TDPS(Thyroid Disease Prediction System)

Revision Number: 1.0

Last date of revision: 20/01/2024

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 20/01/2024 | 1 | Initial HLD — V1.0 | K S Sujith |
|  |  |  |  |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  |  |  |  |

## Contents

[Document Version Control. 2](#_TOC_250021)

[Abstract. 4](#_TOC_250020)

1. [Introduction 5](#_TOC_250019)
   1. [Why this High-Level Design Document?. 5](#_TOC_250018)
   2. [Scope. 5](#_TOC_250017)
   3. [Definitions 5](#_TOC_250016)
2. [General Description. 6](#_TOC_250015)
   1. [Product Perspective 6](#_TOC_250014)
   2. [Problem statement 6](#_TOC_250013)
   3. [PROPOSED SOLUTION 6](#_TOC_250012)
   4. [FURTHER IMPROVEMENTS 6](#_TOC_250011)
   5. [Technical Requirements. 6](#_TOC_250010)
   6. [Data Requirements 7](#_TOC_250009)
   7. [Tools used. 8](#_TOC_250008)
      1. Hardware Requirements. 8
      2. ROS(Robotic Operating System) 9
   8. [Constraints 9](#_TOC_250007)
   9. [Assumptions. 9](#_TOC_250006)
3. [Design Details 10](#_TOC_250005)
   1. [Process Flow. 10](#_TOC_250004)
      1. [Model Training and Evaluation 10](#_TOC_250003)
      2. [Deployment Process 11](#_TOC_250002)
   2. [Event log 11](#_TOC_250001)
   3. [Error Handling 11](#_TOC_250000)
   4. Performance. 12
   5. Reusability. 12
   6. Application Compatibility 12
   7. Resource Utilization 12
   8. Deployment. 12
4. Dashboards. 13
   1. KPIs (Key Performance Indicators) 13
5. Conclusion 14

## Abstract

Recent trends are to build tall buildings in big cities as a way out of the current housing overpopulation problem. These new structures unveil problems that if not addressed in time could cause catastrophes of unimaginable impact. Some of those problems is the incidence of a fire threat happening upstairs in one of those buildings, medical emergencies due to any road accidents or mob that may cause threat to the human kind. This work discusses the implementation of the unmanned ground vehicles to spot the real location of the medical emergencies due to road mishap, mob or illegal activities such as hooliganism, snatching, robbery and the fire emergency and accordingly channelize or route them to the concerned helpline for quick mitigation and avoid disaster.

### Introduction

* 1. Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:
  + Security
  + Reliability
  + Maintainability
  + Portability
  + Reusability
  + Application compatibility
  + Resource utilization
  + Serviceability

###### Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

##### Definitions

*Term Description*

|  |  |
| --- | --- |
| *TDPS*  *Datastore IDE*  *AWS* | Thyroid Disease Prediction System |
| The Thyroid Data Sample |
| Visual Studio Code |
| Amazon Web Services |

#### General Description

##### Product Perspective

The TDPS is a disease prediction system is a Machine Learning Classification algorithm model solution for Thyroid disease prediction to help Doctors or technicians to predict a particular patient to get the chance of affecting Thyroid Disease based on different parameter provided as input to the system.

* 1. Problem statement

To create an AI solution for medical diagnosis using TDPS and to implement the following use cases.

* To predict chances of Thyroid Disease by doctors in future for the patience.
* To detect chances of Thyroid Disease by a person for prevention of it in advance in future for the patience.
  1. Proposed Solution

The solution proposed here is TDPS requires some input based on the knowledge of a patient or person. Apart from this he need to do some medical test input also. And these parameter values if we feed to the TDPS it will predict the chances of occurrence of Thyroid in future.

* 1. Further Improvements

TDPS, can we enhanced with features like the different severity of Thyroid which may affect for a person in future and can say like whether he need any doctor consultation or can be controlled by healthy habits and food control etc.

##### Technical Requirements

This document addresses the requirements for classifying people whether there any chance of Thyroid. Following are required as technical as software and hardwire.

* TDPS requires any mobile or laptop to input parameters and see the predicted results.
* AWS cloud to deploy solution with security, safety, and optimized cost.

##### Data Requirements

The details of a patience and medical test results required to input to the ML model to classify and predict the results of Thyroid. This can be loaded in Database or CSV.

###### Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn,Anaconda, ML flow, VS Code, Jupiter Notebook, AWS cloud are used to build the whole model





* Visual code is used as IDE.
* For visualization of the plots, Matplotlib, Seaborn and PyPlot are used.
* AWS is used for deployment of the model.
* Docker as container.
* MySQL/MongoDB is used to retrieve, insert, delete, and update the database.
* Front end development is done using HTML/CSS
* Flask Python for backend development.
* Anaconda for data cleaning, EDA
* GitHub is used as version control system.
  + 1. Hardware Requirements
       - Laptop or mobile devices

##### Constraints

The TDPS based medical diagnosis solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

##### Assumptions

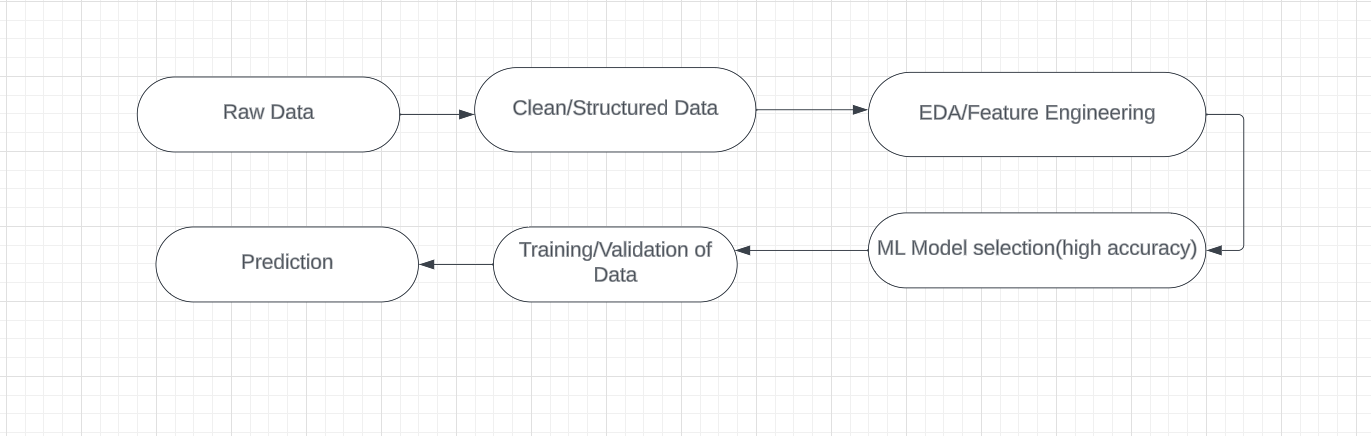
The main objective of the project is to implement the use cases as previously mentioned and the doctor or technician is collected the input parameters required.

## Design Details

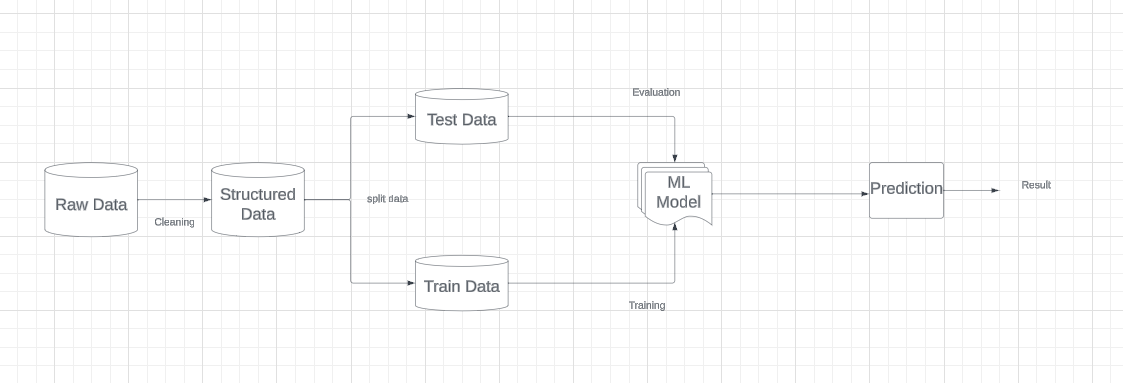
###### Process Flow

For classifying the data, we will use a machine learning base model. Below is the process flow diagram is as shown below.

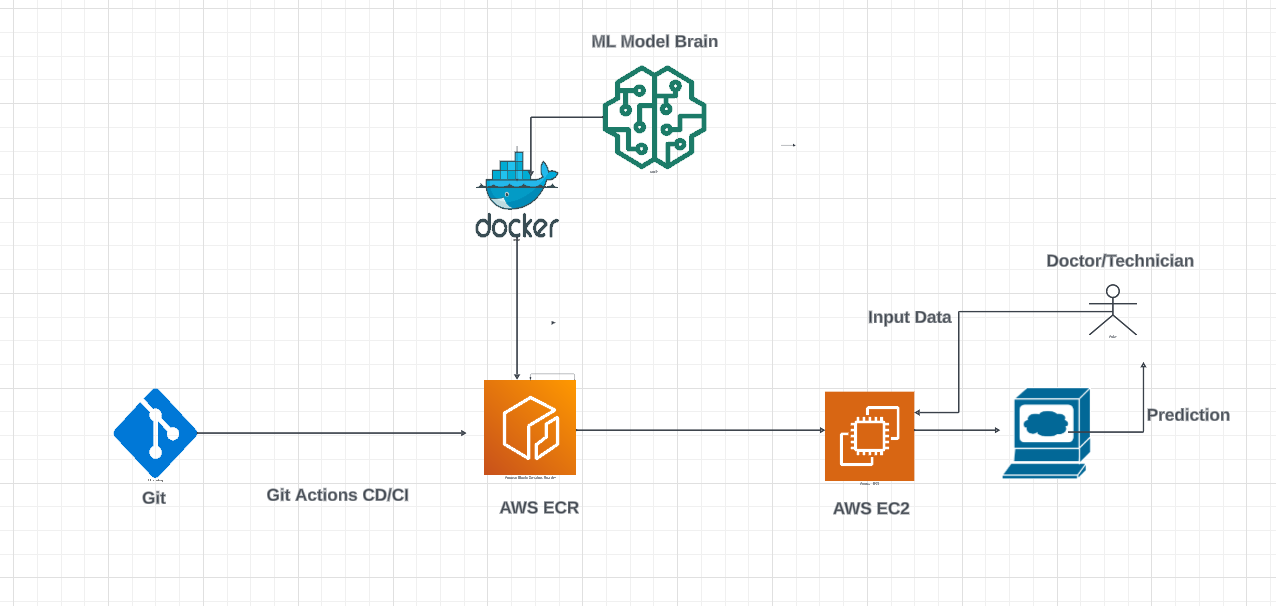
Proposed methodology



###### Model Training and Evaluation



###### Deployment Process



###### Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

##### Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

1. Performance

The UGV based surveillance solution is used for detection of anomalies in the society whenever UGV detects any anomalies (mob, medical emergency, fire, smoke, etc...) it will inform concern authorities and takes necessary action, so it should be as accurate as possible. So that it will not mislead the concern authorities (like hospitals, cops, etc..). Also, model retraining is very important to improve the performance.

##### Reusability

The code written and the components used should have the ability to be reused with no problems.

##### Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

##### Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

* 1. **Deployment**



## Dashboards

##### KPls (Key Performance Indicators)

1. Key indicators displaying a summary of the anomaly detection in the society/area.
2. Time and workload reduction using the UGV based surveillance.
3. To detect mob (illegal) activities and inform police.
4. On time alert to nearest hospital on medical emergency (accident).
5. Taking adequate evidence of mob.
6. Send disaster details to concerned authorities.
7. Display of battery life and percentage of UGV.
8. Distance travelled by UGV.
9. Get the exact location of UGV.

## Conclusion

The Designed UGV (Unmanned Ground Vehicle) will detect an anomaly in the locality based on various anomalies data used to train our algorithm, so we can identify the imbalance in the society in early stages and can take necessary action to stop them immediately, so we can have a pleasant environment in that area or location.

## References

1. https://en.wikipedia.org/wiki/Unmanned ground vehicle
2. Google.com for images of UGV hardware.
3. [https://www.ros.org/](http://www.ros.org/)