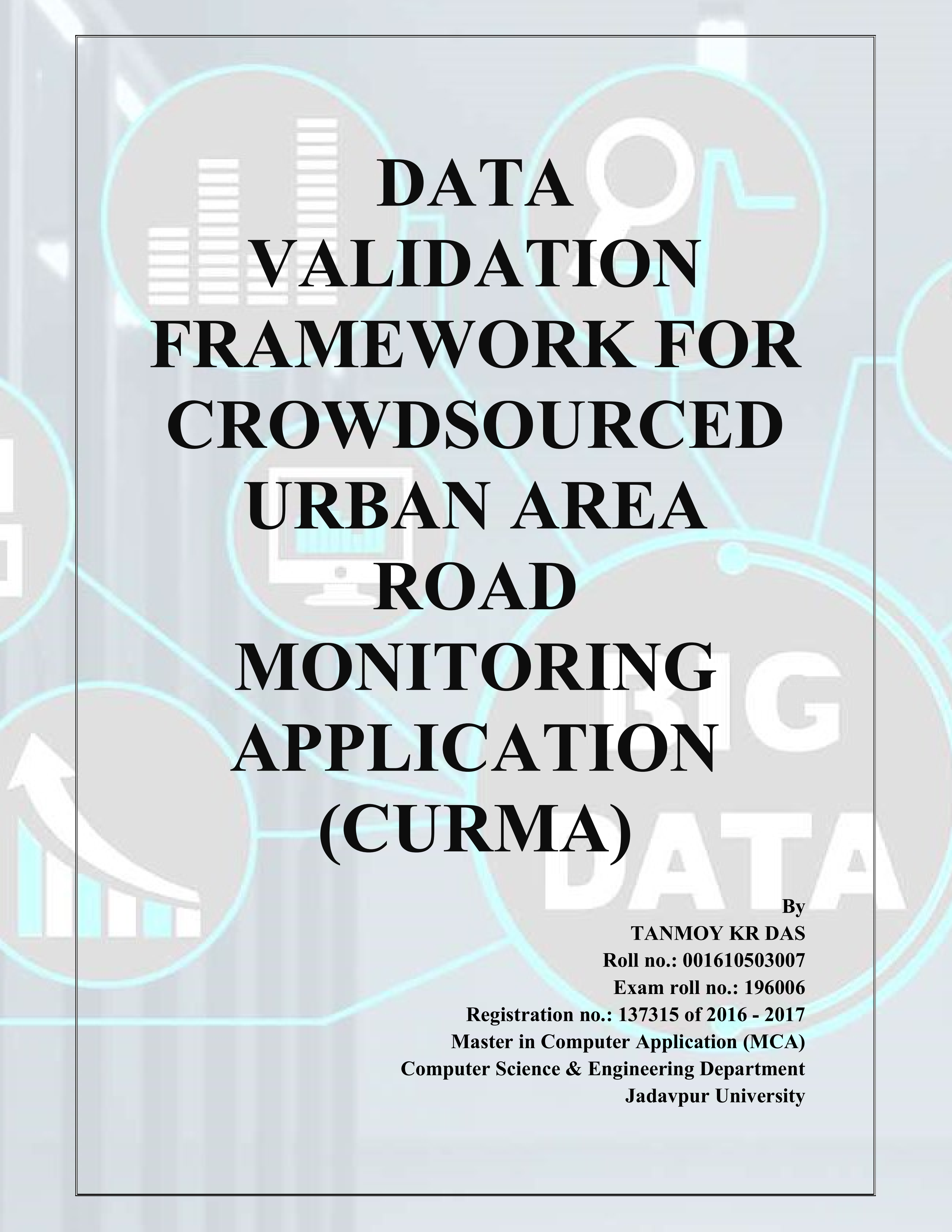


DATA VALIDATION FRAMEWORK FOR CROWDSOURCED URBAN AREA ROAD MONITORING APPLICATION (CURMA)

By
TANMOY KR DAS
Roll no.: 001610503007
Exam roll no.: 196006
Registration no.: 137315 of 2016 - 2017
Master in Computer Application (MCA)
Computer Science & Engineering Department
Jadavpur University

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the project entitled "DATA VALIDATION FRAMEWORK FOR CROWDSOURCED URBAN AREA ROAD MONITORING APPLICATION (CURMA)" has been satisfactorily completed under my guidance and supervision by **TANMOY KR DAS** (University Registration No.:137315 of 2016-17, Examination Roll No.: MCA196006, Class Roll No.: 001610503007). I hereby recommend that the project be accepted in partial fulfilment of the requirement for the Degree of Master of Computer Application, Department of Computer Science and Engineering in Faculty of Engineering and Technology, Jadavpur University, Kolkata for the academic year 2018-2019.

Dr. Chandreyee Chowdhury (Project Supervisor)
Assistant Professor
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032

Countersigned

Prof. Mahan Tapas Kundu
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-700032.

Prof. Chiranjib Bhattacharjee
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-700032.

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the project entitled “DATA VALIDATION FRAMEWORK FOR CROWDSOURCED URBAN AREA ROAD MONITORING APPLICATION (CURMA)” is a bona fide record of work carried out by TANMOY KR DAS in partial fulfilment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University during the period of February 2019 to May 2019. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for the purpose for which it has been submitted.

____

Signature of Examiner

Date:

____

Signature of Supervisor

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this project entitled "DATA VALIDATION FRAMEWORK FOR CROWDSOURCED URBAN AREA ROAD MONITORING APPLICATION (CURMA)" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Application. All information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: TANMOY KR DAS
University Registration No. : 137315 of 2016-17
Examination Roll No. : MCA196006

Project Title: **DATA VALIDATION FRAMEWORK FOR CROWDSOURCED URBAN AREA ROAD MONITORING APPLICATION (CURMA)**

Signature
Date:

ACKNOWLEDGEMENT

I express my deep sense of gratitude to my respected and learned guide, Prof. CHANDREYE CHOWDHURY for her valuable help and guidance. I am grateful to her for the encouragement she has given me in completing the project.

I convey my gratitude to our honourable HOD Prof. MAHAN TAPAS KUNDU and our curriculum coordinator Prof. DIGANTA SAHA. I would also like to thank PHD scholar Mrs. MOHONA BAKSHI for her valuable suggestions for this project.

I am also grateful to my teammate, KRITI PURKAIT for her kind cooperation and help.

TANMOY KR DAS
Roll no.: 001610503007

Contents

1. Introduction

Chapter 1: Introduction

From the title of the work, it is pretty clear that we are going to deal with some aspects of Crowdsourcing. Before we proceed, let's figure it out what is Crowd sourcing. How it came into existence, how it can be related to the field of computer science, how it can be beneficial to provide efficient services to the end users and most importantly, how it can be mapped to our work.

1.1. What is Crowdsourcing

"The term "crowdsourcing" was coined in 2005 by Jeff Howe and Mark Robinson, editors at *Wired*, to describe how businesses were using the Internet to "outsource work to the crowd", which quickly led to the *portmanteau* "crowdsourcing." Howe, first published a definition for the term crowdsourcing in a companion blog post to his June 2006 *Wired* article, "The Rise of Crowdsourcing", which came out in print just days later:[1]

Simply defined, crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers."[1]

"In a February 1, 2008, article, Daren C. Brabham, "the first [person] to publish scholarly research using the word crowdsourcing" and writer of the 2013 book, *Crowdsourcing,* defined it as an "online, distributed problem-solving and production model." Kristen L. Guth and Brabham, found that the performance of ideas offered in crowdsourcing platforms are affected not only by their quality, but also by the communication among users about the ideas, and presentation in the platform itself."[1]

"After studying more than 40 definitions of crowdsourcing in the scientific and popular literature, Enrique Estellés-Arolas and Fernando González Ladrón-de-Guevara, researchers at the Technical University of Valencia, developed a new integrating definition:

Crowdsourcing is a type of participative online activity in which an individual, an institution, a nonprofit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task; of variable complexity and modularity, and; in which the crowd should participate, bringing their work, money, knowledge [and/or] experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and use to their advantage that which the user has brought to the venture, whose form will depend on the type of activity undertaken".[1]

In other words we can say "Crowdsourcing is the practice of engaging a 'crowd' or group for a common goal — often innovation, problem solving, or efficiency. It is powered by new technologies, social media and web 2.0. Crowdsourcing can take place on many different levels and across various industries. Thanks to our growing connectivity, it is now easier than ever for individuals to collectively contribute — whether with ideas, time, expertise, or funds — to a project or cause. This collective mobilization is crowdsourcing."[2]

"This phenomenon can provide organizations with access to new ideas and solutions, deeper consumer engagement, opportunities for co-creation, optimization of tasks, and reduced costs. The Internet and social media have brought organizations closer to their stakeholders, laying the groundwork for new ways of collaborating and creating value together like never before."[2]

"Crowds are a hit. Millions of people, connected by the Internet, are contributing ideas and information to projects big and small. Crowdsourcing, as it is called, is helping to solve tricky problems and providing localized information. And with the right knowledge, contributing to the crowd — and using its wisdom — is easier than ever." -The New York Times[2]

1.2. Types of Crowdsourcing

Before starting any research, it is always beneficial to know in advance about the versatility of the field on which the research will be reflected. From a researcher point of view or from application designer point of view, classification of the core idea always gives advantages to the work and also reduces the time overheads by less searching the different applications of the core concept. Keeping this phenomenon on mind we have tried to represent the categorization of our core concept "Crowdsourcing". Let's learn how crowdsourcing can be classified on several categories based on several parameters.

1.2.1. Crowdsourcing classified based on Labor Performed (Nicholas Carr)[3]

"Crowdsourcing categorised based on the **type of labor performed** by the crowd and the way individuals in the crowd communicate and collaborate with one another:

1.2.1.1. **Social-production crowds:** This is when a large group of individuals lends their distinct talents to the creation of some product (as an example Wikipedia or Linux).

1.2.1.2. **Averaging crowds:** Provide an average judgement on some complex matter that can be, in some cases, more accurate than the judgement of any one individual (as an example the stock market)

1.2.1.3. **Data-mine crowds:** This is when a large group of people, without any knowledge of its members, produces a set of behavioral data that allows to gain insight into market patterns (as an example Ebay's or Amazon's recommendation systems).

1.2.1.4. **Networking crowds:** A group that trades information through a shared communication system such as Facebook or Twitter.

1.2.1.5. **Transactional crowds:** A group that coordinates mainly around point-to-point transactions (as an example eBay and Innocentive).

This categorization is useful as it allows understanding the different abilities crowds possess and the many ways they can work together or in isolation to perform a task."

1.2.2. Crowdsourcing based on How Various Applications Function [3]

"Another interesting approach is to categorise crowdsourcing based on **how various applications function**.

Jeff Howe used this approach to classify crowdsourcing in the following 4 categories:

1.2.2.1. **Crowd wisdom:** Using the "collective intelligence" of people within or outside an organization to solve complex problems (Innocentive is the classic example).

1.2.2.2. **Crowd creation:** Leveraging the ability and insights of a crowd of people to create new products. Since Howe's original definition this is an area that has evolved significantly and that I am following with particular interest (as an example I love Quirky's co-creation community).

1.2.2.3. **Crowd voting:** Where the community votes for their favorite idea or product (Threadless is Howe's original example).

1.2.2.4. **Crowd funding:** There is a proliferation of crowdfunding platforms in the market of different types (rewards-based such as Kickstarter and equity-based such as CrowdCube) and serving different purposes."

1.2.3. Crowdsourcing classified based on the Problem Being Solved (Daren Brabham)[3]

1.2.3.1. “**Knowledge discovery and management:** An organization tasks a crowd with finding and collecting information into a common format (examples: Peer-to-patent peertopatent.org, SeeClickFix or one recently launched by my friend Gianluca Petrelli BeMyEye). Ideal for information gathering, organization and reporting problems.

1.2.3.2. **Broadcast search:** Organizations tasks a crowd with solving empirical problems (e.g. Innocentive, Goldcorp Challenge). Ideal for ideation problems with empirical provable solutions such as scientific challenges.

1.2.3.3. **Peer-vetted creative production:** Organizations tasks a crowd with creating and selecting creative ideas. (E.g. Threadless, Doritos contest).

1.2.3.4. **Distributed human intelligence tasking:** Appropriate not to produce designs, find information, or develop solutions, but to process data. Large data problems are decomposed into small tasks requiring human intelligence, and individuals in the crowd are compensated for processing bits of data. Monetary compensation is a common motivator for participation. Amazon Mechanical Turk is the perfect example.”

1.3. Applications of Crowdsourcing

One of the first ever case of crowdsourcing is the Oxford English Dictionary. The project aimed to list all the words that enjoy any recognized lifespan in the Standard English language with their definition and explanation of usage. That was a gigantic task. So the dictionary creators invited the crowd to help them on a voluntary basis.

By now, crowdsourcing has firmly ingrained in both our societies and our economies. This paved the way for the next-level crowd sourcing sites such as Amazon Mechanical Turk, Fiverr, Upwork, TaskRabbit, 99designs. Some applications of crowdsourcing are described as follows:

1.3.1. Amazon Mechanical Turk (MTurk):

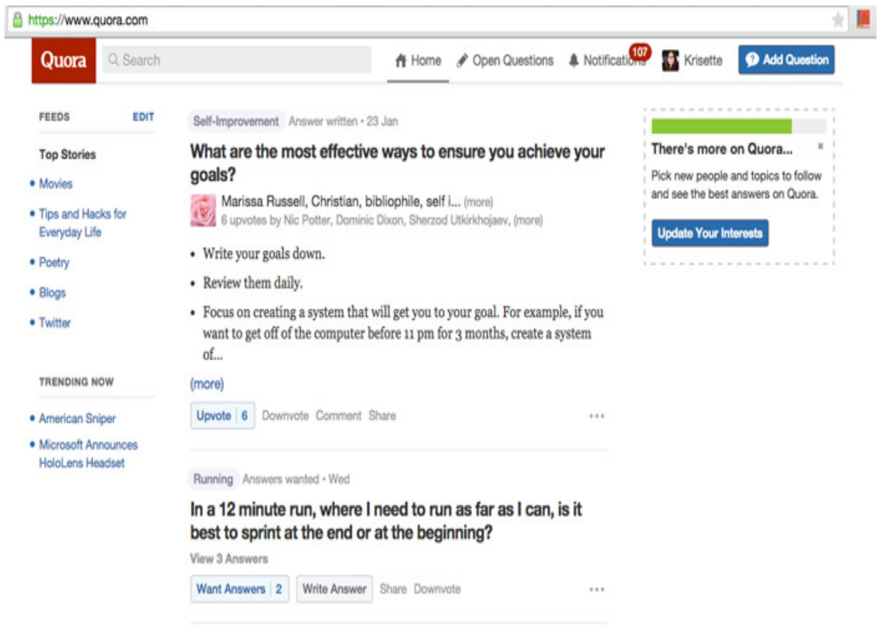
It is an Internet crowdsourcing marketplace enabling individuals and businesses (known as Requesters) to coordinate human labor to perform tasks that computer is currently unable to do. It is operated under Amazon Web Services and is owned by Amazon. Employers post jobs known as Human Intelligence Tasks (HITs), such as identifying specific content in an image or video, writing product descriptions, or answering questions, among others. Workers, colloquially known as Turkers or crowdworkers, browse among existing jobs and complete them in exchange for a rate set by the employer. To place jobs, the requesting programs use an open application programming interface (API), or the more limited MTurk Requester site.

1.3.2. **Wikipedia**: A free, web-based, multilingual and collaborative encyclopedia built on a non-for-profit business model. The platform has more than 100,000 active volunteer contributors who add new knowledge to the system daily.

1.3.3. **My Starbuck Ideas:** In 2008 Starbucks launched a crowdsourcing initiative to raise customer satisfaction levels and innovation within its outlets, called My Starbuck Ideas. A team of Idea Partners has so far read over 312,000 customer submissions. Taking in to account public votes and their level of innovation, a selection of ideas are regularly presented to Starbucks' key decision makers to figure out how to implement them.

1.3.4. Some crowdsourcing mobile apps that are shaping our digital habits

1.3.4.1. Quora:



Self-Improvement Answer written • 23 Jan

What are the most effective ways to ensure you achieve your goals?

Marissa Russell, Christian, bibliophile, self i... (more)
6 upvotes by Nic Potter, Dominic Dixon, Sherzod Utkirkhojaev, (more)

- Write your goals down.
- Review them daily.
- Focus on creating a system that will get you to your goal. For example, if you want to get off of the computer before 11 pm for 3 months, create a system of...

(more)

Running Answers wanted • Wed

In a 12 minute run, where I need to run as far as I can, is it best to sprint at the end or at the beginning?

View 3 Answers

There's more on Quora...

Pick new people and topics to follow and see the best answers on Quora.

Figure 1: Quora - A crowdsourcing mobile app

Ask, and people, enthusiasts and experts in their field, will provide you the answers. You can drop your questions and the crowd can answer them simultaneously. This open community fosters learning and encourages you to contribute as well in answering other users' questions. It comes with social media integration, so you can connect your Facebook, Twitter and other accounts to connect with other followers.[4]

1.3.4.2. Uber:

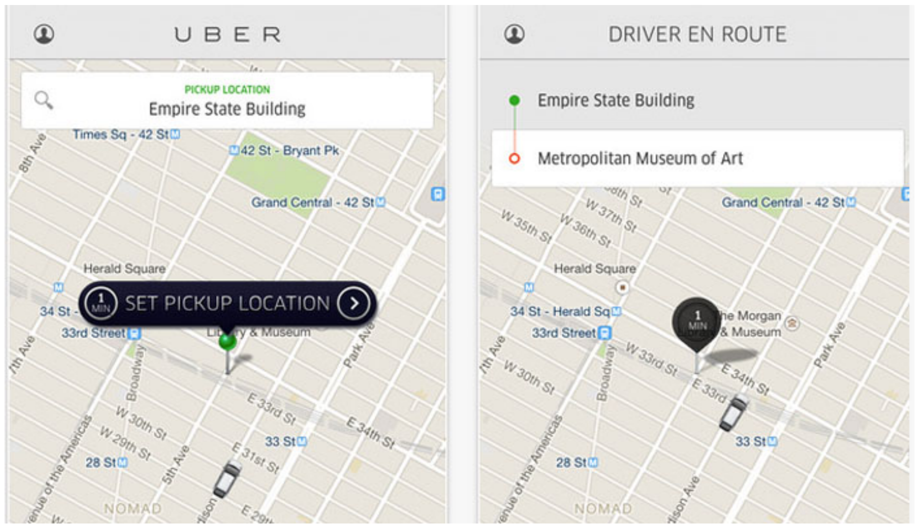


Figure 2: Uber - A crowdsourcing mobile app

Uber provides you a private car while on the go. This on-demand service is already available in over forty countries and saves your waiting time in the taxi line. The app allows riders to pay via PayPal or credit card, compare the fare rates and be picked up from their current location within a couple of minutes.[4]

1.4. Motivation

Road conditions in Indian cities have improved a lot in last few years. But still due to poor drainage system in certain places, water logging takes place during monsoon season which leads to pandemonium during busy hours with increasing traffic. Due to increasing population, the traffic condition of many cities has worsened. There are frequent traffic jams and some places are prone to accidents. Moreover though the country has taken a lot of measures for the empowerment of women but still there are roads in its city and other places that are unsafe for women at night. Another problem for people here is there are not sufficient amount of public toilets in roads and with the growing era of digitalization and internet there are very few roads in the city that have a Wi-Fi zone.

Now, often it will be beneficial for people if they come to know about the road conditions of the city i.e. whether water logging occurs or not in a particular route between places during monsoon, whether the route is safe for women at night, whether the route is prone to traffic jams or accidents or not. Keeping this present scenario in mind we have thought that crowd sourcing i.e. the common people of the city who have travelled through different routes can help others with all these information, through their feedback on different routes. Now Google maps provide us with the shortest route between any two places and recently it has added a feature which shows the current traffic jam of an area and Google Maps users can now report accidents, speed traps. But still it does not provide the places where water logging takes place or whether it is safe for women, Wi-Fi zone area and moreover what common people have to say about a particular route.

All of the above reasons led us in developing a Crowdsourced Urban Area Road Monitoring Application(CURMA).The application we have designed based helps users to get information about these particular parameters with the scope of what percentage of people have a positive or negative feedback. It is always better to take information from the common people or pedestrians who are actually using the road. And nowadays since internet is accessible to almost all so giving a feedback is not that difficult. Now there can be several other parameters like public toilets, sufficient number of dustbins etc on which feedback can be taken for the routes. Later this crowdsourced data is used to give information about the routes between areas and what is the best (optimized) route depending on the criteria that the user gives more priority on.

As we are taking feedback from the crowd, so here another question arises regarding the validity and quality of the feedbacks given by the crowd. “Are the feedbacks trustworthy enough?” – Answer to this question is very much needed as well as important because later optimized routes will come out based on the analysis of the received feedbacks using efficient algorithm. Apart from that, validation will definitely remove the noise from the database, restrict poor feedbacks from being stored into database. Validated data will always produce high quality results which will gain end users’ satisfaction while they use the application. Validated data will be less in comparison to the randomly stored data but searching time complexity will be improving by any algorithm where search space is relatively small. Search space created by validated data will be small but ensures high quality data for further processing. In other word data validation can be considered as a “Firewall” system for any crowdsourcing application. So before proceeding to the optimization, validation of the received feedback is mandatory to ensure better results along with reducing the anomalies of database. To ensure quality of the data, we have proposed a validation framework for CURMA which is further described in the “Description of CURMA” part.

1.5. Contribution

Crowdsourced Urban Area Road Monitoring Application (CURMA) is a team project. It has mainly two modules in it. First being ‘Data Validation framework for CURMA’ and second being the ‘Route optimization for CURMA’.I have done the first module. ‘Data Validation framework for CURMA’ is done using Geographic location based validation, Screen time based validation and finally followed by Machine Learning based validation. Each and every validation represents a model to get a modular structure for the Validation framework. The first two models (i.e. Geographic location based validation and Screen time based validation) are implemented through the communication between hardware and software of end user’s device. Lastly Machine Learning based validation acts as the final guard for the Validation framework. Naïve Bayes classifier is used as the machine learning tool for this purpose.

Apart from these, I have also developed the front end (i.e. design part) of CURMA. To get better and user interactive design “MapQuest API” [5] and “Leaflet Map Plug-ins”[5] are efficiently used along with the front end tools. All these will be described in “Implementation details” part in details.

1.6. Organization of this report

1.6.1. **Related Work:** It consists of some descriptions, references etc. of some previous works done on Data Validation of Crowdsourced data.

1.6.2. **Description of CURMA:** This portion of the report consists of an overview of what CURMA is and what are the modules. CURMA mainly consists of two modules namely the Validation Module and the Route Optimization module. Each of the modules again consists of two sub modules namely the Input Module and the Output Module. How continuous data flow is occurring between the input and output modules, is also described in this part.

1.6.3. **Data Validation Framework:** This part consists of the core idea of Data Validation. Why do we need Data Validation, how data are received from the end users in form of feedbacks, how various layers or phases of validation come into existence on the received feedbacks, how Machine Learning can be implemented for Validation purpose, how data can be classified using Naïve Bayes classifier as Machine Learning tool, Data Flow Diagram (DFD), small code snippets, algorithms – all these are described in this part.

1.6.4. **Implementation Details:** It consists of how the CURMA and Data Validation of CURMA were implemented and what tools were used to implement it. It consists of the screenshots of user interfaces and screenshots of real database after data validation etc. This part also describes how MapQuest API with Leaflet Map Plug-ins is used efficiently to get user interactive interfaces.

1.6.5. **Conclusion and Future Aspects:** It comprises of how CURMA can be used not only for monitoring roads but the same data validation and optimization framework can be used for other crowd sourcing applications which use users' ratings and feedbacks. The report generated by Data Validation framework for CURMA also promises that the framework can be used by any organization, government sectors to validate data collected from the crowd.

Chapter 1: Introduction

2. Related Work

Chapter 2: Related Work

With growing population and technical advancement researchers are trying to utilize the crowd to solve various complex problems. Till present day many Crowdsourcing platforms have come up like MTurk, Upwork etc. Researchers have also proposed certain algorithms for validation of data in crowdsourcing system. Let's have a look on some of the existing works relating data validation in crowdsourced system.

2.1. SwiftRiver[7]

"Shortly after a devastating earthquake struck Haiti in January, a small team of workers with Ushahidi[6], a project that enables people to crowdsource and map crisis information, started sifting through information online and mapping reports of damage, security threats, people in need of assistance, and other data. But it was very challenging for them to verify the crowdsourced data. SwiftRiver[7], a software project that grew out of Ushahidi and calls itself a "free and open source software platform that uses algorithms and crowdsourced interaction to validate and filter news."

During the first hours after the earthquake, the Ushahidi Haiti team consisted of just a few people. Valuch focused on analyzing international and Haitian media, Twitter, and Facebook to look for reports of people trapped, violence, collapsed buildings, those in need of medical attention, or other pieces of information that could explain what was happening on the ground. In the end, the Ushahidi Haiti map and information helped Marines and other responders figure out where to go to provide help, especially outside of the capital. To make this happen, the Ushahidi Haiti team had to sift through the river of reports, tweets and information and figure out which items deserved to be added to the map, and which should be discarded. In the end they decided to err on the side of inclusion and to use tags to highlight the level (or lack) of trust they had in a given piece of information.

Things became more challenging for them when they found that even though the information from Twitter is not particularly reliable and things are being retweeted. So it was kind of messy. The Ushahidi platform enables users to tag reports as "not verified" if they didn't come from a reliable source. The Ushahidi Haiti team discovered that by mapping the unverified reports, they were able to see if different sources were reporting similar things in similar areas. It was verification by aggregation. They would also attempt to verify tweets by seeing if they were retweeted by trusted sources, checking if the originating Twitter account was followed by people in Haiti, and looking to see if the user had enabled location data in their tweets.

The process wasn't perfect; but it showcases some of the techniques that can be used in crowdsourced verification. It's interesting to note how the team used a mass of unverified reports in order to achieve accuracy. Ushahidi is a map-driven project, so it chose to cluster the unverified reports in order to look for patterns, but there are other ways of collecting, analyzing, and presenting this information. The challenge is to find a way to quickly and accurately sort and evaluate a mass of incoming reports according to your preferences. This is a core element of distributed verification.

This is where SwiftRiver comes in. The big motivation behind SwiftRiver was to solve two problems Ushahidi was having. One is how to verify crowd sourced information, and two is how to filter realtime streams of data when it became overwhelming, without sacrificing the integrity of the stream. In other words, how can you speed up the process of vetting information from Twitter, RSS feeds, SMS and email.

At the core of SwiftRiver is an acknowledgement that accuracy can be a matter of perception, or situation. The tool is meant to enable people to define what accuracy means to them, and then filter based on those parameters."

2.2.Visual Tools for Crowdsourcing Data Validation Within The Globeland30 Geoportal[8]

"This research aims to investigate the role of visualization of the user generated data that can empower the geoportal of GlobeLand30 produced by NGCC (National Geomatics Center of China). The focus is set on the development of a concept of tools that can extend the Geo-tagging functionality and make use of it for different target groups. The anticipated tools should improve the continuous data validation, updating and efficient use of the remotely-sensed data distributed within GlobeLand30.

Apart from the creation of global land cover data, it is of paramount importance to communicate this information to the scientific community and to the public. Therefore, the GlobeLand30 platform is a valuable source where users can access the fine resolution GLC (Global Land Cover) data and also contribute to improvements of the data and knowledge collection. To enhance the GlobeLand30, a framework design is proposed that supports crowdsourcing data collection based on intended target audiences and involves visualization of the collected data. The collected results can give a helping hand for the validation process and the overall improvement of the system. Moreover, based on the contributed data, the users may have an opportunity to create their own personalized visualizations for efficient data understanding.

Crowdsourcing is able to assist science for identifying and solving uncertainties that occur and accumulate during data acquisition, data processing and geo-visualization. The concept of uncertainty is broad and can be caused due to several reasons (Rae et al., 2007). This article includes the concepts of accuracy and error as these components can be visualized and validated using the crowdsourcing approach.

Based on the literature review of the user requirements in the field of land cover mapping, a data structure model is defined. As a result, a workflow is established for the development of the visual tool prototypes."

2.3. A flexible framework for assessing the quality of crowdsourced data[9]

“Data collected by the crowd often lacks metadata about its quality that can lead to it being disregarded by scientists (Alabri et al. 2010), however it can frequently complement or update authoritative surveys (Jackson et al. 2010). A prevalent issue within crowdsourcing is the ability to verify and validate data collected by participants, directly contributing to the assessment of the data quality of some existing authoritative dataset (Foody and Boyd 2012). At the same time, authoritative data can be used to control the validity of volunteered information (Comber et al. 2013). An alternative to assess the quality of volunteered information is to employ experts as validators (See et al. 2013). Several methods of gaining knowledge about the quality of citizen collected data have been proposed; they include using a majority decision or control group (Hirth et al. 2012), using a reputation system (Alabri et al. 2010), (Clow et al. 2011), and using user mobility patterns with their previous quality to assess credibility of the contributed data (Mashhadi and Capra, 2011). A different approach is to attempt conflation of the citizen collected data with an authoritative source, such as OpenStreetMap and Ordnance Survey GB (OSGB) Open Data (Pourabdollah et al. 2013). Metadata about data quality plays an important role when attempting to conflate limited authoritative and crowd sourced data in regions that do not have resources to produce complete authoritative data, such as Iraq (Fairbairn et al. 2013). Analysing the ISO 19157 metadata standard, data quality can be split into two main categories: internal quality, which refers to aspects such as completeness, attribute accuracy, positional accuracy and consistency, and external quality such as fitness for use (Wang et al. 1996, Brown et al. 2012, Li et al. 2012).

The stakeholder model proposed in the introduction sits between internal and external quality as a source of uncertainty linked to the user and their device(s). If the QA/QC framework is aimed at producing metadata about spatial data quality in the form of the ISO 19157 (the producer quality model), this process requires other types of quality elements. Table 1 describes an overview of quality elements that are considered as part of the QA process, with a focus on active volunteers.

A framework is required for quality assurance to understand and improve quality in crowdsourced data, with a view to increasing the quality of the entire database over time through directed data collection and error reduction. During this process, quality metadata values for the producer model, the consumer model and the stakeholder model are derived. In a more general context, the stages for validation constituting the QA may be thought of as a series of discrete processes that could be flexibly (and iteratively) called under the control of a business process execution design that is specific to a case study but derived from generic principles. We have designed a QA process based on authoring a workflow for each type of data collected. The system is enabled by the Workflow Quality Control Authoring Tool (WoQC-AT) for chaining quality processes.

Using Automatic Validation, the data are accessed via automatic, computational techniques that apply a preliminary credibility check to the data collected. An example of employing these techniques is the OSMGB project where the aim was to check road names in OSM against the names released in the OS Open Data initiative as well as correcting the topology (Pourabdollah et al. 2013). The findings included the rate of error for OSM road labels is somewhat inversely proportional to the density of roads shown in the mapping. Validating topological relations between datasets, as a prerequisite for low level conflation has been a requirement in GIS technologies for some time. For an attribute manually input by the user, an attribute range check may relate to some obvious misunderstanding of units, as could automatic correction of spelling.

The purpose of Authoritative Comparison of QC is to compare the collected data with authoritative data sources. This stage can be used to improve the confidence and validity of collected data, add attribution, and assign error bounds to the spatial, temporal and thematic attribute of a data item. Research has focused on the user validating or updating authoritative data, e.g. (Foody et al. 2013) who describe a process where users add or change information on land cover data and Du et al. (2012), who use distributed logic to integrate crowdsourced vector road data with authoritative data. The reverse view is to use authoritative data to validate the crowdsourced observations. Therefore the final validation process takes place after the quality assessment is done and a conflated dataset produced. Some of the quality elements for the crowdsourced data depend on other data sources, controlled by reference to a time stamp (e.g., other crowdsourced data from Model-based validation). Records in the database enabling multiple representations are therefore tagged with a time and quality of real-world representation.

Model based validation of QC is focused on comparison of the crowd data with data from models or previously validated crowdsourced data. Models are likely to be environmental, but can also refer to different ways of prompting the users to harness contextual input. For environmental models it assesses the discrepancy between crowd inputs and model predictions. Validation through directing the user geographically and through feedback of potential items of interest is assessed dynamically. The principle of improving quality by real-time data feeds and corrections (Pawlowicz et al. 2011) requires a server connection, a well-designed mobile application, or an ad hoc network between devices. Data collectors in the field are acting as a team without being aware of the other team members, and are in a sense multiple sensors, used to improve accuracy of a measurement. The community of users, from the casual user to the domain expert can be used to derive a trust metric and personalise pushed tasks. Should the system know this information through a signup system, domain experts can be consulted to validate an observation if required.”

Chapter 2: Related Work

3. Description of CURMA

Chapter 3: Description of CURMA

3.1. CURMA at a glance

In Crowdsourced Urban Area Road Monitoring Application (CURMA) crowdsourced means the source of data for the application is the crowd, i.e. the common people. The term Urban Area Road Monitoring is given since the application can be used to collect data from the crowd on different routes between various places in urban area.

In CURMA we have considered the city of Kolkata as the urban area and the crowd is the people of Kolkata. Through CURMA information is collected on several road parameters like traffic condition, accident vulnerability, water log, women safety, WIFI facility, number of public toilets in the road, whether there is potholes on the road etc from the crowd as feedback and later this data collected is used to generate a report or statistics describing road conditions depending on these parameters and an optimized roué based on user given priority on these parameters.

Now the question comes why did we think of developing such an application and why have we depended on the crowd to contribute data to our application. The reason behind the thought it is that, maps like Google maps and others do not provide information about the road parameters that I have mentioned in the earlier Para and do not allow users to give priority on their choice of parameters on which they want to see an optimized route between two selected areas. We have considered the crowd i.e. the people of Kolkata as our data source because it is the people who actually live in Kolkata can give the actual information about the route conditions and in today's 21st century with the increasing availability of internet it will be easier and convenient for people to contribute data i.e. give feedback as well as easier for them to get route information.

Generally in every crowdsourcing application after data collection part there is data validation part. After data validation it is stored in a database for further analysis. For any crowd sourcing application' building data sets with the help of large group of people is the most elementary level work. The big question is that how data can be collected in an organized manner so that it can fit into specified applications. We can use web based survey tool or shared spreadsheets [2] to collect data from the crowd. Online tolls like Google forms, Survey Monkey allow us to build survey quickly and cheaply.

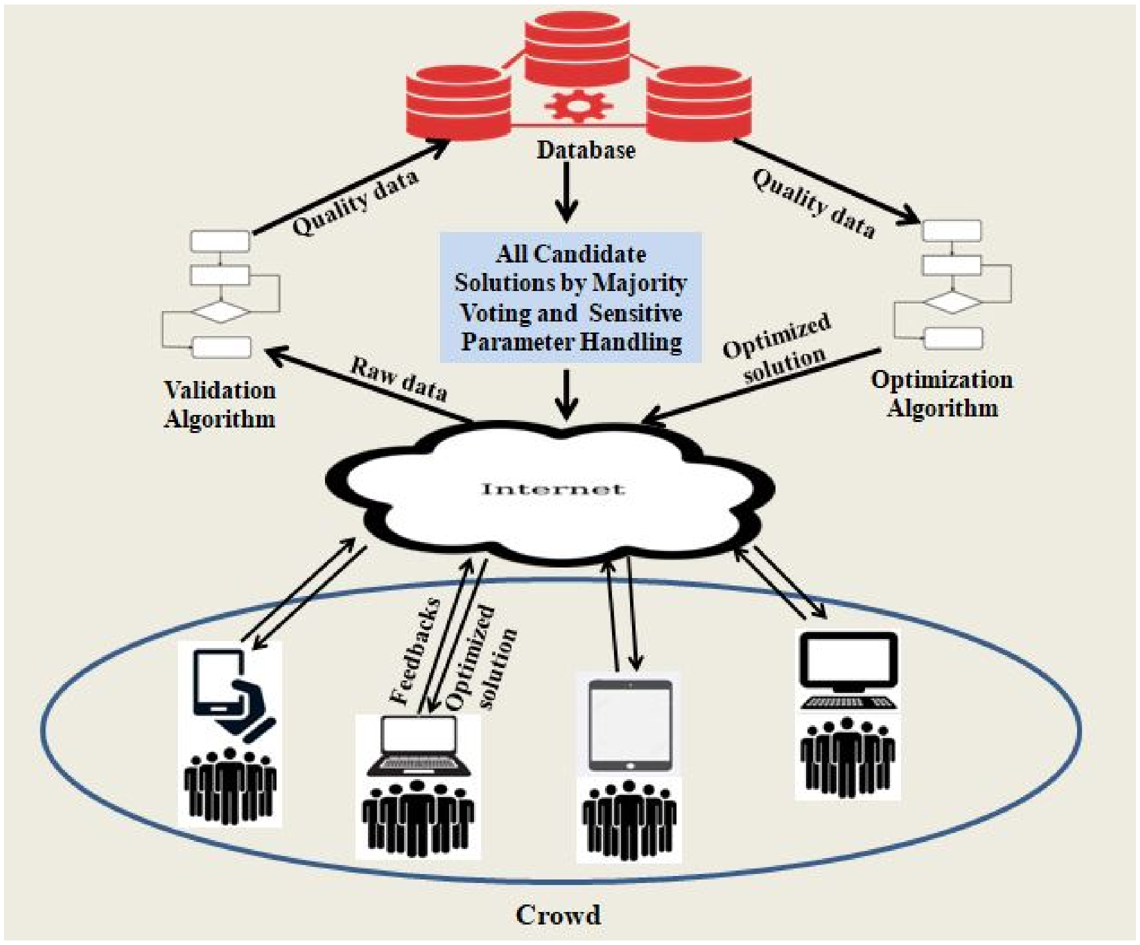


Figure 3: Proposed Workflow for Crowd Sourced Data Analysis using Pre Validation and Post Optimization

CURMA mainly has two modules namely:

- Data Validation Module
- Validated Data Optimization Module

3.2. Data Validation Module

This module again consists of the input module and the output module. The input module comprises of data collection from the crowd as feedback on different parameters like traffic condition, accident vulnerability, women safety, water log etc. on alternative routes between two user selected areas. The output module consists of the data validation. Generally collected data are in form of raw data and needed to make sense of it very quickly. Raw data can be stored into the database in tabular forms after passing through some validation process. Validation can be carried out through Automatic validation, Authoritative data comparison and Model based validation [9]. In CURMA data is validated using different three phases of validation. The first one is Geo location based validation where it is validated whether the user who is giving the feedback is actually present on the route or not. Second is the Screen Time based validation where it is checked whether the user takes the minimum required time to read the feedback page properly despite of some hardware level changes like screen turn off, document visibility changes like browser minimizing, switching to other tabs to give their feedback or not. The third and final phase of validation consists of the machine learning based validation where Naïve Bayes classifier is used to classify data as valid and finally store into the database.

3.3.Validated Data Optimization Module

This module again consists of the input module and the output module. The input module comprises of data collection in the form of user priority on the route parameters like traffic condition, women safety etc. The output module consists of the generation and display of all the candidate solutions i.e. the route statistics of each of the alternative routes containing the percentage of users saying about the different conditions of the routes. Now for users convenience optimization of the routes is done based on user priority using an optimization algorithm. Optimization algorithms are used to narrow down a set of candidate solutions to an optimum solution. Among the optimization algorithms we have chosen (GA) Genetic Algorithm to find out the optimal i.e. the best route according to user priority.

4. Data Validation Framework

Chapter 4: Data Validation Framework

4.1. What is Data Validation

“In computer science, data validation is the process of ensuring data has undergone data cleansing to ensure they have data quality, that is, that they are both correct and useful. It uses routines, often called "validation rules" "validation constraints" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system. The rules may be implemented through the automated facilities of a data dictionary, or by the inclusion of explicit application program validation logic.” [10]

In other words, “Data validation means checking the accuracy and quality of source data before using, importing or otherwise processing data. Different types of validation can be performed depending on destination constraints or objectives. Data validation is a form of data cleansing.” [11]

4.2. Why do we need Data Validation

According to general concept, when moving and merging data it’s important to make sure data from different sources and repositories will conform to business rules and not become corrupted due to inconsistencies in type or context. The goal is to create data that is consistent, accurate and complete so to prevent data loss and errors during a move [11]. Let’s figure it out why do we need Data Validation in CURMA

As CURMA is a crowdsourced data based web application, so crowd/ end users (i.e. people of Kolkata) are allowed to give their feedback on several routes between any two places of Kolkata metropolitan city. From a practical view point, any crowdsourced application handles huge amount of data from the end users. So here definitely a question arises that “Are the collected data trustworthy enough for further processing?” And another question relates this – “If the data are not trustworthy enough then how to sort out the valid data from the collected raw data or how to validate them before further processing?”

Before answering these above mentioned two questions, let’s consider a situation where collected raw data is directly used for further processing. Then what are the difficulties we are going to face while optimizing the data? Our optimization algorithm will find an optimized solution in term of a route among n (n = 3) routes between any two chosen places of Kolkata city using the data stored in the database. So it is impossible for our optimization algorithm to check for data validity before optimizing. It will find a solution based on stored data and if the data stored in the database are not trustworthy enough then definitely there will be a huge impact on the result returned by the optimization algorithm. And of course below standard result may create bad impression to the end users. It will not give guarantee for end users’ satisfaction. So to return a better result using optimization, data pre processing (i.e. validation) is a mandatory task.

Now let’s answer the above mentioned two questions. To check the trustworthiness of raw data, it should be passed through some validation checking to measure the degree of data validity. At each step of validation checking, raw data will be given some weight value (if it passes the validation test) or 0 weight value (if it fails the validation test). After all the validation checking, then we can measure the trustworthiness of the raw data using their score. Based on their score, data will be given some tags (i.e. degree of validation) and low quality data will be restricted from being stored in the database.

After completion of the validation process, data stored in the database are said to be the reliable for further processing. Optimization using the reliable and quality data ensures tremendous improvement in the results as well as guarantees end users’ satisfaction.

4.3. Data Validation in CURMA

For CURMA, we have done validation and quality checking in three phases. Reason behind using three phased validation is that a user’s feedback (i.e. raw data) can be considered as valid one even though it fails in one or two phases but manages to pass the last phase of validation. In other words, data failed in one or two phases may get a chance to be stored in the database if it passes the last phase. For our work three phases of validation are:

- Geographic Location based validation
- Screen Time based validation
- Machine Learning Classifier based validation

4.3.1. Geographic Location based validation

As CURMA is an Urban Area Road Monitoring web application, so multiple routes (i.e. collection of roads) between any two chosen places of Kolkata metropolitan area are represented at the user interface part using a third party API service “MapQuest APT with Leaflet Map Plug-in”. This API service is achieved using HTTP get request to MapQuest server in form of XML object.

Code snippets:

```
var request = new XMLHttpRequest();

request.open("GET","https://www.mapquestapi.com/directions/v2/alternaterou
tes?key=m30Kp3RqyiTmSjAjScGtiQnb7S0sboTm&from="+source1+"%2C+Kol
kata%2C+West+Bengal%2C+India&to="+dest1+"%2C+Kolkata%2C+West+Ben
gal%2C+India&outFormat=xml&ambiguities=ignore&routeType=fastest&maxR
outes=3&timeOverage=25&doReverseGeocode=false&enhancedNarrative=true&
avoidTimedConditions=false&unit=M", false);

    request.send();
    var xml = request.responseXML;
```

4.3.1.1. Getting all latitudes and longitudes of all routes

Once the XML object is available, then it is possible to get all the latitudes and longitudes of all the routes represented by that XML object. All the latitudes and longitudes from the XML object are put into suitable data structure after parsing the XML object.

Code snippets:

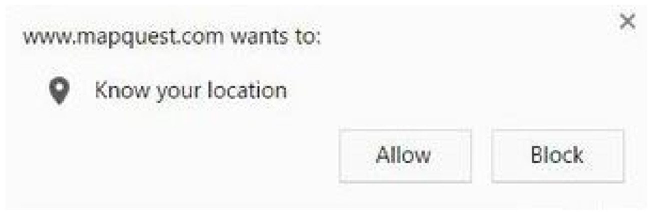
```
var locs = xml.getElementsByTagName("startPoint");

for(var i = 0; i < locs.length; i++)
{
        var loc = locs[i];
        var lat = loc.getElementsByTagName("lat");
        for(var j = 0; j < lat.length; j++)
        {
                lat_loc.push(lat[j].childNodes[0].textContent);
                row1++;
        }
        var lang = loc.getElementsByTagName("lng");
        for(var j = 0; j < lang.length; j++)
        {
                lang_loc.push(lang[j].childNodes[0].textContent);
                row2++;
        }

}
```

4.3.1.2. Fetching user device's latitude and longitude

Now at the user's interface, user will be requested to give access to his/ her respective device's location (Figure 4). If the user permits then at the back end respective device's location (i.e. latitude and longitude) will be fetched.


www.mapquest.com wants to:

Know your location

Figure 4: Seeking user's permission to access device's location

Code snippets:

```
function geoFindMe()
{
    function success(position)
        {
        latitude = position.coords.latitude.toPrecision(6);
        longitude = position.coords.longitude.toPrecision(6);

    }

    function error()
   {
        alert("Location can not be accessed");
    }

   navigator.geolocation.getCurrentPosition(success, error);
}
```

4.3.1.3. Search for user device’s latitude and longitude in the list of all latitudes and longitudes

After that, there will be a search operation to find respective device’s latitude and longitude among the previously listed latitudes and longitudes. If any combination from the listed latitudes and longitudes matches with the device’s latitude and longitude, then we can consider that user’s feedback is going to pass the Geographic Location based validation. It means user is currently at the route for which user is giving feedback.

Code snippet:

```
for(var i=0; i < row1; i++)
   {
        if(lat_loc[i] == latitude)
        {
            flag1 = 1;
```

```
                break;
            }
    }
    for(var j=0; j < row2; j++)
    {
            if(lang_loc[j] == longitude)
            {
                flag2 = 1;
                break;
            }
    }
    if(flag1 == 1 && flag2 == 1)
            validation_signal = 1;
```

4.3.1.4. Initialising weight value to feedback after the completion of Geographic Location based validation

To indicate the pass or fail at Geographic Location based validation, user's feedback will be given a weight value which will be used later to calculate total validity fitness value of a user given feedback.

- If a feedback passes the Geographic Location based validation then it will be given 20 as weight value.
- If a feedback fails the Geographic Location based validation then it will be given 0 as weight value.

4.3.2. Screen Time based validation

After completing the Geographic Location based validation, then user's data has to face Screen Time based validation. This part is an interesting one but a little bit complicated to implement. Let's try to understand the core concept behind this.

Let's consider some online activity where a user has to use his/ her presence of mind and intelligence to give feedback or data related to some topics. In such scenarios, "the time taken by a user to read and understand the topic visible on a specific tab, on which feedback will be given" should be taken under consideration. A cross judgemental question can simplify the concept – "Is the user spending sufficient amount of time to read the concept visible on the opened tab of his/ her device?"

From the question, two things are very much clear and they will lead us to the Screen Time based validation. The two things are:

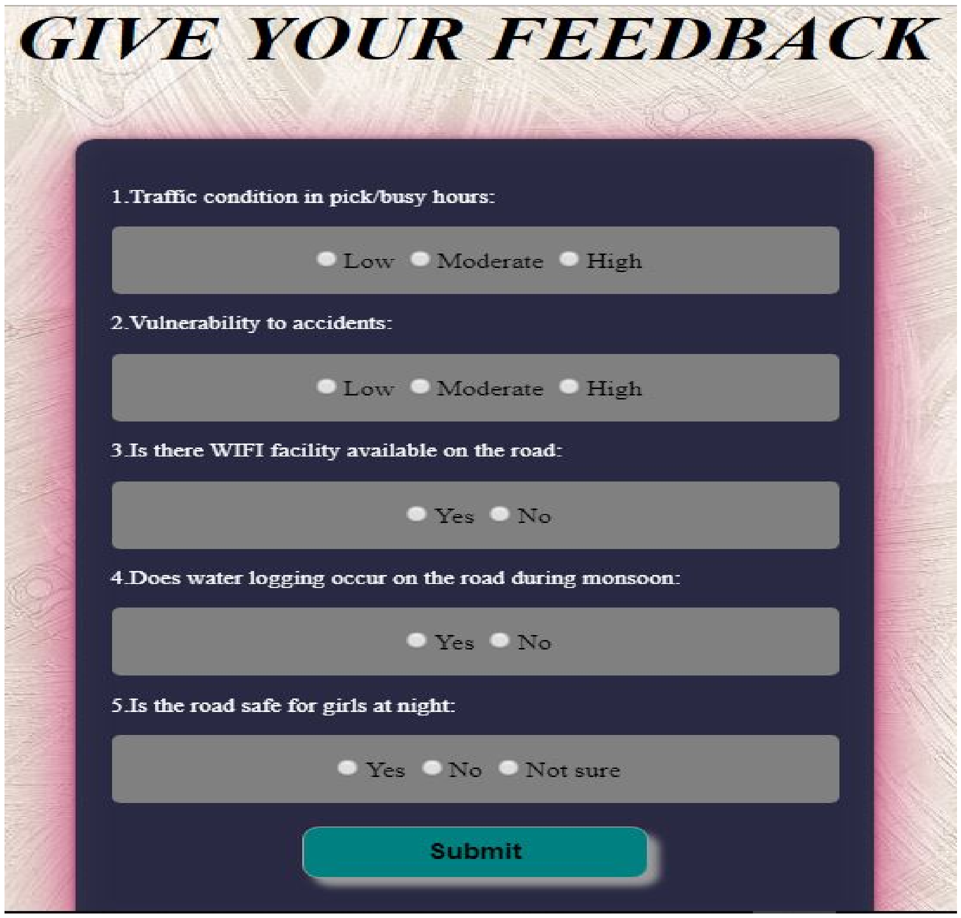
- How much time will be the sufficient time to read something on a tab during any online activity

- How to measure that a user is spending sufficient time or more than sufficient time on the tab while giving the feedback

Before calculating the sufficient time, we need to know about the average reading time of a grown up person. A full grown up person can read about 200 to 250 words in a minute in non technical material [12]. But for technical material scenario is quite different. For technical material word counts reduces to 70 to 75 per minute [12]. Now from this data we can roughly estimate that how much time a user should spend on the tab while giving the feedback in our designed user interface.

4.3.2.1. Calculation of Sufficient Reading Time

Let's have a look on how our user interface tab looks like. We are going to calculate the sufficient reading time using the total number of words contained by it.



GIVE YOUR FEEDBACK

1.Traffic condition in pick/busy hours:

○ Low ○ Moderate ○ High

2.Vulnerability to accidents:

○ Low ○ Moderate ○ High

3.Is there WIFI facility available on the road:

○ Yes ○ No

4.Does water logging occur on the road during monsoon:

○ Yes ○ No

5.Is the road safe for girls at night:

○ Yes ○ No ○ Not sure

Submit

Figure 5: User interface of Feedback form module

Form the user interface we can find the total number of words in the tab is 50 (counting every word). So using the above facts (mentioned at the last paragraph of section 4.3.2.), minimum time required to read the content of the tab is merely 15 seconds. Now with this value we can add some adjustment factors like page loading time, sudden change in bandwidth etc. Let's consider the adjustment factor is not more than 5 seconds.

So the sufficient reading time of this tab will be 15 + 5 = 20 seconds.

4.3.2.2. Measuring the time spent by the user on a specific tab

Now the big question comes and it is how we check whether a user is spending that sufficient amount of time (mentioned at section 4.3.2.1.) for reading the content before submitting the feedback. Finding the time spent by a user on a specific tab is a little bit complicated and requires lots of calculation like total time spent on the tab, time wasted in any minimize or resizing of tab occurs, time wasted in screen turning off etc.

4.3.2.2.1. Measuring the total time spent by the user on a specific tab

First we will calculate the total time spent on the tab by user. Later we will adjust it by other additional calculations. To find total time spent on the tab we have used a timer count. Timer count will start incrementing by 1 per second once the tab finishes its loading at client side. When the user clicks to go the next module (i.e. clicking on Submit button), then only timer count stops and it will return the total time spent on the page at the server side. We have used following code snippet for calculating total time spent on the page.

Code snippets:

```
page_time_total = setTimeout("total_time_on_page()",1000);
```

4.3.2.2.2. Measuring the time wasted in any additional event

Once we get the total time spent on the page, then we will do some additional calculation to find the wasted time due to any of the following events occurs:

- Minimizing or resizing the specific tab.
- Sudden turning off of the screen.
- Switching to any other tab before completing the feedback.

To calculate the time wasted in each of the above mentioned scenarios, we have used some timer counts to calculate the time spent in all those events.

Code snippet for calculating time wasted in tab minimizing, tab switching, and screen turning off:

```
function total_time_on_page2()
{
     minimize_time = setTimeout("total_time_on_page2()",1000);
}
function minimize_and_screenoff_time()
{
document.addEventListener("visibilitychange", function() {
  flag++;
  if(flag%2!=0)
      {
              total_time_on_page2();
      }
}, false);

function getTime() {
    return (new Date()).getTime();
}

var lastInterval = getTime();

function intervalHeartbeat() {
    var now = getTime();
    var diff = now - lastInterval;
    screen_off_time = diff - 1000; // 1000 = the 1 second delay I was
expecting
    lastInterval = now;

    if(screen_off_time > 100) { // don't trigger on small stutters less than
100ms
        //console.log('interval heartbeat - off by ' + offBy + 'ms');
              //alert(screen_off_time/1000);
    }
}

setInterval(intervalHeartbeat, 1000);
}
```

Once we get the time wasted in all those events, then we have used following calculation to get the actual time spent by the user on the specific tab:

Actual time spent on tab= Total time spent on tab – Time wasted in tab minimizing – Time wasted in tab switching – Time wasted in screen turn off

Code snippets:

```
page_visible_time = page_time_total - minimize_time - screen_off_time/1000;
```

4.3.2.3. Comparing the time spent on the tab with the sufficient time for reading the content

After getting time spent on the tab and sufficient time for reading the content of the tab, then we can compare the time spent on the tab with the sufficient time for reading the content. There could be three possible scenarios.

- If time spent on the tab is less than the sufficient time, then we will consider that the feedback has failed the screen time validation.
- If time spent on the tab is equal to the sufficient time, then we will consider that the feedback has passed the screen time validation.
- If time spent on the tab is greater than the sufficient time, then we will consider that the feedback has passed the screen time validation.

4.3.2.4. Assigning weight value to feedback after the completion of Screen Time based validation

To indicate the pass or fail at Screen Time based validation, user's feedback will be given a weight value which will be added to the previously given weight value (mentioned at section 4.3.1.4.) and it will be used later to calculate total validity fitness value of a user given feedback.

- If a feedback passes the Screen Time based validation then it will be given 20 as weight value.
- If a feedback fails the Screen Time based validation then it will be given 0 as weight value.

4.3.3. Machine Learning Classifier based validation

After completing the Geographic Location based validation and Screen Time based validation, user's feedback has to face Machine Learning Classifier based validation. We can consider Machine Learning Classifier based validation as the "final guard of the castle" for our work. For our work, we are discarding a feedback if it fails to pass the Machine Learning Classifier based validation. That means a feedback will be allowed to be stored in the database only when it passes the Machine Learning Classifier based validation. Then definitely a question arises – "What are the significances of Geographic Location based validation and Screen Time based validation when we are considering Machine Learning Classifier based validation as our prior method?" This will be discussed at section 4.3.3.3.3. To be more precise about this topic, we must say that we are using PHP as our back end tool for database connectivity, optimization and classifier for data validation using machine learning. There are several classifier and clustering methods available in PHP machine learning library. We have used Naïve Bayes classifier for data validation. Let's figure why we have chosen Naïve Bayes classifier.

4.3.3.1. How good is Naïve Bayes classifier as ML tool[13]

"The first assumption of a Naive Bayes classifier is that the value of a particular feature is independent of the value of any other feature. Which means that the interdependencies within data are comfortably neglected? Hence the name is 'Naive.'

A naive Bayes classifier considers every feature to contribute independently to the probability irrespective of the correlations.

For unsupervised or in more practical scenarios, maximum likelihood is the method used by naive Bayes model in order to avoid any Bayesian methods, which are good in supervised setting."

4.3.3.2. How Naïve Bayes Classifier is used as ML tool[14]

"Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as **Maximum A Posteriori (MAP)**.

The MAP for a hypothesis is:

$$\begin{aligned}&\boldsymbol{MAP}(\boldsymbol{H})\\&= \boldsymbol{max}(\boldsymbol{P}(\boldsymbol{H}|\boldsymbol{E}))\\&= \boldsymbol{max}((\boldsymbol{P}(\boldsymbol{E}|\boldsymbol{H}) * \boldsymbol{P}(\boldsymbol{H}))/\boldsymbol{P}(\boldsymbol{E}))\\&= \boldsymbol{max}(\boldsymbol{P}(\boldsymbol{E}|\boldsymbol{H}) * \boldsymbol{P}(\boldsymbol{H}))\end{aligned}$$

P(E) is evidence probability, and it is used to normalize the result. It remains same so, removing it won't affect.

Naive Bayes classifier assumes that all the features are **unrelated** to each other. Presence or absence of a feature does not influence the presence or absence of any other feature. We can use Wikipedia example for explaining the logic i.e., A fruit may be considered to be an apple if it is red, round, and about 4″ in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

In real datasets, we test a hypothesis given multiple evidence (feature). So, calculations become complicated. To simplify the work, the feature independence approach is used to 'uncouple' multiple evidence and treat each as an independent one."

$$\boldsymbol{P(H|Multiple\ Evidences) = P(E1|\ H) * P(E2|H) \ldots\ldots * P(En|H) * P(H) \,/\, P(Multiple\ Evidences)}$$

4.3.3.3. Naïve Bayes Classifier for CURMA

According to the traditional approach, some training data sets will be provided to the Naïve Bayes classifier and the system will be learned using these training data sets. Each of the training data set will have some class value. Later some test data sets will be provided to the classifier and using the training data sets, classifier will predict the class value of the test data sets. For our work we have used the traditional approach. First we provide some training data sets to our classifier along with their class value. Then some test data sets will provided to classifier to get their class values by the prediction by the classifier.

4.3.3.3.1. Training data sets for CURMA

Let's consider a scenario where a user chooses to give feedback on a route between two chosen places (say A and B) of Kolkata city. Now we use Naïve Bayes classifier for validation of user feedback then first we need to create a training data set for this route with a class value.

As you can see there are five fields (mentioned at Figure 1) in our feedback form module. And there are some radio buttons available for each of the fields. According to the flow of our work, now user can give values (i.e. user ratings) of 5 specific parameters of a route between two places A and B. We are going to store the values in tabular form (Figure 2) using MYSQL RDBMS after validation. To understand the structure of training data set it is necessary to understand how a complete feedback will look like once it will be stored in the database.

Here each row of the MYSQL table will represent a complete feedback accepted from a user on a particular route in between A and B. Each route in between A and B will be uniquely identified by a key. The key value satisfying the criteria of RDBMS will be used for further mapping with the output module and displaying all candidate solutions. In our application we are receiving the values of the parameters in form of user ratings like “High”, “Moderate”, “Low” or “Yes/ No/ Not sure” etc. Now before storing into the database, these ratings will be encoded to some integer values as follows:

Ratings	**Corresponding Integer Value**
High	3
Moderate	2
Low	1
Yes	1
No	2
Not sure	3

Table 1: Rating Conversion

Encoding the values of parameters in above mentioned manner (Table 2) requires specific domain range information. So it will be advantageous if the domain range of each parameter is made limited and small in size. In our application, we have used the domain range from “Low” to “High” for parameters like “Traffic Condition”, and “Accident Vulnerability”. But parameters like “Wi-Fi Facility”, “Water Log” and “Women Safety” have different types of range as “Yes” or “No” or “Not sure”. Assignment of domain range can be application specific or parameter specific or some other specifications based. In our application, “Traffic Condition” and “Accident Vulnerability” are ordinal attributes which is a classification of qualitative attribute. In accordance with the nature of both parameters, we have defined their possible values as “Low”, “Moderate”, and “High” with corresponding integer encoding as 1, 2, and 3 respectively. In the other hand “Wi-Fi Facility” and “Water Log” are asymmetric binary attributes which is also a classification of qualitative attribute. So we have set limitations on their values to be either “Yes” or “No” with corresponding integer encoding as 1 and 2 respectively. Let us consider, a user has chosen a route between two places A and B and provided some feedbacks for some parameters like “Traffic Condition” as “High”, “Accident Vulnerability” as “Low”, “Wi-Fi Facility” as “No”, “Water Log” as “Yes” “Women Safety” as “No” etc.

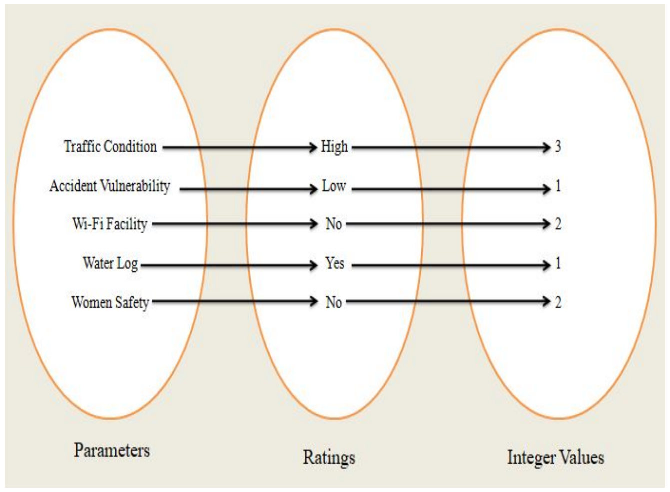


Figure 6: Mapping of Sample User Ratings into Integer Values

Once we are done with the structure of a sample record in database, then we can create our training sets for particular route between A and B on which user chooses to give feedback. From the above explanation, it is pretty clear that our training data sets will also containing some integer values respective of some ideal values of the parameters used in the feedback form for a specific route. It is not necessary to define values of all the parameters in the training set. If we do that then it will be most specific scenario for a training set according to Naïve Bayes classifier. So we can define ideal values for some of the parameters and leave rest of the fields for the acceptance of most general scenarios. In other words, we can use some ground truths as our training sets for a particular route.

4.3.3.3.2. Ground Truth in CURMA

Let us consider a route between two places (say Howrah and Saltlake) of Kolkata city. We are going to define a ground truth for this route. Let's consider the route id is HowrahSaltlake1. We can find that there are two fields named "Traffic condition during peak/ busy hour" and "Does water logging occurs during monsoon" at the feedback module (Figure 1). Now if we consider the ideal and actual scenarios for these two fields of HowrahSaltlake1, then traffic condition will always be high at busy hour and water logging occurs during monsoon. So according to the actual scenario our ground truth will be [3, 1] by using integer conversion of user rating (Figure 2). Here "3" is denoting "High" and "1" is denoting "Yes".

4.3.3.3.3. Defining class value for our training datasets

Once we get the ground truth, then we can assign class value to it. Let's assume we are assigning a class value 60 to the defined ground truth mentioned in above example (section 4.3.3.3.2). Choosing the class value for training data set is some kind of application specific. It could be any integer value or any significant string or character etc. We have chosen integer value for our work.

One thing is clear from the above methodology and it is – "When a user feedback carries the values of the two chosen parameters similar to the values of the ground truth then the class value of the user feedback will be predicted as 60". But what if any user feedback carries values of those parameters unmatched to the ground truth? According to the actual scenario any feedback carries values of those parameters unmatched to the ground truth should be considered as "Invalid" feedback and should not be stored in the database. To ensure this thing, we have to define more training sets for the opposite scenarios like this.

All possible combinations using all possible values (i.e. [1, 1], [1, 2], [3, 2], [2, 1], [2, 2]) of the chosen parameters should be defined as training sets along with their class values. For the above mentioned scenario, we have assigned class value as 0 for all other combinations apart from the ideal combination (mentioned at section 4.3.3.3.2.). Thus the way we have defined our training data sets for Naïve Bayes classifier.

Code snippets:

```
$samples = [[3, 1], [1, 1], [1, 2], [3, 2], [2, 1], [2, 2]];
$labels = [60, 0, 0, 0, 0, 0, 0];
```

4.3.3.3.4. Predicting class value for our test datasets

Once we have defined our training data sets, then we can send user's feedback to the classifier to predict the class value. According to the training sets (mentioned at section 4.3.3.3.3.), a user's feedback may get either 60 or 0 as its class value.

Code snippets:

```
$classifier = new NaiveBayes();
$classifier->train($samples, $labels);
```

4.3.3.4. Assigning weight value to feedback after the completion of Machine Learning Classifier based validation

After achieving the class value, user's feedback will be given a weight value equal to its class value which will be added to the previously given weight value (mentioned at section 4.3.1.4. and 4.3.2.4.) and it will be used later to calculate total validity fitness value of a user given feedback.

- If a feedback passes the Machine Learning Classifier based validation then it will be given 60 as weight value.
- If a feedback fails the Machine Learning Classifier based validation then it will be given 0 as weight value.

4.3.4. Calculating total validity fitness value

After facing all the above mentioned validations, finally total validity fitness value will be calculated for a user's feedback using all the weight values assigned to it. Following formula is used to calculate total validity fitness value:

$$\begin{aligned}&\boldsymbol{Total\ validity\ fitness\ value}\\&=\sum(\boldsymbol{Weight\ value\ achieved\ at\ each\ phase\ of\ validation})\end{aligned}$$

Code snippets:

```
$total_validation_weight=$classifier>predict([$radio1,$radio4])+$loc_validation_
weight+$time_validation_weight;
```

4.3.5. Storing of feedback in the database based on validation result

Once a user's feedback faces all the validations, then definitely it is carrying some fitness values by summing up all the weight values achieved at each phase of validation. Based on the fitness it will be decided whether the feedback is fit enough to be stored in the database or to be discarded. Before proceeding into the decision making part, let's have a look on all possible combination of fitness value achieved by summing up the weight values after each phase of validation.

Geographic Location based validation Status	Screen Time based validation Status	Machine Learning Classifier based validation Status	Total validity fitness value
PASS	PASS	PASS	20+20+60 = 100
PASS	FAIL	PASS	20+0+60 = 80
FAIL	PASS	PASS	0+20+60 = 80
FAIL	FAIL	PASS	0+0+60 = 60
PASS	PASS	FAIL	20+20+0 = 40
PASS	FAIL	FAIL	20+0+0 = 20
FAIL	PASS	FAIL	0+20+0 = 20
FAIL	FAIL	FAIL	0+0+0 = 0

Table 2: All possible combinations of Total validity fitness value

Now question is that what should be the minimum fitness value for a feedback to be stored in the database. We have considered the minimum fitness value as 60. The reason behind choosing 60 as minimum fitness value is that a user feedback has to pass the Machine Learning Classifier based validation. Let's consider the 5th scenario from the Table 2. In this combination, both the Geographic Location based validation and Screen Time based validations have been passed but Machine Learning Classifier based validation failed. So the above combination is getting only 40 which is a below standard value according to our considered minimum value (i.e. 60). From the combinations at Table 2, it is very clear that no feedback will be allowed to be stored in the database unless and until it is passing the Machine Learning Classifier based validation. So all possible fitness values for which feedbacks can be stored in the database are 100, 80 and 60. Now question is that how we store them in the database based on their validity fitness value?

Valid and fit feedbacks will be stored in the database with some specified tags. All these tags will be indicating their total validity fitness value. Let's consider the following table:

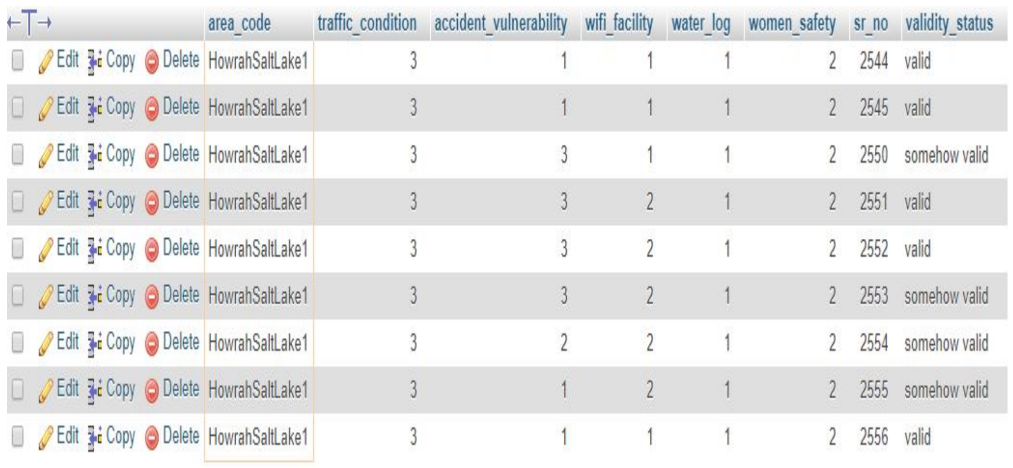
Total validity fitness value	Tag used in the database
100	"highly valid"
80	"valid"
60	"somehow valid"

Table 3: Specified tags based on validity fitness value

From the Table 3, it is pretty clear that if a feedback passes all the three validations then it will be considered as "highly valid" feedback. If a feedback only one of the first two validations (i.e. Geographic Location based validation and Screen Time based validation) and Machine Learning Classifier based validation then it will be considered as "valid". And lastly, if a feedback passes only Machine Learning Classifier based validation and fails the first two validations then it will be considered as "somehow valid". Feedbacks will be stored in the database with their respective integer values after user rating to integer conversion (Figure 6) along with the tags decided by their validity fitness values.

Code snippets:

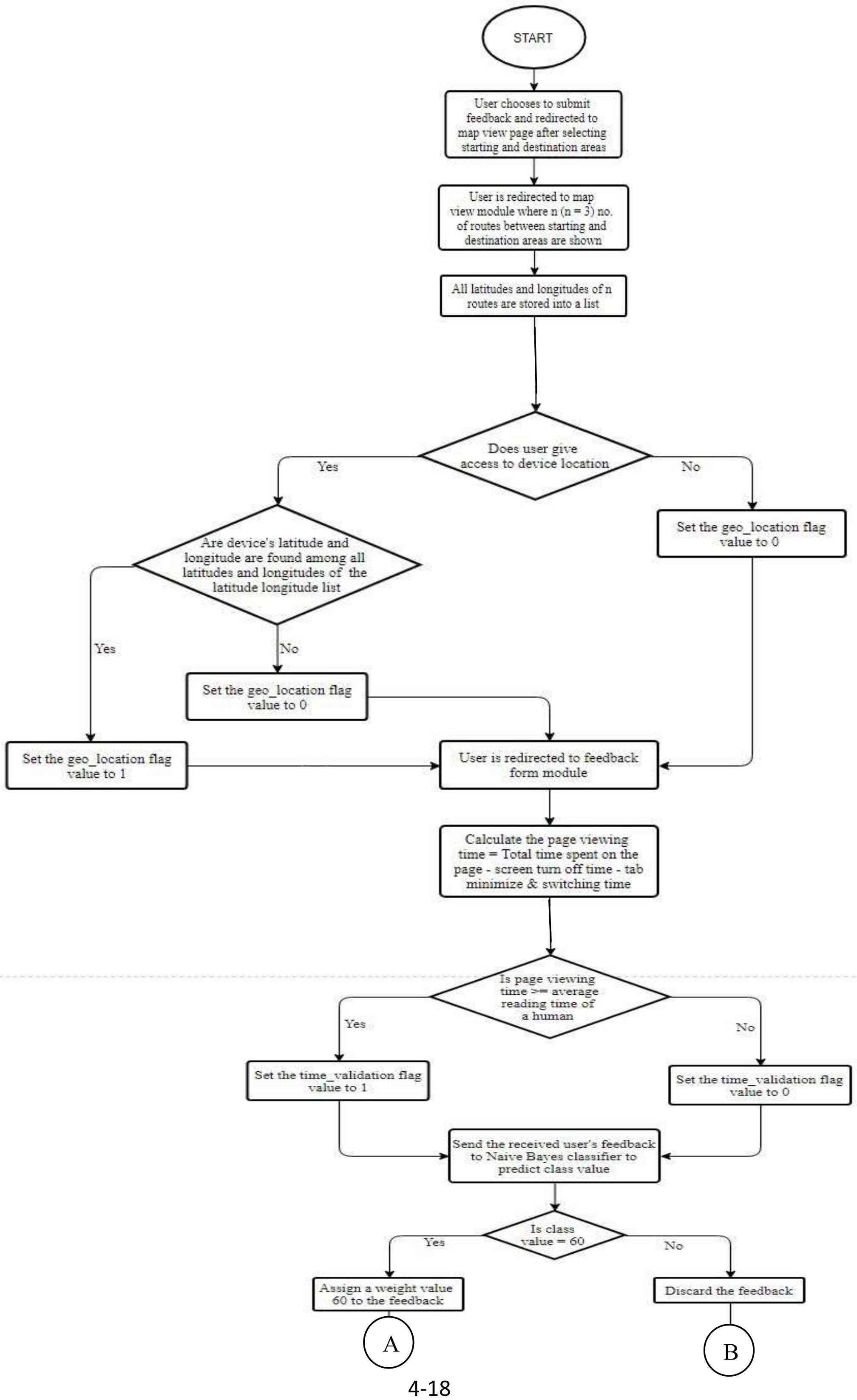
```
if($total_validation_weight==100)
     $q="insert into
feedback(area_code,traffic_condition,accident_vulnerability,wifi_facility,water_lo
g,women_safety,validity_status)
values('$area_code','$radio1','$radio2','$radio3','$radio4','$radio5','highly valid')";
     if($total_validation_weight==80)
     $q="insert into
feedback(area_code,traffic_condition,accident_vulnerability,wifi_facility,water_lo
g,women_safety,validity_status)
values('$area_code','$radio1','$radio2','$radio3','$radio4','$radio5','valid')";
     if($total_validation_weight==60)
     $q="insert into
feedback(area_code,traffic_condition,accident_vulnerability,wifi_facility,water_lo
g,women_safety,validity_status)
values('$area_code','$radio1','$radio2','$radio3','$radio4','$radio5','somehow
valid')";
     $s=mysqli_query($dc,$q);
```

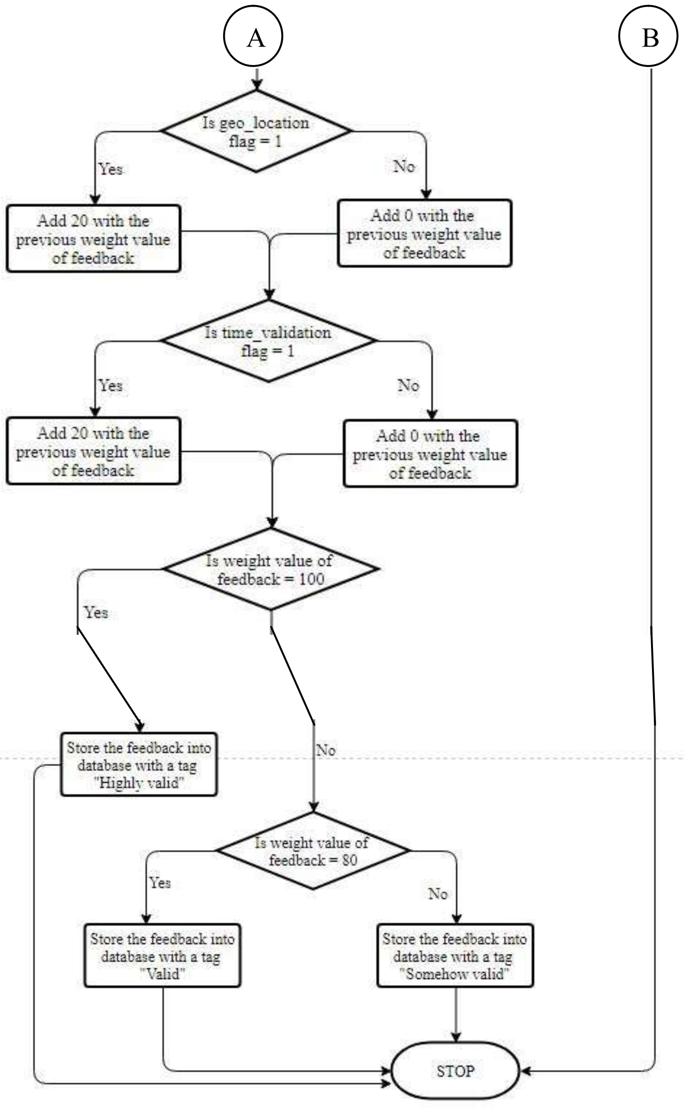


←T→	area_code	traffic_condition	accident_vulnerability	wifi_facility	water_log	women_safety	sr_no	validity_status
Edit Copy Delete	HowrahSaltLake1	3	1	1	1	2	2544	valid
Edit Copy Delete	HowrahSaltLake1	3	1	1	1	2	2545	valid
Edit Copy Delete	HowrahSaltLake1	3	3	1	1	2	2550	somehow valid
Edit Copy Delete	HowrahSaltLake1	3	3	2	1	2	2551	valid
Edit Copy Delete	HowrahSaltLake1	3	3	2	1	2	2552	valid
Edit Copy Delete	HowrahSaltLake1	3	3	2	1	2	2553	somehow valid
Edit Copy Delete	HowrahSaltLake1	3	2	2	1	2	2554	somehow valid
Edit Copy Delete	HowrahSaltLake1	3	1	2	1	2	2555	somehow valid
Edit Copy Delete	HowrahSaltLake1	3	1	1	1	2	2556	valid

Figure 7: Real Database scenario after validation

4.3.6. Integrated Flow Chart of data validation in CURMA


START
User chooses to submit feedback and redirected to map view page after selecting starting and destination areas
User is redirected to map view module where n (n = 3) no. of routes between starting and destination areas are shown
All latitudes and longitudes of n routes are stored into a list
Does user give access to device location
Yes
No
Set the geo_location flag value to 0
Are device's latitude and longitude are found among all latitudes and longitudes of the latitude longitude list
Yes
No
Set the geo_location flag value to 0
Set the geo_location flag value to 1
User is redirected to feedback form module
Calculate the page viewing time = Total time spent on the page - screen turn off time - tab minimize & switching time
Is page viewing time >= average reading time of a human
Yes
No
Set the time_validation flag value to 1
Set the time_validation flag value to 0
Send the received user's feedback to Naive Bayes classifier to predict class value
Is class value = 60
Yes
No
Assign a weight value 60 to the feedback
Discard the feedback
A
B


A
B
Is geo_location flag = 1
Yes
No
Add 20 with the previous weight value of feedback
Add 0 with the previous weight value of feedback
Is time_validation flag = 1
Yes
No
Add 20 with the previous weight value of feedback
Add 0 with the previous weight value of feedback
Is weight value of feedback = 100
Yes
No
Store the feedback into database with a tag "Highly valid"
Is weight value of feedback = 80
Yes
No
Store the feedback into database with a tag "Valid"
Store the feedback into database with a tag "Somehow valid"
STOP

4.3.7. Data comparison graph of CURMA

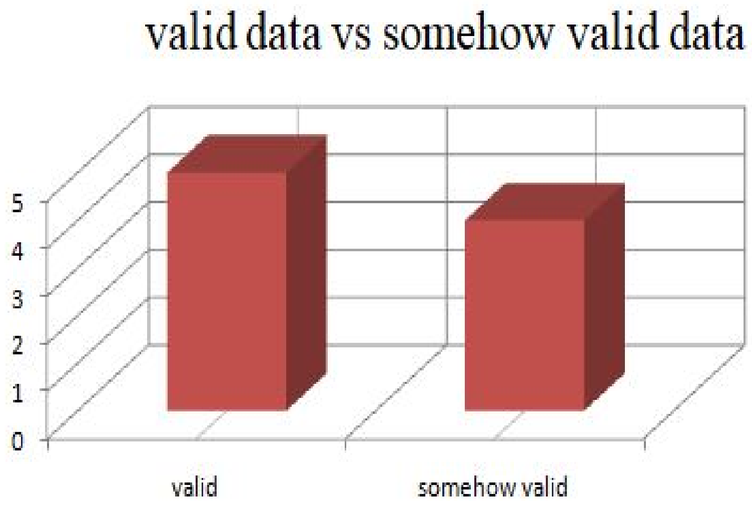


Figure 8: Graphical view_1 of various degrees of valid data

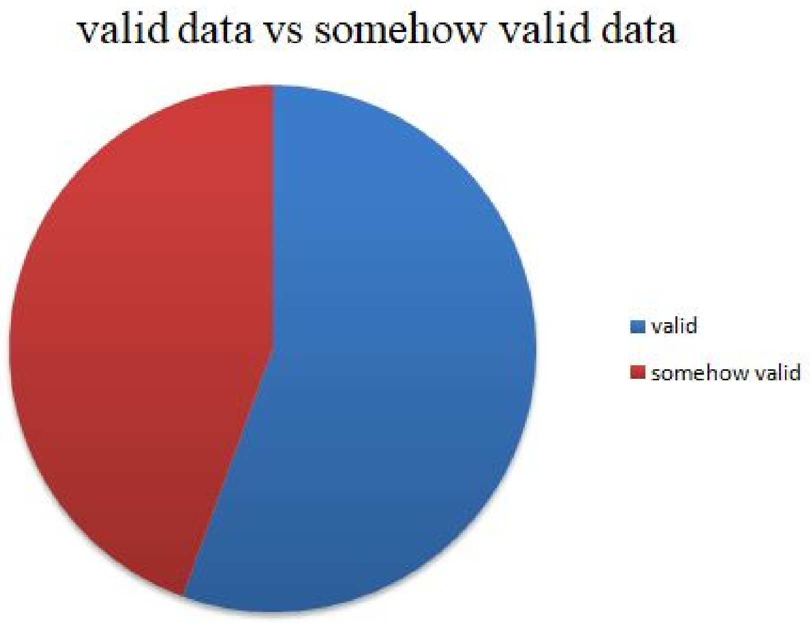


Figure 9: Graphical view_2 of various degrees of valid data

5. Implementation Details

Chapter 5: Implementation Details

5.1. Software and Tools Requirement Specifications

5.1.1. Operating System Used

CURMA is implemented on Windows 10 platform. But it can also be implemented on Linux, Mac OS platform with some changes in specifications.

5.1.2. System Configuration

- **RAM:** 4 GB or more for better performance.
- **HDD:** Minimum 20 GB or more.
- **Processor:** Intel Pentium 4 or any upgraded version.

5.1.3. Browser Compatibility

Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge, Opera Mini etc.

5.1.4. Local Host Server

XAMPP Server

5.1.5. Back End Tool

PHP with Machine Learning Composer Packages

5.1.6. Front End Tool

HTML, CSS

5.1.7. Scripting Tool

JAVASCRIPT

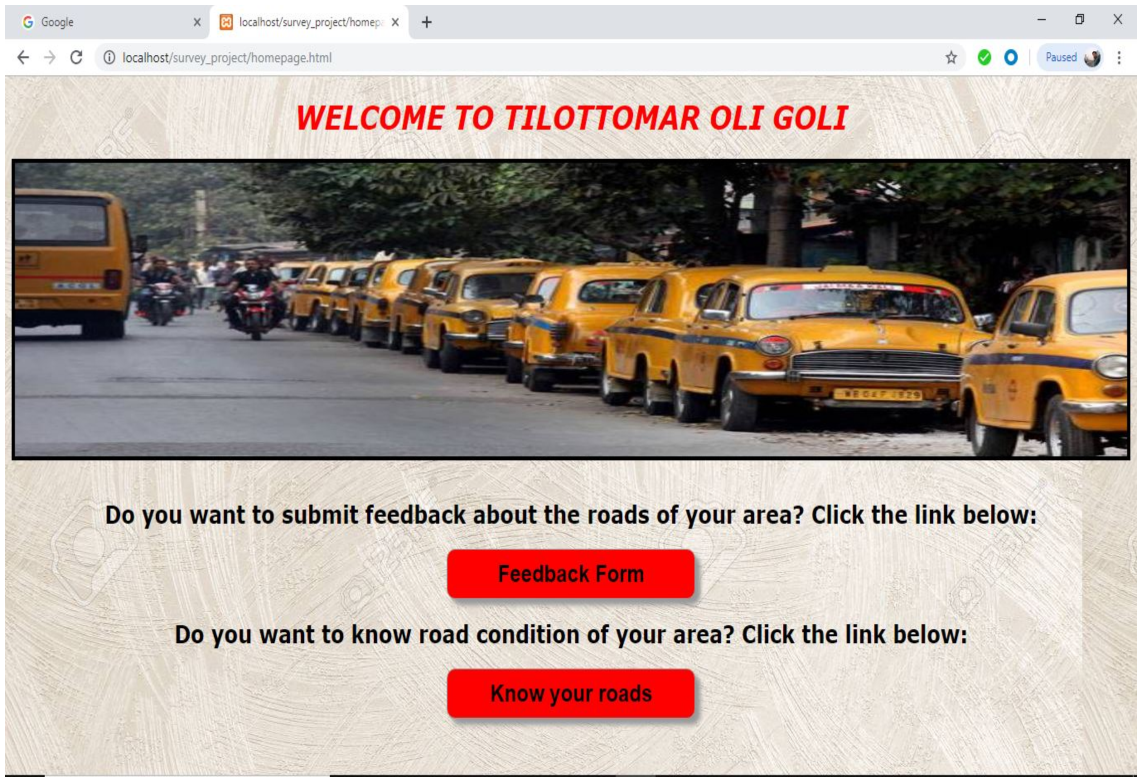
5.1.8. Database Connectivity

MySQL Server

5.2. User Interfaces at Data Validation Module

5.2.1. Homepage

The name our application is “**Tilottomar Oli Goli**”. There are two buttons on the home page of the application as shown below in Figure 10. One is the ‘Feedback Form’ which leads to the validation module of CURMA where users are allowed to select two areas of Kolkata city and submit feedback on any of the routes between chosen areas which will be stored in the database after three phases of validation. The second button ‘Know your roads’ leads to the Route optimization module. An automated slide show of various pictures regarding the roads Kolkata is added to make the Homepage more attractive to the users.


Google
localhost/survey_project/homep
localhost/survey_project/homepage.html
Paused
WELCOME TO TILOTTOMAR OLI GOLI
Feedback Form
Know your roads

Do you want to submit feedback about the roads of your area? Click the link below:

Do you want to know road condition of your area? Click the link below:

Figure 10: User interface of Homepage

5.2.2. Select Area page

After clicking on 'Feedback Form' button we can see the next page as shown in Figure 11 where the user is allowed to select two areas from the drop down list of areas of Kolkata city.

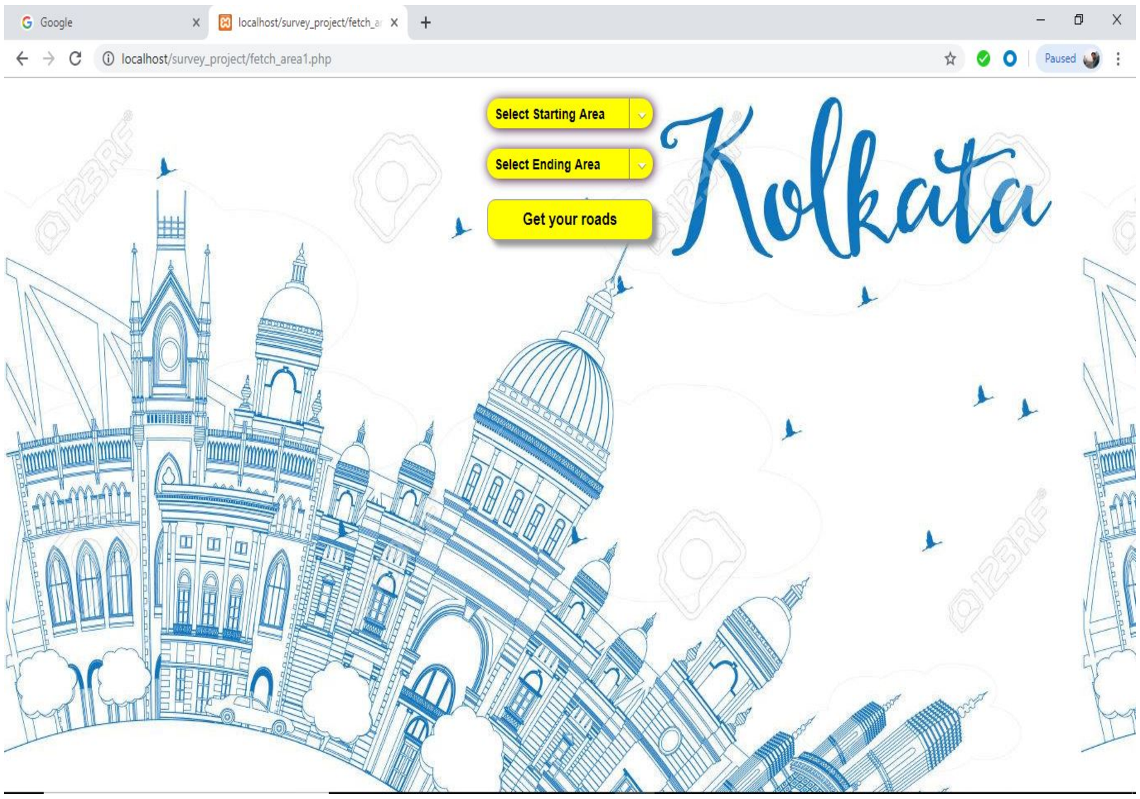
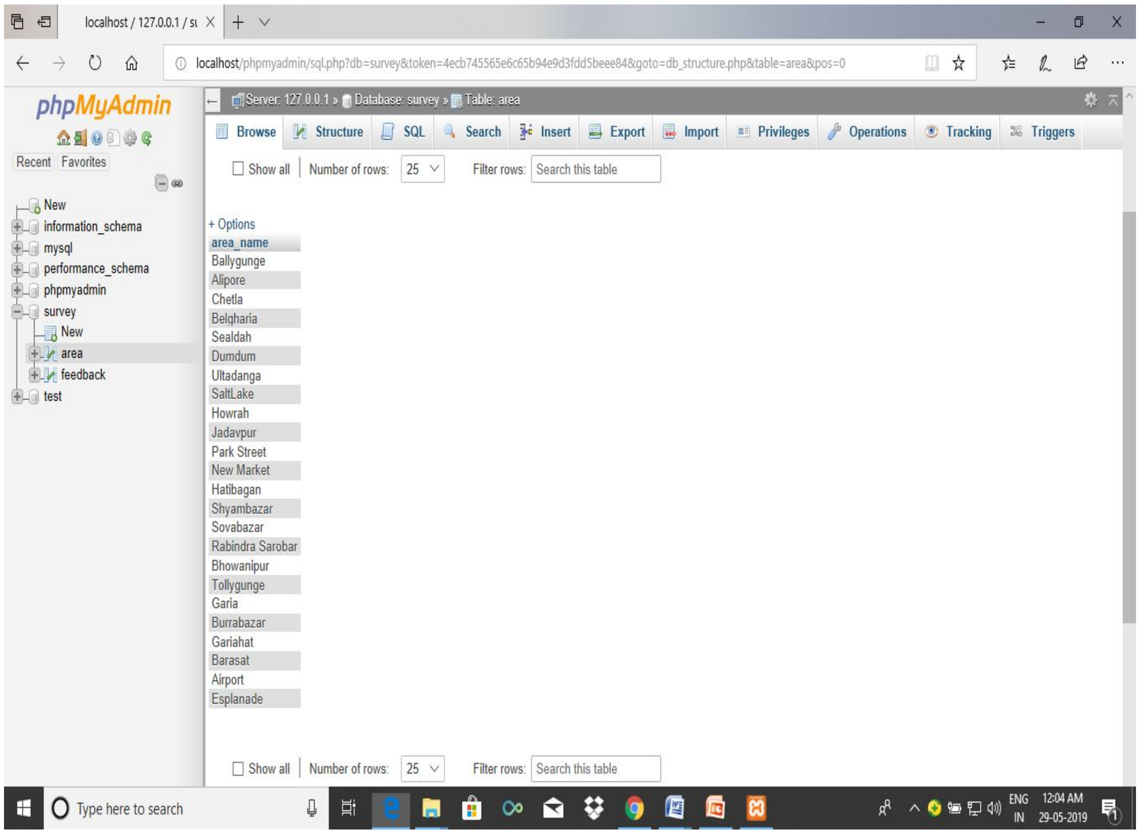

Google
localhost/survey_project/fetch_ar
localhost/survey_project/fetch_area1.php
Paused
Select Starting Area
Select Ending Area
Get your roads
Kolkata

Figure 11: User interface of Select Area page

Chapter 5: Implementation Details

Now in CURMA we have taken a list of some areas of Kolkata. The list of areas is stored along with a unique area code in the database, in a table named ‘area’. This is shown in Figure 12. So among the list of areas we are fetching it in a dropdown box and allowing users to select from it. For e.g. let us select Howrah from the first dropdown list and SaltLake from the second dropdown list.



localhost/phpmyadmin/sql.php?db=survey&token=4ecb745565e6c65b94e9d3fdd5beee84&goto=db_structure.php&table=area&pos=0

- New
- information_schema
- mysql
- performance_schema
- phpmyadmin
- survey
  - New
  - area
  - feedback
- test

area_name
Ballygunge
Alipore
Chetla
Belgharia
Sealdah
Dumdum
Ultadanga
SaltLake
Howrah
Jadavpur
Park Street
New Market
Hatibagan
Shyambazar
Sovabazar
Rabindra Sarobar
Bhowanipur
Tollygunge
Garia
Burrabazar
Gariahat
Barasat
Airport
Esplanade

Figure 12: Database scenario of Area table

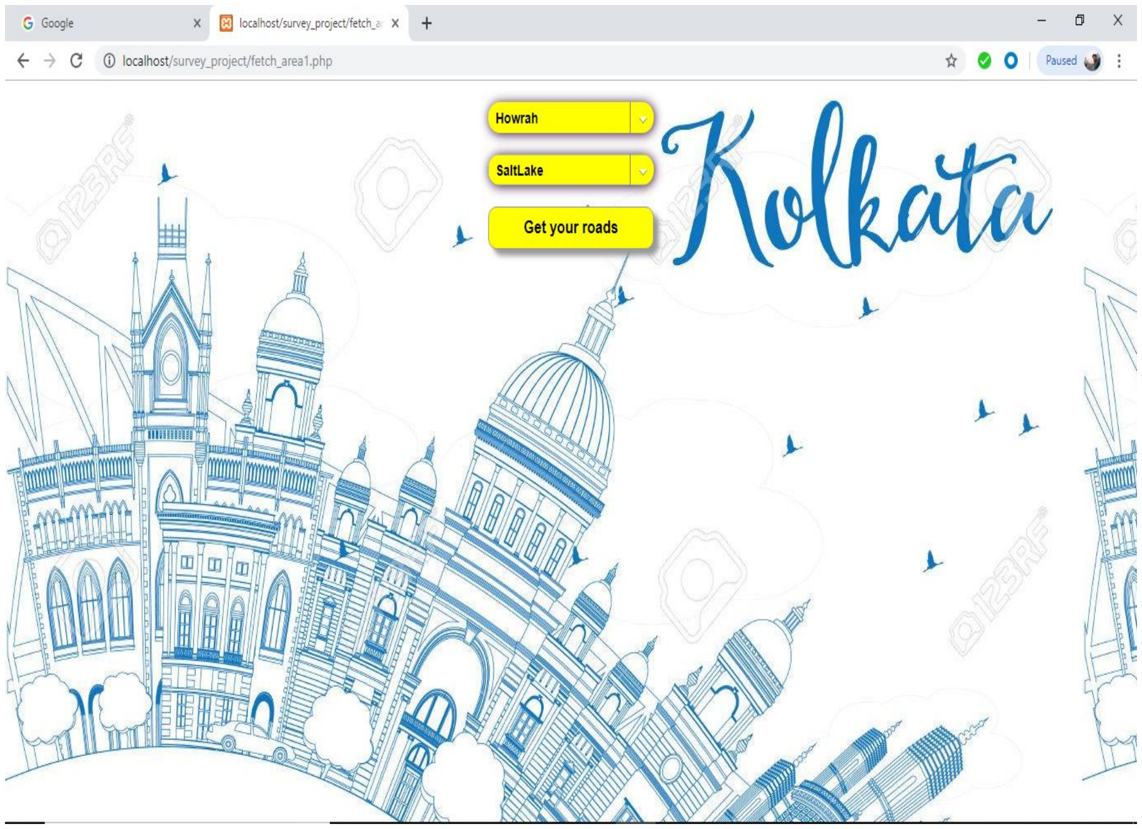
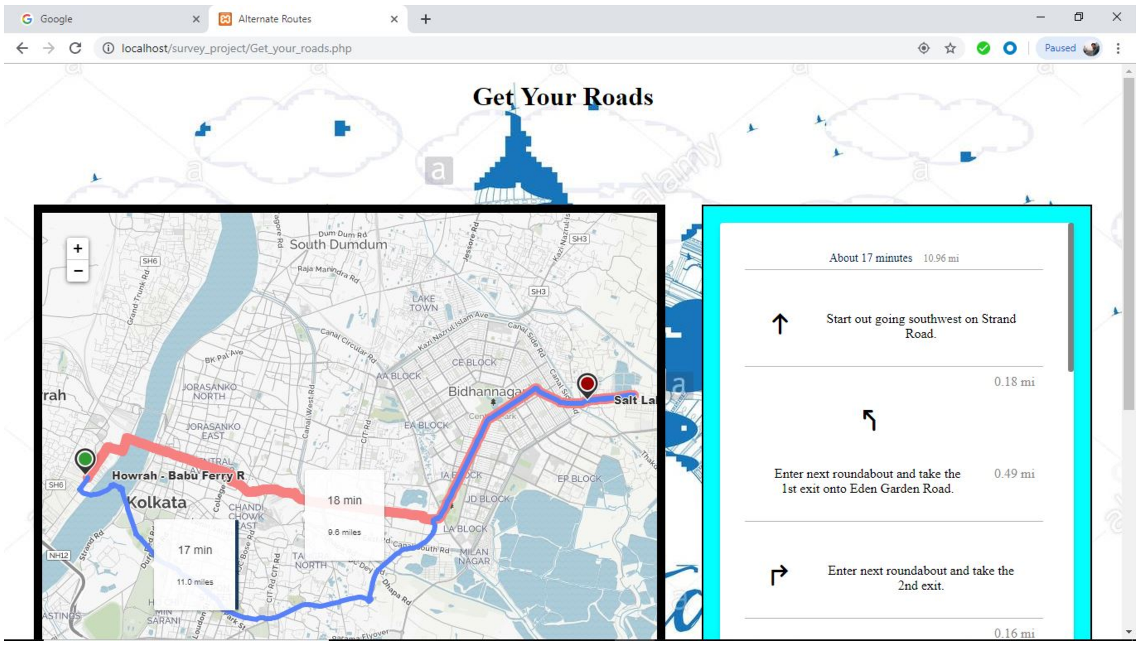

Google
localhost/survey_project/fetch_a
localhost/survey_project/fetch_area1.php
Paused
Howrah
SaltLake
Get your roads
Kolkata

Figure 13: User interface of Select Area page

5.2.3. Map View page

Now after clicking on ‘Get your roads’ button we are directed to the next page where multiple routes between Howrah and SaltLake will be visible in a map view of Kolkata city. This map service is achieved by using a third party API service provider named MapQuest API Service with Leaflet Map Plug-in (mentioned at section 4.3.1.). The right hand side dialogue box shows the directions corresponding to the route indicated by blue colour shown on the map among the multiple routes shown. This is depicted in Figure 14 and Figure 15 below. As we scroll down the page we see the description of each route as a combination of some roads of Kolkata.

Chapter 5: Implementation Details



About 17 minutes 10.96 mi

Start out going southwest on Strand Road.

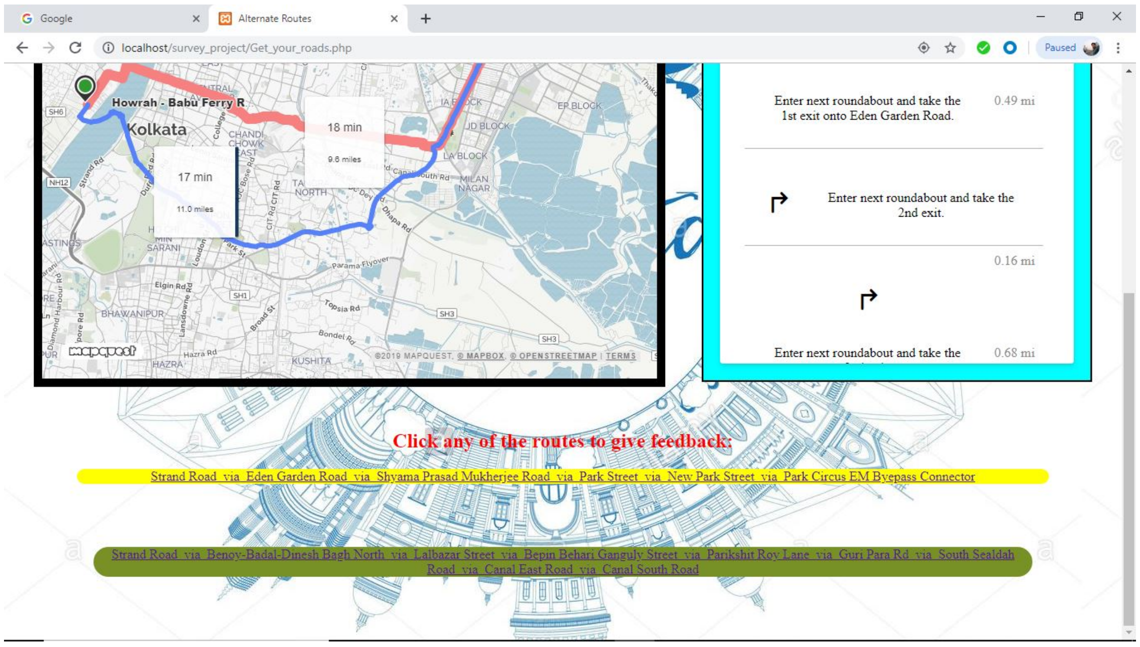
0.18 mi

Enter next roundabout and take the 1st exit onto Eden Garden Road. 0.49 mi

Enter next roundabout and take the 2nd exit.

0.16 mi

Figure 14: User interface of Map View page



Enter next roundabout and take the 1st exit onto Eden Garden Road. 0.49 mi

Enter next roundabout and take the 2nd exit.

0.16 mi

Enter next roundabout and take the 0.68 mi

Click any of the routes to give feedback:

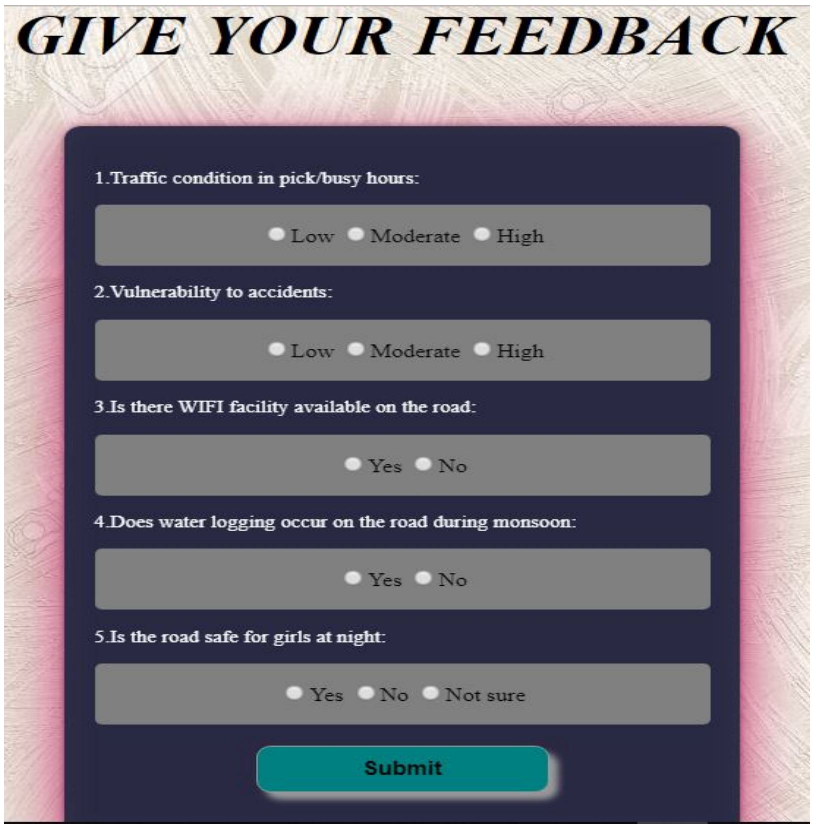
Strand Road via Eden Garden Road via Shyama Prasad Mukherjee Road via Park Street via New Park Street via Park Circus EM Byepass Connector

Strand Road via Benoy-Badal-Dinesh Bagh North via Lalbazar Street via Bepin Behari Ganguly Street via Parikshit Roy Lane via Guri Para Rd via South Sealdah Road via Canal East Road via Canal South Road

Figure 15: User interface of Map View page

Each of the descriptions appeared at this interface is a hyperlink. On clicking each of them users will be redirected to a "Submit Feedback" module where users can give their valuable feedbacks on the selected route.

5.2.4. Submit Feedback page



GIVE YOUR FEEDBACK

1.Traffic condition in pick/busy hours:

○ Low ○ Moderate ○ High

2.Vulnerability to accidents:

○ Low ○ Moderate ○ High

3.Is there WIFI facility available on the road:

○ Yes ○ No

4.Does water logging occur on the road during monsoon:

○ Yes ○ No

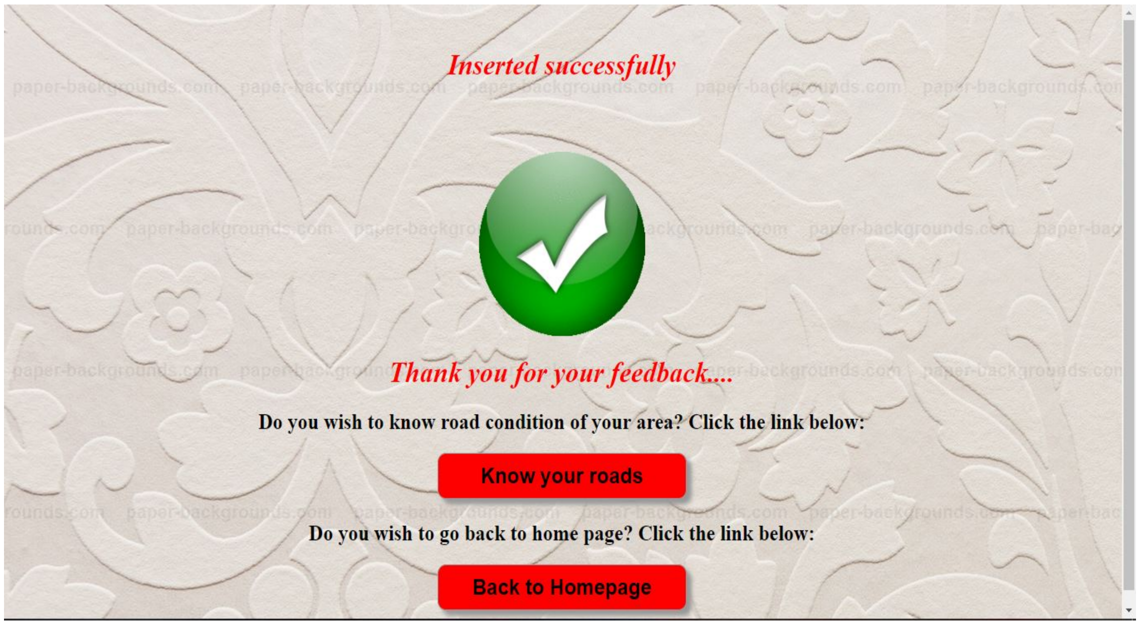
5.Is the road safe for girls at night:

○ Yes ○ No ○ Not sure

Figure 16: User interface of Submit Feedback page

This page looks like a more or less feedback-form where user can give their feedback in form of user ratings based of some pre parameters like "Traffic condition", "Accident vulnerability", "WIFI facility", "Water log", "Women safety" etc. After filling up each and every field then only a user will be able to redirect to the next user interface by clicking on "Submit" button.

5.2.5. Confirmation page


Inserted successfully

Thank you for your feedback....

Do you wish to know road condition of your area? Click the link below:

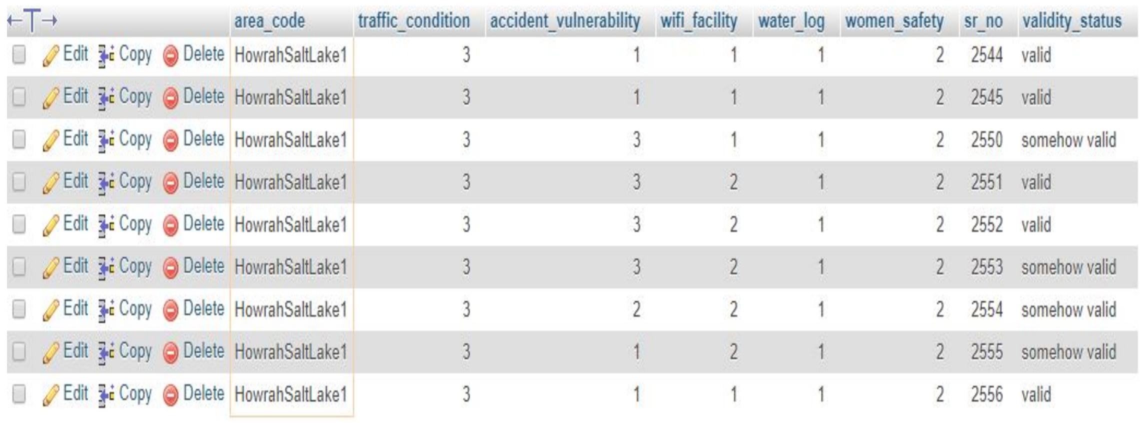
Do you wish to go back to home page? Click the link below:

Figure 17: User interface of Confirmation page

After clicking on the ‘Submit’ button on ‘Submit Feedback’ page, user will be redirected to a confirmation page depicted at Figure 17. There are two more buttons available in this page. One is ‘Know your roads’ button and another is ‘Back to Homepage’ button. By clicking on the ‘Know your roads’ user will be redirected to the Data Optimization module of CURMA. User will be redirected to the homepage by clicking on ‘Back to Homepage’ button.

5.3. Data Validation as a background process

The most interesting part of Data Validation module is that redirecting to the 'Confirmation page' does not give any guarantee for a user feedback being stored in the database. Actually data validation starts when a user is redirected to the 'Map View' page. At 'Map View' page, Geographic Location based validation (mentioned at section 4.3.1.) is done. Later when a user is redirected to the 'Submit Feedback' page, then Screen Time based validation (mentioned at section 4.3.2.) is done. And lastly when a user submits feedback then Machine Learning Classifier based validation (mentioned at section 4.3.3.) is run over the received data from the user. A user feedback will be stored in the database only when it gains sufficient validity fitness value (mentioned at section 4.3.5.) after passing all the three phases of validation. Real database containing valid data will look like as follows:



area_code	traffic_condition	accident_vulnerability	wifi_facility	water_log	women_safety	sr_no	validity_status
HowrahSaltLake1	3	1	1	1	2	2544	valid
HowrahSaltLake1	3	1	1	1	2	2545	valid
HowrahSaltLake1	3	3	1	1	2	2550	somehow valid
HowrahSaltLake1	3	3	2	1	2	2551	valid
HowrahSaltLake1	3	3	2	1	2	2552	valid
HowrahSaltLake1	3	3	2	1	2	2553	somehow valid
HowrahSaltLake1	3	2	2	1	2	2554	somehow valid
HowrahSaltLake1	3	1	2	1	2	2555	somehow valid
HowrahSaltLake1	3	1	1	1	2	2556	valid

Figure 18: Real database scenario of valid data

6. Conclusion and Future Aspects

6.1. Scope of Improvement

6.2. Future aspects of data validation

Chapter 6: Conclusion and Future Aspects

Today is the era of big data. Billions and billions of data are generated in every second in today's world and in next moment these data are needed to be analysed to get optimized outcome. Crowdsourced data collection is definitely a new trend in this field of data generation. In every moment huge amount of data are coming from the crowd. As discussed earlier that data collected from the crowd need to be checked for quality assurance. Otherwise optimization using the raw, unchecked data will lose its value to the end users in near future. For example nowadays our lives are dependent on the social media like Facebook, Twitter, Instagram etc. All these platforms are the boost of crowdsourced data. And all these social media has huge impact on our daily, social lives. Let's consider a very practical scenario. At our facebook newsfeed we are watching a controversial post and in the next moment without judging the source or the content of the post we are sharing it. In other words, we may be spreading a rumour which is not even true unknowingly. But what if there is a validation checking on this post before it goes viral. Then definitely actual truth may come out and false would be restricted from spreading over the net. Our data validation framework speaks of a possibility of data pre processing in more efficient and accurate way, speaks of a judgement day for any data before making any sense of it. Though we are presenting a framework for data validation, but still it is just the cape of the ice glair. There are lots of corners where we can improve data validation, modify our framework to be more accurate. Let's try to make a list where modification can be done.

6.1. Scope of Improvement

Improvement or modification can be done in following fields:

- Screen Time based validation can be made more accurate and real time.
- Machine Learning Classifier based validation can be more decorated and variety of classification can be included with the existing module
- Cross Judgement based validation – an entirely new trend can be introduced.

6.2. Future aspects of data validation

Data validation can be a trend setter as well as a show stealer in near future. Day by day data generation is increasing at a high rate. At the same time data is becoming less reliable and losing its trustworthiness. Data is becoming noisy. To get rid of all such data failure data validation can be a great help. Data validation can be used in every sector including banking, sports, health care, insurance, consumer trade etc. Google is recently allowing its users to give information regarding any traffic jam or accident occurring in the roads of metropolitan cities through their Google Map API service [15]. This sounds similar to our work. Considering all the above mentioned facts we can conclude that data validation is a quality assurance model in data science for now and future.

BIBLIOGRAPHY

[1] https://en.wikipedia.org/wiki/Crowdsourcing

[2] https://crowdsourcingweek.com/what-is-crowdsourcing/

[3] https://blog.skipsolabs.com/types-of-crowdsourcing

[4] https://www.maketecheasier.com/crowdsourcing-mobile-apps/

[5] https://developer.mapquest.com/

[6] https://www.ushahidi.com/

[7] https://archives.cjr.org/behind_the_news/the_challenge_of_verifying_cro.php

[8] VISUAL TOOLS FOR CROWDSOURCING DATA VALIDATION WITHIN THE GLOBELAND30 GEOPORTAL
E. Chuprikova, H. Wu, C. E. Murphy, L. Meng
The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B8, 2016 XXIII ISPRS Congress, 12–19 July 2016, Prague, Czech Republic.

[9] A flexible framework for assessing the quality of crowdsourced data
Sam Meek, Mike J Jackson, Didier J Leibovici
Huerta, Schade, Granell (Eds): Connecting a Digital Europe through Location and Place. Proceedings of the AGILE'2014 International Conference on Geographic Information Science, Castellón, June, 3-6, 2014. ISBN: 978-90-816960-4-3

[10] https://en.wikipedia.org/wiki/Data_validation

[11]https://www.informatica.com/in/services-and-training/glossary-of-terms/data-validation-definition.html#fbid=3t8Sj3jMEp_

[12] http://www.execuread.com/facts/

[13]https://www.analyticsindiamag.com/what-is-a-naive-bayes-classifier-and-what-significance-does-it-have-for-ml/

[14] http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/

[15]https://www.themobileindian.com/news/google-maps-users-can-now-report-accidents-speed-traps-through- latest-update-25594