacts when a noise is detected  
Pre Condition: The model detects that the user is in danger.

|  |  |
| --- | --- |
| **Step** | **Result** |
| **1.** The message has been initiated by the user side | **PASS** |
| **2.** The message is sent and received by the SOS contacts | **PASS** |

Test Case#: T003  
Subsystem: Detect Scream  
Short Description: The model detects if the user is in danger or not.

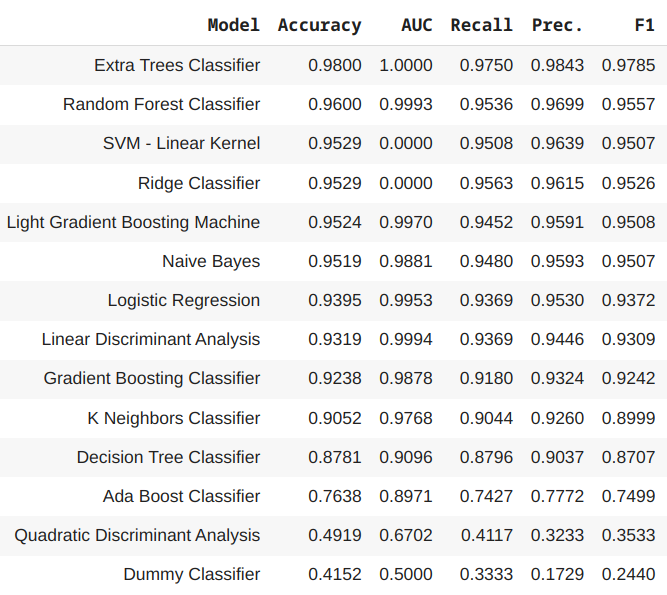
Precondition: The application has access to the device’s microphone and the microphone is working properly.

|  |  |
| --- | --- |
| Step | Result |
| 1. The scream is detected by the model | PASS |
| 2. The scream is being classified by the model | PASS |

**5.4.6 Test Results** Each feature of the system is validated and works correctly. The system works on an acceptable level and is able to make intelligent decisions and subsequently issue alerts when needed.

**5.5 Results And Discussions**Each feature of the system is validated and works correctly. The system works on an acceptable level and is able to make intelligent decisions and subsequently issue alerts when needed **5.6 Inferences Drawn**The inferences drawn from our experimental study is tabulated in the below table.

**Fig. 9  
  
  
  
  
  
  
  
  
  
5.7 Validation of Objectives**



|  |  |  |
| --- | --- | --- |
| **S. No** | **Objective** | **Status** |
| **1.** | To predict and classify if a person is in danger or not. | **SUCCESS** |
| **2.** | To compare different ML models and use the best model to assess, the better performance. | **SUCCESS** |
| **3.** | To develop a user-friendly UI. | **SUCCESS** |

**6. Conclusions and Future Scope**

**6.1 Work Accomplished**

* Trained the model with acceptable accuracy.
* Made an alpha front-end prototype.

**6.1 Conclusions**

When completed ,the application will be able to detect noise and then classify if the user is in danger or not. When classified , the application will send a message to SOS contacts providing the noise sample and probable current location.

**6.2 Social Benefits**

If implemented correctly , we can :

* Decrease casualties from road accidents.
* Decrease rape cases
* Significantly fasten the rescue time in cases like domestic violence.

**6.3 Reflections**Various skills were enhanced by working on this project as team work, time management, decision making, multitasking etc. All these reflections from the project helped us to develop personally as well as on academic grounds.

**6.4 Future Work Plan**

* Live Location tracking
* Software optimization
* Send voice clip in the SOS message

**7. Project Metrics**  
  
**7.1 Challenges Faced**

* Lack of proper support of sklearn in flutter and smartphones.
* Backwards compatibility.
* Proper references.

**7.2 Relevant Subjects**

* Machine Learning
* Neural Networks
* Flutter
* Software Engineering

**7.3 Interdisciplinary Knowledge Sharing**1.The concepts of Software Engineering were of great help in helping us decide on the functionality of the project. It helped us to choose an architecture that suits our needs best. All the diagrams use cases, data flow and others were made using the basic principles of the subject. Also, the working model and its prototype were made keeping the steps of software engineering in mind along with different stages.

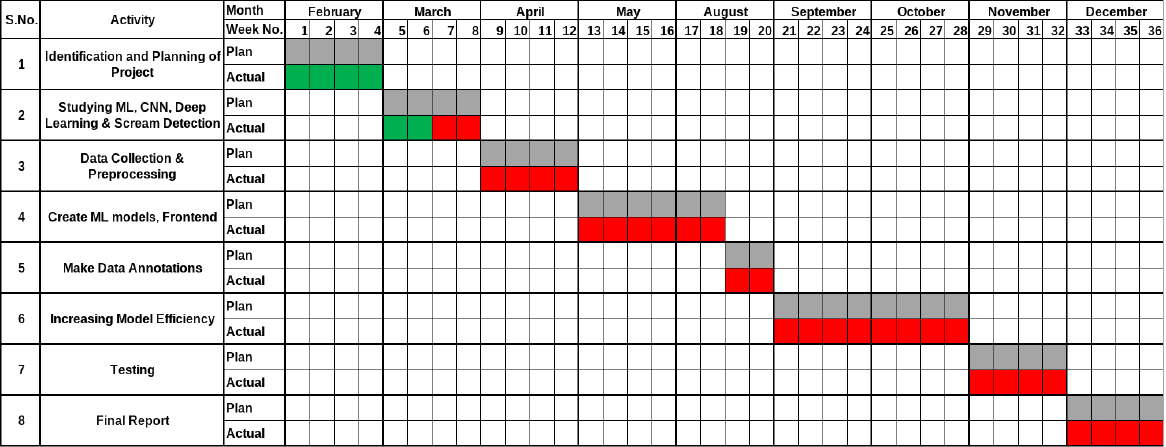
2. Using the concept of Machine Learning and Deep learning, we were able to formulate a model that is capable and trained enough to detect screams and check if a person is in danger or not.  **7.4 Peer Assessment Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Evaluation of** | | | |
| **Akshat** | **Ritik** | **Kavya** | **Harshita** |
| **Evaluation by** | **Akshat** | **-** | **5** | **5** | **5** |
| **Ritik** | **5** | **-** | **5** | **5** |
| **Kavya** | **5** | **5** | **-** | **5** |
| **Harshita** | **5** | **5** | **5** | **-** |

**Table 5.**

**7.5 Role Playing and Work Schedule**

**7.6 Student Outcomes Description and Performance   
Indicators**



|  |  |  |
| --- | --- | --- |
| **SO** | **SO Description** | **Outcome** |
| **A1** | Ability$to$identify$and formulate\_problems&related%to computational domain | Identified problem of obtaining an optimal ML classifier to accurately predict the severity of an accident, balancing between specificity and sensitivity |
| **A2** | Apply\_engineering,\_science,\_and mathematics\_body\_of\_knowledge to\_obtain\_analytical,\_numerical, and\_statistical\_solutions\_to\_solve engineering\_problems. | Experimental study on 6 ML classifiers was performed to obtain the optimal model based on their plots of Accuracy, Precision, Recall, F1-Score and ROC-AUC. |
| **A3** | Ability%to%perform experimentations%and%further analyze% the% obtained results. | Experimental study on 6 different ML classifiers was performed. |
| **A4** | Ability%to%analyze and interpret data,%make%necessary judgment(s)%and%draw conclusion(s). | The classifiers were analyzed and compared based on their accuracy, precision, recall, F1-score and ROC-AUC. |
| **A5** | Able%to%explore%and%utilize resources\_to%enhance self-learning. | A major part of the project was made possible due to appropriate utilization of self-learning platforms by the members. |

**7.7 Brief Analytical Assessment  
Q1. What sources of information did your team explore to arrive at the list of possible Project Problems?  
Ans**:The group was aware of the understanding of the capstone requirements and quickly identified a few project problems to be explored. We read IEEE research papers and journals to identify the limitations and constraints in the present systems of these shortlisted topics. Later, we had lengthy discussions with our mentor who helped us formulate these ideas.

**Q2. Did the project demand demonstration of knowledge of fundamentals, scientific and/or engineering principles? If yes, how did you apply?**

**Ans:** Yes , to complete the project, the knowledge of basic subjects learnt during previous semesters came into use like machine learning , deep learning and software engineering. In addition to this , our team also learned the building blocks to develop a good front end and the back end.

**Q3. What resources did you use to learn new materials not taught in class for the course of the project?**

**Ans:** Online free learning platforms were used to learn new concepts and technologies that fall outside the sphere of our college curriculum.

**Q4. Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?**

Ans.Working on the capstone project gave us an idea about the problems we can expect when we move to the software industry. We also learned the importance of documentation and how to collect and gather data from various sources and combine it into one. Furthermore, it made us appreciate the challenges that are bound to come in any project whatsoever be its field of domain.

**Appendix A : REFERENCES**

[1]https://en.wikipedia.org/wiki/Rape\_in\_India

[2]M.K. Nandwana, A. Ziaei, J.H. Hansen, Robust unsupervised detection of human screams in noisy acoustic environments, in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE, 2015)

[3]G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonaczci, A. Sarti, Scream and gunshot detection and localization for audio-surveillance systems, in 2007 IEEE Conference on Advanced Video and Signal Based Surveillance (IEEE, 2007)

[4] S. Ntalampiras, I. Potamitis, N. Fakotakis, On acoustic surveillance of hazardous situations, in IEEE International Conference on Acoustics, Speech and Signal Processing

[5]M.A. Sehili, B. Lecouteux, M. Vacher, F. Portet, D. Istrate, B. Dorizzi, J. Boudy, Sound environment analysis in smart home, in International Joint Conference on Ambient Intelligence (Springer, Berlin, Heidelberg, 2012)

[6] Inventive systems and control by V.Suma , Joy iong-Zong Chen,Zubair Baig, Haoxiang Wang Editors

**Appendix B: PLAGIARISM REPORT**