A Smart Phone Based Bus Tracking System

Ganesh Yadav[1] ,Rajkumar Gupta[2] , Rajashree Gadhave[3]
Student[1,2],Faculty[3]
Pillai HOC College of Engineering & Technology,
Rasayani, Raigad, INDIA.

Abstract : . Tracking system involves the installation of an electronic device in a vehicle but electronic device is costly so we can use android app for bus tracking with an installed android app on any smart phone .user get the vehicle's location. There are two applications one for server and the other for the client. Bus driver start application on there android phone to track their positions. By this positions to server are periodically updated. Client application displays map showing the position of bus. It shows where buses are on a map and provide users the updated information at different time interval. The server will monitor location and will store its data in the database. It is a real-time system as this method automatically sends the information on the gps system to a central computer or system/smart phone.

IndexTerms **- Passenger inconvenience, real-time information, Real-Time Bus Tracking System (RTBTS), smart phone application, API, web server.**

1.INTRODUCTION

Transport systems are one of the basic and daily needs of common man. People spend a lot of time in travelling and availing the facilities of these systems. But a lot of time is being wasted in waiting for the means of transport. Many times, the waiting time is more than the travelling time. Hence a proper solution must be provided to tackle this problem of waiting time which is wasted almost daily.Everyone would have experienced that whenever we wait for a bus to come and are actually in a dilemma that weather to wait for a bus or to opt for other means of transport? This question needs a proper answer. A proper solution to this problem faced which is being faced daily. Waiting time must be minimized if not eliminated.

Real Time tracking of all vehicles of travel agency using Global Positioning System as the name suggests, it uses GPS to track any vehicles. Our area of concentration in this project is on tracking travel agency vehicles, and can be further implemented on government transport system as well. It will help both the passengers and the agency to track the vehicles, to get real time position of the vehicles, changed routes (If any), it can also act as an anti theft application by detecting the exact position of the vehicle.
The application will ask the user (Passenger) to enter the bus number in which he/she wishes to travel. Then the user will enter the source and the destination of their journey. After entering all the necessary information, the user will click on the locate button. Upon clicking the locate button, the user will get all the detailed information about the location of the bus, the fare user has to pay for his/her journey.

2.OVERVIEW OF THE PROPOSED SYSTEM

2.1 PROBLEM STATEMENT:

Management of buses of public transportation system is the main problem now a day.Based on to the current system there is no such system which provides information about the bus,its expected arrival time, the expected waiting time and what is the current location of the bus.

1 In the daily operation of bus transport systems, mainly that of buses, the movement of vehicles is affected by different uncertain conditions as the day progresses, such as :

- traffic congestion
- unexpected delays
- Irregular vehicle-dispatching times
- other incidents.

2 Many students are late for classes because they decide to wait for the bus instead of just simply using a alternate transportation

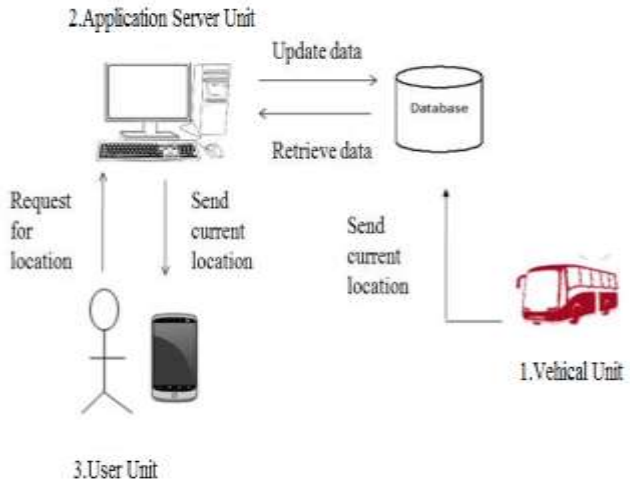
2.2 ARCHITECTURE OF PROPOSED SYSTEM :

The system aims to eliminate the waiting time of the commuter and provide them with actual positioning of the vehicles. Keeping this in mind we have developed a system taking into consideration various parameters that would reduce this time if not eliminate it completely.
.The Track It system consists of three major units as shown in the diagram. These are the User Unit ,vehicle Unit, application server unit.

User Unit:
The system comprises of three user interface:
Home Page: This interface consists of two buttons, one for viewing the available vehicles and second for closing the application.
Bus list: This interface displays all the available number of buses.
Map: This interface will show the position of the selected bus by the user on the G-map.

Vehicle Unit:
The vehicle unit is implemented on the driver side of the system and is responsible for sending position to the application server.
The driver enters the bus number he is driving and presses the start button before commencing the journey. Once pressing the start button, the application starts taking its self position and sends it to the application server after every specified time interval.
A toast showing the bus details along with the position details confirms that the data is being successfully sent to the application server
The application keeps on sending the position to the server until the drivers presses the stop button.

Application Server Unit:
This unit is responsible for storing, sending and receiving location data from the vehicle and from the users as well. Application server stores the position of the vehicles and sends the latest updated value to the user upon their request. It is a WAMP server and the database used to store the position data is MYSQL. The data in the application server is updated after every specific interval of time and sends data to the users on demand.



LITERATURE SURVEY

Following data represents already implemented techniques for software bugs classification:

1) **Towards effective bug triage with software data reduction techniques** JifengXuan , He Jiang , Yan Hu , ZhileiRen , WeiqinZou , Xindong Wu

This paper highlights the two important algorithms, Instance and Feature algorithms that deals with mining repositories. These algorithms are combination of multiple algorithms that we can use for mining process . Naïve Bayes algorithm will be used for performing text classification.

2) Mapping bug reports to closely connected files : A ranking model , a fine- grained feature evaluation

XinYe ,RazvanBunescu , Chang Liu
This paper let us know that the source files happen to be in large contain compared to actual bug files present in it. Bug files could be way much smaller compared to whole source files. Hence, ranking approach has been implemented to source files to rank them according to their relevance.

3) Micheal W. Godfrey, Olga Baysal and Robin Cohen developed a model for automatic assignment of bugs to developers for fixation using vector space model .

In this paper, the authors have proposed a specimen of the intelligent system that instinctively conducts the bug assignment. They have employed the vector space model to infer information about the developer's expertise from the history of the previously fixed bugs. The vector model is used to retrieve the title and the description from the report to build a vector which later can be used to find similar reports by mining the data in the bug repository. In order to create an efficient bug triage model, the authors conducted a survey wherein they collected a feedback from the developers regarding their previous bug fixing experience, their satisfaction with the bug assignment, whether they were successful and confident in handling bugs in the past, etc. The overall information provided them the initial estimates for the proposed model. This in turn helped them to implement the specimen model and test it within a software team working on the maintenance activities.

4) Lei Xu , Lian Yu , Jingtao Zhao , Changzu Kong , Huihui Zhang put forth a technique which used data mining techniques to automatically classify bugs of web- based applications by predicting their bug type.

The authors have put forth that the debug strategy acquaints us with the errorneous part of the source code. Once the errors are found then it is very easy for the developers to fix them. The determined association rules help to predict files that usually change together such as functions and variables.

I. PROBLEM SOLUTION

To obtain the solution for the bugs that are generated, we can implement our two important algorithms, instance selection and feature selection. These algorithms will help to reduce the time amount taken for resolving the generated bugs.

II. INPUT DATA

The data of software's such as Eclipse and Mozilla Firefox happen to be obtained from bugzilla -an open bug repository
Datasets of bug reports are obtained. This data is divided into training and testing groups, experiments are performed on different set of data from these groups. In this Project, we are using our Bug Repository and we will also make our own compiler for taking input. First we have to compile file and take input from there.

III. PRE-PROCESSING

Data preprocessing is the most important step of data mining . Raw data can be obtained from bug repositories which cannot be directly used for training the classification algorithm. Hence, the data needs to be pre-processed to make it functional for training purpose. Data pre-processing is a monotonous step of data mining and important as well. Stop-words dictionary and regular expression rules are used to filter redundant and irrelevant words and filter the punctuations respectively. Stemming algorithm is used to stem the vocabulary.

IV. FEATURE SELECTION

After implementing bunch of words approach on data, the result that is obtained has very large dimensionality. Many of these dimensions are not related to text categorization and thus in turn result in reducing the attainment of the classifier. Feature selection is the process that helps to reduce the intensity of the obtained vocabulary. In this technique, best k terms out of the whole vocabulary are elected which contribute to accuracy and efficiency.

V. There are a number of feature selection techniques such as Document Frequency (DF),Chi-Square Testing, Information Gain (IG), Term Frequency Inverse Document Frequency (TFIDF). In this research, we will use feature selection algorithm.

Algorithm and it's brief explanation :

Applied to bugs obtained recently:
IG, CH, SU,RF.
DEMONSTRATION:
Input : Create new department and new bug pattern.
Output: Generate new department in database and define new bug pattern in new department **.**
D(I) : Instant selection department
D(F) : Feature Selection department
P(f) : bug PatternSteps are as followed:
1.If department D (I) not exist;
2.Go to feature selection
3.Create D (F) and send to database;
4.Define bug pattern p(f)
5.P(f) should be save in new D (F)
6.Terminate D(F);
7.Repeat 2 to 6 if again new department has to come.
8.end.

VI. INSTANCE SELECTION

Instance selection uses a methodology which is used for reducing the dimension of vocabulary obtained after applying to bunch of words. As most of the dimensions are related to our pre-defined bugs and result in reducing the performance of the classifier, hence to decrease the time, the process of Instance selection is used which chooses the best k terms out of the complete vocabulary which contribute to accuracy and efficiency. This selection is fast instance of feature selection.
Algorithm and it"s explanation:
Applied sequentially:

ICF, LVQ, DROP, POP
DEMONSTRATION:
Input: training set T with n stop words and m bug report,
Reduction order IS -> FS
Final number n(f) of words,
Final number m(I) of bug report,
Output: reduced, triage and send data set T (FI) to matching department.
Steps are as followed:

1. Admin (project manager) apply IS to n stop words T and calculate objective values for all words;
2. Select the top n(I) words of T and generate a training set T(I);
3. Filter bugs F(I) and send to suitable department D (I);
4. Terminate IS when new department to come;
5. end

VII. CLASSIFIER MODELING

An automated process to find some metadata about a document is considered as "Text classification." It has been used in various areas like document indexing by suggesting its categories in a "content management system", "spam filtering", "automatic help desk requests" sorting etc.
For bug classification, Naïve Bayes text classifier is used in this research. Naïve Bayes classifier is based on Bayes‟ theorem with maverick assumption and is a probabilistic classifier. It implies that the classifier speculates that any feature of a class is unassociated to the presence or absence of any other feature.

VIII. MODULES AND DESCRIPTION

This project shall require the implementation of following modules:
Authentication users, Products, Bug details, View, Admin and Logs.
Authentication users:
The Bug Tracking System first activates the login form. Here the user is prompted to enter the "user name‟ and „password‟ and our system starts the authentication process in which the username and password are verified with the help of existing username and password in the database. If the password matches then it is allowed to the main page else it warns the user for Invalid User name and password. After successful authentication, the system activates menus. The activity log has also been prepared for failures.
Bug Details:
The user is provided with the facility for adding bugs, updating the existing bugs in this module. As the number of bugs for a product can be very large, this system is equipped with efficient filtering. The user can filter the bugs based on priorities, databases, operating systems and status. After the user applies filter the list of bugs are displayed from the database.
More modules might be added as and when needed in to the stage of project development.

IX. CONCLUSION

In open source bug repositories, bugs are reported by users numerous times. Triaging of these bugs is a repetitive and protracted task. If some proper class is assigned to these bugs ,then they could be easily allotted to a relevant developer and thus bugs can be fixed efficiently. Still, as reporters of these bugs are mostly non-technical it would be unfeasible for them to assign correct class to these bugs. In this research multinomial Naïve Bayes text classifier is used for classifying software bugs. Instance selection algorithm and Feature selection algorithm are used for bug triage. Maximum accuracy at prediction can be obtained using this system. Bug triage is an expensive step of software maintenance in both labor and time cost. In this project, advantages like reduction in bug data sets and improved data quality can be obtained because of combination of instance and feature selection.

X. FUTURE WORK

The main challenge would be performing classification for numerous numbers of domains that could be tedious but important process. This will help teams of developers under one roof to work on their separate domains with fairness and utilization of time will be done properly.

REFERENCES

[1] A. Hotho, A. Nürnberger and G. Paaß, "A Brief Survey of Text Mining," vol. 20, GLDV Journal for Computational Linguistics and Language Technology, 2005, pp. 19-62.

[2] A. E. Hassan, "The Road Ahead for Mining Software Repositories," IEEE Computer society, pp. 48-57, 2008.

[3] S. Diehl, H. C. Gall and A. E. Hassan, "Special issue on mining software repositories," in Empirical Software Engineering An InternationalJournal © Springer Science+Business Media, 2009.

[4] O. B. Michael and G. C. Robin, "A Bug You Like: A Framework for Automated Assignment of Bugs.," IEEE 17th international conference, 2009.

[5] C. Zhang, H. Joshi, S. Ramaswamy and C. Bayrak, "A Dynamic Approach to Software Bug Estimation," in SpringerLink, 2008.

[6] L. Yu, C. Kong, L. Xu, J. Zhao and H. Zhang, "Mining Bug Classifier and Debug Strategy Association Rules f or Web-Based Applications," in 08 Proceedings of the 4th international conference on Advanced Data Mining and Applications , 2008.

[7] N. Jalbert and W. Weimer, "Automated Duplicate Detection for Bug Tracking Systems," in IEEE computer society, 2008

[8] T. Bruckhaus, C. X. Ling, N. H. Madhavji and S. Sheng, "Software Escalation Prediction with Data Mining," in Data Mining, Fifth IEEEInternational Conference, 2006.

[9] [Online]. Available: https://bugzilla.mozilla.org/.

[10] [Online].Available:https://bugs.eclipse.org/bugs/.

[11] [Online].Available:https://en.wikipedia.org/wiki/Naive_Bayes_classifier.