Addis Ababa University

College of Natural and Computational Sciences

School of Information Sciences

Designing Location Aware Active ATM Recommender for Banking Service

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Information Science

By: Berihun Hadis

Addis Ababa, Ethiopia

Adviser: Million Meshesha (PhD)

Addis Ababa, Ethiopia

February 10, 2020

Addis Ababa University

College of Natural and Computational Sciences

School of Information Sciences

Designing Location Aware Active ATM Recommender for Banking Service

Name and Signature of Members of the Examining Board

Name	**Title**	**Signature**	**Date**
______________________________	, Advisor	_________________	_________________
______________________________	, Examiner	_________________	_________________
______________________________	, Examiner	_________________	_________________

Dedication

I dedicated this work to my mother Zewude Tadesse, my father Hadis Wolde, my wife Meron Dagne and my children Absalat Berihun and Rediet Berihun.

Declaration

I declare that the thesis is my original work and it has not been presented for a degree in any other university. All the material sources used in this work are duly acknowledged.

Acknowledgment

First I would like to thank the almighty God for giving me the patience to have a positive feeling towards completing this thesis.

I would like to express my-deep felt gratitude to my advisor Dr. Million Meshesha for his guidance, constructive comments, and great deal of patience during conducting this research which helped me to face challenging conditions and finish this thesis. He also taught me the importance of intuition and simplicity of thought in the research process.

I would also like to thank Taddese Kassa for his motivation, valuable comment, as well as helping in organizing the document.

In this study, I received feedback from many peoples. In particular, from Bank of Abyssinia staffs i.e. Daniel Retta, and Surafel Negash for helping me to understand ATM domain area and evaluating the ATM prediction result; and Addisu Andarge helping me in collecting the physical ATM data. I would like to thank them for their valuable feedback and support.

I would also like to thank Getachew Dejene for his valuable advice related to GIS domain area.

My special thanks extend to my family for their support which made this work possible.

Abstract

Card banking customer in Ethiopia faces the problem of identifying nearby active ATM service. Research shows that more than 50% of card banking customers are unsatisfied with the existing card banking service at their respective banks in Ethiopia. The aim of this study is to explore and design location aware active ATM recommender for banking service. To this end, the study first did an experiment to construct a predictive model for determining active ATM. This is followed by identifying nearby active ATM and presenting the optimal path from customer current position to nearest active ATM.

For active ATM prediction J48 and PART algorithms are experimented. In this experiment 44,105 instances with 8 attributes employed and WEKA tool is also used to construct a prediction model. For determining nearest active ATMs, Euclidean, City-block and Haversine algorithms with 13 records were employed. Also the experiment is conducted using GoogleMaps tool. For determining the optimal path Dijkstra's, A*, and BFS algorithms and 54,654 instances were used. To carry out this experiment GoogleMaps tool has been used. For developing a prototype application java program and MySQL tools has been used.

The experimental result for predicting active ATMs shows that J48 decision tree has better accuracy i.e. 99.81 compared to PART rule induction algorithm. The experiment result for determining nearest active ATMs shows that City-block has better performance than Euclidean, and Haversine algorithms. The experiment result for determining the optimal path shows that BFS has better performance than Dijkstra's, and A* algorithms.

Finally, we design a prototype for recommending nearby active ATMs using J48 model, City-block and BFS algorithms. Accordingly, the prototype recommends the nearest active ATMs with the optimal path as per the user preference (ATM type and radius) and other related nearest active ATMs which is different than the user preference. User acceptance test result shows that, the prototype system save time for the customer. As a result, the customers are satisfied by the prototype system. This study has limitation with respect to user need in how much effort the customer wants to get ATM service and studying this issue enhance customer satisfaction.

Key word: predict active ATM, location aware, determine nearest active ATM, determine shortest/optimal path, recommender.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Equations

List of Abbreviations

BOA - Bank of Abyssinia

A* - A Star Algorithm

Dijk- Dijkstra's Algorithm

BFS- Breadt-First Search

ATM - Automated Teller Machine

Gmap - GoogleMap

GPS - Global Positioning System

IDE – Integrated Development Environment

ISO - International Standard Organization

JDBC - Java Data Base Connectivity

JSON - JavaScript Object Notation

Lat - Latitude

LBS - Location Based Services

Long - Longitude

OSM - OpenStreetMap

PDA - Personal Digital Assistant

QGIS - Quantum Geographical Information Systems

SQL - Structured Query Language

XML – Extensible Markup Language

Chapter 1

Introduction

1.1 Background

Connectivity to the Internet and software automation has changed the basic idea of traditional business operations. As a result of this, since 1990s, the way of doing businesses in the banking sector has massively changed and moved to centralized databases, online transactions and wide spread of ATMs throughout the world [1]. This makes banks to become closer to the customer and technically capable as they upgrade their skills and knowledge to cope up the challenge [1].

Banks need to find new customers as well satisfy the existing customers to survive in the fierce competition of the banking market [2]. Thus, banks introduce enormous services as a means to retain the existing customers as well as to attract new customers. One of the most important of these services is Automated Teller Machine (ATM) [2].

Automated Teller Machine (ATM) is a shared information system that connects private banks, government banks and other finance sectors with retail card banking customers to carry out banking transactions such as cash withdraw, fund transfer, payment, deposit and account balance inquiry [3]. ATM provides bank customers with 24-hour access to banking products/services; they are easy to use and are faster than human tellers in the banking halls. ATM services have improved the organizational efficiency of banks and customer's services in the banking sector [3].

Banks increase the availability of ATMs to give 24/7 ATM services by monitor the progress of ATMs using dedicated skilled team members, network infrastructure and monitoring software's. Active ATM monitoring or ATM monitoring is the process of checking the condition of the ATM and its hardware components mounted to it [4]. It also keeps track of transactions carried out on the ATM. It maximizes availability of the ATM service for the bank customer by reducing the downtime. As the downtime increases the customer use other banks ATMs which makes the owner bank to pay commission fee for the service. This increase the expense of the owner bank. In addition to this, if there is high downtime, the bank reputation adversely impacted, thus, leads to loss of loyal customers and challenge to acquire new customers as well [4].

Active ATM is an ATM that gives service after the prediction model identified them as in-service and propose them by the recommender system to be used by the customer. Alternatively, also called available, availability, up, uptime, and online are used interchangeably with in service when the independent self-service machine ATM can make electronic payment successfully. Or Active ATM also called in-service ATM or up ATM which gives usual service for customers such as cash withdrawal, balance inquiry, fund transfer, cash deposit. Active ATM status information is obtained by constructing predictive model using classification algorithms.

Optimizing ATM service also needs to provide information for customers concerning location based active ATM. Location Based Service (location aware service) provide information by considering the user's current location. In order to function, the location based service needed as a core component Geographical Information System (GIS). Location Based Service (LBS) provides opportunity for different stakeholders such as developers and cellular service network operators to give customer centered service for their mobile clients [5].

Gugapriya et al. [5] attempt to design nearest ATM center locator using GPS. Das [6] work come up with a prototype that identifies the nearest ATM location, show shortest path on map, show marker on map and allow volunteer to add new ATM. Faried [7] design a system that allow customers to search ATM location in the neighborhood, allow navigation and augmented reality and provide ATM availability condition.

The essential requirement of location-based services components are five [8]. The first component is mobile handheld devices, which are small computers that can be held in one hand or PC. For most cases, they are smartphones [8]. The second component is positioning system, which is a navigation satellite system that provides location and time information to anyone with a receiver. The third component is mobile and wireless networks, which relay the query and location information from devices to service providers and send the results from the providers to devices. The fourth component is service providers, which provide the location-based services. The fifth and the last component is Geographical data providers, which are databases storing a huge amount of geographical data such as information about restaurants and gas stations.

Location based service address various problems in different domain areas including the following [9], [10], and [11]: identify the nearest emergency center for those who are in emergency, identify the nearest hospital for patients, identify the nearest ATM for card banking customers, and show the shortest route with less traffic congestion for the community.

To design LBS, the essential requirement should be there i.e. smart phone/PC which identify the customer location; LBS server, contains the business logic of the LBS application; and LBS data store which store the physical location of objects that the customer seeks to find [12]. The client connection with the LBS server is established via http protocol of the Internet. The LBS server connected to the LBS data store via JDBC driver. The other key component to design LBS is straight-line distance algorithms which finds the nearest item for the customer. The other component to design LBS is optimal path finder algorithm which finds the shortest path from the customer location to the identified object(ATM).

Giving the right to choose services that fit the customer preference improve customer service satisfaction. One mechanism to provide this type of service is through the implementation of recommender system. "A recommender system or a recommendation system (sometimes replacing 'system' with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications" [34].

1.2 Statement of the problem

As the adoption of ATMs increases in Ethiopian banks, the number of card banking customers also increased by double in the past four years (2015-2019) [13]. For instance, in Bank of Abyssinia, the number of ATM card holders increase from 15,000 in 2015 G.C to 365,390 in 2019 G.C [13]. With respect to service provision, the work of Gezahegn [14] reveals that, out of 379 ATM card holders, around half are unsatisfied with ATM services from their respective banks. Due to inconvenient ATM service the customers are unsatisfied and therefore, ATM service delivery for card banking customer in Ethiopian banks requires further work to increase the satisfaction level of customers.

Preliminary survey reveals that finding nearest ATMs is challenging for card banking customers (refer appendix IV question no. 2). The current ATM service practice in Ethiopia indicates that, the card banking customer expected to search for existence of ATM as much nearby as possible. For instance, someone who need cash for emergency use or who wants to make foreign exchange, especially when branches are closed must know the exact location of the deployed active ATM to get cash. Finding ATM becomes even more demanding task for tourists in unfamiliar city where they want to spend as much time as possible having fun and visiting instead of looking for in-service ATMs by moving from place to place to get cash. Therefore, the need for nearby active ATM finder is significant for Ethiopia card banking customers as well as tourists.

Getting the nearest ATM doesn't mean that the ATM is working. The discovered ATM sometimes might be out of service. According to Wondimu [15], the main reason for ATM out of services are dispenser devices failure, card reader failure, network connection failure, and application failure. Therefore, the customers may not get the desired ATM service for their immediate use. Currently, the card banking customers of Ethiopian banks has no any means to know which ATMs are in-service and nearest; and which ones are out of service before starting to look for the ATM. Lack of ATM status prediction expose the customers for loss of time, effort and money. Besides, this trend impedes the card banking customers not to use their own resource (money) for their immediate purpose whenever they need it urgently. This problem leads to customer service dissatisfaction, thereby banks reputation adversely impacted and may loss loyal customers which have impact on revenue or profit. Therefore, the need for active or in-service ATMs information for the card banking customers are significant.

Different researches have been conducted on location-based ATM services. Those researches allow customer to get the nearest ATM on the map [6], suggest shortest road route or path on the map [16], identify in-service ATMs [7], support navigation and augmented reality [7], suggest alternative road when traffic congestion occurred [16], see list of ATMs by banks and allow to add new ATM location [6]. The only attempt made in our country is the work of Wondimu [15] that attempted to identify the reason for ATM failure which can be used as a base for predicting the probability of a given ATM active or not, but this work addresses predicting in-active ATMs. Therefore, those works have limitation on: usability of the system, inconsistent information as a result of accepting duplicate map information [5]. Those works also lack active ATM status prediction. Therefore, adopting directly those research outputs is not fit for Ethiopian banks location-based ATM services.

As to the researcher knowledge, there is no study conducted on location aware systems that recommends nearby active ATM for Ethiopian banks. Hence, the card banking customers are challenged in finding the nearest active ATM along with condition of the ATM information.

It is therefore the aim of this study to explore the possibility and design location aware active ATM service recommender for Ethiopian banks. To this end, the following research questions are explored and answered in this study.

i. What are the suitable attributes and algorithms for constructing ATM status prediction model?
ii. What are the major considerations and requirements to recommend active ATM for card banking customers?
iii. Which prediction, shortest straight-line distance and optimal path algorithms work better?
iv. To what extent location aware active ATM service recommender be accepted by customers?

1.3 Objective of the study

1.3.1 General objective

The general objective of this research is to propose and design a location aware active ATM recommender for Ethiopian banks so as to simplify the ATM service provision for users.

1.3.2 Specific objectives

To achieve the general objective of the research, the following specific objectives are formulated:

- ✓ To review literature to identify and understand different approaches, techniques and tools used in active ATM prediction, in determining nearest ATM and optimal path.
- ✓ To prepare data set for training and testing helpful for designing prediction model, determining nearest ATM and optimal path.
- ✓ To experiment different classification algorithms to design a predictive model.
- ✓ To experiment different straight-line distance algorithms for selecting nearby ATM.
- ✓ To experiment different road network route selection algorithms for identifying optimal path to the target ATM.
- ✓ To develop a prototype for nearby active ATM recommender related to user's interest.
- ✓ To evaluate the performance of the prototype.

1.4 Scope and limitation of the study

The scope of this research is to develop location aware active ATM service recommender for Ethiopian banks. To this end the study uses two classification algorithms for constructing ATM status predictive model. This is followed by identifying straight-line distance measuring algorithms to find the nearest ATM and informed road search algorithms to identify the shortest path from user location to recommended active ATM location. Finally, a prototype is designed that integrate the prediction model with location aware recommender so as to suggest closest active ATMs that is related to the requirement of card banking customers.

Besides the interesting findings observed in this study, the following limitations are recorded:

- IP address-based location detection attempted but the result shows description of Addis Ababa city rather than specific location.
- Experimenting using full data set of All Addis Ababa city is attempted but the experiment machine RAM capacity do no support to carry out with the full dataset.

1.5 Significance of the study

For scholars this research has the following benefits with respect to location-based services. This study presents clearly the requirements for location aware service and can provide insight on it. Extensive data and domain understanding on GIS and ATM data has been made in consultation with domain experts, so that this study shows its implementation for predicting active ATM and for designing location aware systems and gives insight to learn on it. Extensive data preprocessing activities has been conducted on this study, hence one can consider it as reference for further similar studies.

With respect to addressing the challenge to find the nearest ATM service, location-based service or location aware service is one of the solutions. It helps card banking customer and tourists who is new to the location/area and in need of cash for emergency purpose, transfer fund and deposit cash by recommending the nearest ATM location, so that the customer can conveniently get the ATM service. Therefore, this research has benefit with respect to time, cost and effort advantages as they get nearest active (in service) ATM information in the customer's surroundings via their hand-held devices or personal computer by rejecting out of service ATMs. This helps them to use their money efficiently and timely for the intended purpose.

For financial institutions or banks this research has the following advantages. First, it helps them promote brand loyalty and improves customer service and satisfaction by enhancing customer relationships through increased engagement level and opportunity to present more relevant content in real time at banks. Second, the result of this research can act as an efficient platform to capture customer's attention, resulting in higher spent time at one place. This can lead to increased retention and potential for lead generation for the bank. This study also drives cross-sales opportunity by delivering relevant information to interested customers when they are at the branch or any other related business area of the bank. In addition, this study will enable banks to deliver better, richer and more efficient applications and services for accessing relevant data in real time. Finally, the result of this study offers a heightened reach as a device-agnostic solution not limited to smartphone.

1.6 Methodology of the study

1.6.1 Research design

This research follows experimental research design to explore and design a prototype location aware active ATM for card banking customers. "Experimental research investigates the possible cause-and-effect relationship by manipulating independent variables to influence the dependent variable(s) in the experimental group, and by controlling the other relevant variables, and measuring the effects of the manipulation by some statistical means" [17]. Steps in experimental research include the following, devising alternative research questions, designing crucial experiments with alternative possible outcomes, each of which exclude one or more possible research questions and finally conducting the experiment, get a clean result and measure the performance of classification, straight line distance and informed route search. So for conducting the experiment we undertake three major task; data preparation, implementation tools and algorithms selection and evaluation of the prototype.

According to preliminary survey, currently there are 19 government and private banks in Ethiopia that operates throughout the country by opening branches and agents (refer appendix IV question no. 3). The preliminary investigation revealed that, studying one Ethiopian bank business operation can represent the others as the products and services provided in Ethiopia banks are similar. Hence, this study considers Bank of Abyssinia to obtain data that is useful for the experiment and for the prototype application consumption, because it is convenient and time saving for the researcher to get the relevant data.

Preliminary investigation revealed that, except National Bank of Ethiopia, all other banks adopted and provide services via E-banking channels such as ATM, POS, Mobile banking and internet banking for their customers and most banks have deployed ATM throughout the country (refer appendix IV question no. 4).

This study is geographically limited in Addis Ababa, which is the capital city of Ethiopia. Addis Ababa is also the largest business city in the country, hence more ATMs are residing in the city.

1.6.2 Sampling and data collection

In this study we use sampling for different purpose and is explained next.

According to the preliminary survey result, the increase in card banking customer is due to key strategy focus by banks in digital banking/e-banking domain area. For instance, Bank of Abyssinia organize digital banking/e-banking department in chief level to better achieve the bank strategic goal (refer appendix IV question no. 7).

According to the preliminary survey result, in Bank of Abyssinia, currently the number of ATMs deployed nationwide are 270 (refer appendix IV question no. 5). According to the revised five years' strategy [18], the bank planned to close to the customer easily by increasing more ATMs [18] to enhance the customer service satisfaction.

According to the survey result, in Bank of Abyssinia, the ATMs are deployed in three districts such as central, east and west Addis Ababa (refer appendix IV question no. 8). To select ATM that exist in all the districts of Addis Ababa, this study uses stratified random sampling. Using stratified sampling methods helps to give ATMs equal chance of being considered for selection from each of the four districts of the bank in Addis Ababa.

Sufficient number of sample ATMs is obtained from all districts using stratified random sampling methods. Using stratified random sampling methods helps to give ATMs equal chance of being considered for selection from each region.

Discussion with expert in the preliminary survey shows that, in Addis Ababa city the total number of ATMs owned by the bank is 150 (refer appendix IV question no. 6). The sample size of ATMs for this study is determined by taking more than average ATM deployed in Addis Ababa city i.e. 121 ATMs which is sufficient compared to other similar studies which use more than 60 objects. This data is maintained in MySQL database to facilitate searching.

The geo-location data for the selected 121 ATMs from Bank of Abyssinia is obtained by observations using the web-based commercial Google Map software as there is no digital ATM location data that contains latitude and longitude in the respective bank website. All Addis Ababa city districts are considered to capture the ATMs geolocation using Google Map. This data is maintained in MySQL database to facilitate searching. Representative sample taken from all districts are presented in table 1-1.

Table 1-1: sampling of ATMs data

District	Sites	No of ATM
Central	Piassa, Sidistkilo, Megenagna, Kara, Legehar	23
East	Bole, Gerji. Saris, Akaki, Kality	56
West	Merkato, abinet, Lebu, Jemo, Ayer Ten	42
	Total	121

This study also uses purposive sampling method to collect three months ATM data from September 31,2019 to November 31, 2019 due to the fact that the transaction volume is high compared to the previous months which is significant for this study. The total number of three months' records obtain from databases are 45,243 ATM, whereas 10,890 ATM records are collected from field maintenance engineer daily report for the same three months. We use purposive sampling because of the increase of the volume of transaction which is significant for this study. Bank of Abyssinia is selected because it is convenient and time saving for the researcher to get relevant data. Using purposive sampling methods helps to capture records of those selected ATMs whose geographic data considered in this research. The procedures that is used while obtaining ATMs data from Bank of Abyssinia ATM database are: first the relevant data is identified from the database. Second, using SQL script, the ATM data is exported in excel format. The procedure used to capture field report ATM data are: first identify relevant data, then using filtering mechanism from the daily report system, the data is exported in excel format.

Google map and OSM are the well-known map data providers among others. This study collects OpenStreetMap (OSM) data of Addis Ababa city road from online http:\\www.extract.bbbike.org. Because OSM is and open source software which any one can use data for free. The procedures that is followed while obtaining roads map of Addis Ababa city are: first, identify where the data source url (http:\\www.extract.bbbike.org) and then download the data in shape file format.

1.6.3 Implementation Tools

The prototype consists of user interface, business logic and backend. The backend is developed using MySQL and business logic and user interface is developed using JavaFx. MySQL is selected because it is lightweight and flexible to integrate with JavaFx applications. JavaFx API is selected because it has rich user interface in Java programing technology which can easily be converted to web and mobile platform.

We-based commercial version of Google Map is used to collect ATM geo-location data as it is user friendly to find and get the exact location of the ATMs than another free map application such as OSM map.

JavaFX technology is used to develop the prototype application user interface to make it compatible with the road routing software and it has rich user interface.

Java SE programming technologies is used to develop the server-side business logic such as to obtain active ATM prediction information, to calculate shortest distance, to merge the ATM geo-location data with the predicted active ATM, implement finding shortest path between user location and recommended active ATM of different algorithms; and to link the user location with recommended active ATM on the road. This study also uses Java SE to preprocess ATM and OSM data to make smooth or compatible for the road route experiment software.

There are various experiment software's that provide routing using road network. Graphhopper, pgrouting and GoogleMaps are the common open source software. Graphhoper has user interface and can perform the routing with great performance using real world map. But, it is resource intensive which requires a minimum of 64GB RAM to run the application.

Pgrouting is a command line based lightweight tool that provide the routing task with optimal performance. But, it is not flexible to customize the source code as per the experiment requirement.

GoogleMaps is another open source variant of the commercial Google Map software developed for academic purpose to provide routing service using road network. It is GUI based desktop application developed using Java, flexible to customize the source code as per the experiment requirement, less resource intensive than Graphhooper. For this study we use GoogleMaps since less resource intensive (lightweight) compared to Graphhooper tools and flexible to customize code and better GUI than pgrouting tool.

Desktop version of GoogleMaps is a tool developed for academic purpose. In this study, it is used to experiment road routing and straight-line distance algorithms as it is a free open source and easy to use academic software. It is developed using JavaFx technology. The academic version of GoogleMaps tool is selected because it is lightweight and less resource intensive tool than the other tool i.e. Graphhoper which is heavyweight that requires 64GB RAM to run the application.

ArcGIS and QGIS are the most commonly software to manipulate GIS data i.e. OSM. ArcGIS is a licensed software which requires fee per the agreement whereas, QGIS is an open source software which any one can use it for free. Therefore, in this study QGIS is used for pre-processing the OSM data as it is open source tool which is available for free, easy and flexible to use.

WEKA tool is used to experiment ATM data, to construct classification model. Weka is a knowledge discovery tool [19] that is easy to do experiment, easy to integrate with other external applications [19] and the researcher also familiar with it.

1.6.4 Evaluation procedure

In this study three major tasks done to achieve the objective; predicting active ATM, determine nearest ATM and determining the optimal route, and then develop the recommender system. Evaluation on each experiment is done using appropriate metrics.

For the purpose of constructing predictive model that determines active ATMs, two classification algorithms such as J48 decision tree and PART rule induction are used. To evaluate the performance of each algorithm the well-known recall, precision, f-measure and accuracy are considered as key parameters.

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate [20]. To calculate precision equation 1.1 is used.

$$\text{PRECISION} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \dots\dots\dots\dots\dots\dots (1.1)$$

Equation 1-1:Precision formula

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class [20]. To calculate recall equation 1.2 is used.

$$\text{RECALL} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \dots\dots\dots\dots\dots\dots (1.2)$$

Equation 1-2:Recall formula

F1 score is the weighted average of Precision and Recall. This score takes both false positives and false negatives into account, and it is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall [20]. Here under equation 1.3 shows how to compute F1 score.

$$\text{F1 SCORE} = \frac{2(\text{RECALL} * \text{PRECISION})}{(\text{RECALL} + \text{PRECISION})} \quad \dots\dots\dots\dots\dots\dots (1.3)$$

Equation 1-3:F1 score formula

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model [20]. Accuracy is computed using equation 1.4.

$$\text{ACCURACY} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad \dots\dots\dots\dots\dots\dots (1.4)$$

Equation 1-4:Accuracy formula

In this study, for determining the nearest ATM using Euclidean distance, City-block distance and Haversine distance algorithms, two parameters are used i.e. distance and processing time. Distance tells how long the distance from start to destination node. Processing time implies the time that take to complete the task.

In this study, for identifying the optimal route three road network algorithms; A* (informed) and uninformed (Dijkstra's and BFS) are used in the experiment. Common metrics have been used to evaluate the performance of searching algorithms such as distance [21], processing time [22] and selected path [21] has been used as key parameter to evaluate how the three algorithms perform. Distance is the length from start road (start node) to destination road (goal node) and measured in terms of kilometer unit. Processing Time implies how long it takes to process searching the shortest path from starting to destination road and measured in terms of millisecond. Node count is another criterion used to count the nodes or the intersection points that is used to construct the path from start to destination nodes.

This research also employs purposive sampling to select respondents to evaluate the prototype application. We use purposive sampling due to the fact that those selected respondents are required to be aware of computer technologies to catch up the demonstration so that they can easily evaluate the application. The sample size for evaluation is thirteen respondents, because similar consulted researches [23], [15] and [24] have used six, seven and twenty four respondents respectively to evaluate the prototype applications. To conduct user acceptance testing, ISO standard is adopted [25], to prepare questionnaires for collecting user feedback about the application. The procedures we followed are; first, a questionnaire is prepared on the subject matter for all selected users. Second, convenient time for the respondents are arranged. Third, quite class room having projector is selected for the demonstration. Fourth, a hands-on demonstration on the application is given for the respondents on the schedule date and time. Fifth, the questionnaire papers are distributed for the respondents to evaluate and finally, collect the paper upon finishing it and analysis on it proceed. Ten questionnaires are prepared with respect to cost, effort, time and usability; which are related to efficiency, and effectiveness. Appendix I presents the details of the questionnaire.

A questionnaire is also prepared to collect preliminary data with respect to ATM service in Ethiopia, and customer preferences using two domain expert respondents. It has three parts. The first part is about respondent's background. The second part of the questionnaire helps to capture information related to the general ATM service in Ethiopia. It consists of eight question with fill in the blank and yes/no type. The third part of the questionnaire is about ATM and radius data which we use to develop content profile (ATM service and radius) for the recommender system. (refer appendix IV user/customer preference section).

1.7 Operational definition

The following terms definition are purposefully contextualized for this research.

- Active ATM is an ATM that gives service after the prediction model identified them as in-service and propose them for use by the customer in the recommender system. Alternatively, also called available, availability, up, uptime, and online are used interchangeably with in service when the independent self-service machine ATM can make electronic payment successfully.
- Inactive ATMs are those ATMs that are rejected by the predictive model and are not proposed to the customers to be used in the recommender system. Non-functionality of different inside and outside components of the machine itself; hardware parts, communication, application (software), operating system, and unresponsive remote server. Terms such as unavailability, downtime, and down are also used interchangeably with out of service when the independent self-service machine ATM can't make electronic payment successfully.
    - Active ATM Recommender is a recommender system that recommends nearest active ATM similar and/or related to user preference by developing content profile (ATM and radius).

1.8 Ethical consideration

The researcher is responsible to protect the respondents right (keep safe personal profile, like name) and interest who have been selected to evaluate the artifact of this research while reporting and dissemination of the data.

1.9 Thesis organization

This thesis is organized in to seven chapters. The first chapter introduces the thesis main themes such as it begins by introducing the subject matter, discuss statement of the problem, explain objective of the study, discuss scope of the study, elaborate significance of the study, and methodology.

The second chapter presents literature review which highlights conceptual review and related works emphasizing on approaches of route selection and planning using road network, shortest distance calculation, and discussed classification algorithms. On the other hand, related works presented, based on which we showed research gaps.

The third chapter discuss about methods and approaches of the research followed such as architecture, predicting active ATM, determining nearest ATM and determining shortest path.

The fourth chapter is about problem domain understanding, data understanding, and data preparation for constructing a predictive model that determines the status of the ATMs. In this chapter domain and data understanding discussion is also presented to determine the nearest active ATM and optimal path.

The fifth chapter is about experimental result of classification algorithms for predicting active ATMs. This chapter discuss experiment set up, KDD models designing, perform experiment, carried out to design predictive data model for active ATMs and evaluation. This chapter also presents experimentation result towards determining nearest active ATM, and optimal path finding along with the evaluations.

Chapter six discuss about the prototyping application integration, and user acceptance testing. Chapter seven is about conclusion and recommendation for further researches.

Chapter 2

Literature Review

2.1 Overview

This chapter is organized as follows. In section 2.1, an overview of ATM and active ATM monitoring presented. Section 2.2 introduces in detail location-based services. Next, section 2.3 shows the techniques and algorithms for predicting active ATM, section 2.4 deals with recommender system, section 2.5 deals with related work and finally section 2.6 deals research gaps that this study attempt to address.

Automated Teller Machine (ATM) is a shared information system that connects private banks, government banks and other finance sectors with retail card banking customers to carry out banking transactions such as cash withdraw, fund transfer, payment, deposit and account balance inquiry [3]. ATM provides bank customers with 24-hour access to banking products/services; they are easy to use and are faster than human tellers in the banking halls. ATM services have improved the organizational efficiency of banks and customer's services in the banking sector [3].

ATM or active ATM monitoring is the process of monitoring the health of the ATM terminal and the various hardware devices installed to it [4]. ATM Monitoring also includes keeping a track of transactions taking place on ATM terminal. The main need of ATM monitoring is maximizing the availability of ATM terminals to the end users (i.e. customers) of financial institutions and reducing the downtime of the terminals. The reason behind this is that downtime of the ATM terminals makes the customers of financial institutions visit the ATM of other financial institutions for doing transactions such as Cash Withdrawals. In such kind of off-us transactions, the financial institution has to pay certain amount as a transaction fee to the other one. Another major concern of having a down time is that if the ATM terminal is having a down time frequently, it adversely impacted the reputation of the financial institution causing the loss of its existing customers and difficult to acquire new customers [4].

Bank ATM service provides 24/7 service. This makes the customer to use this service at any time, which increase the customer service satisfaction with the bank.

The need for location aware active ATM service is to provide information about the condition of the ATM by predicting ahead at the customer residence or customer current location. Therefore, it makes easy for the customer to decide where and when to get the service since the information is provided on-time or ahead.

2.2 Location based service

Location Based Service (LBS) provide information by considering the user's current location. Location based service needs as a core component Geographic Information System (GIS) to work properly [26].

The growing demand for location-based application is high due to the advent of GoogleMap with location positioning system such as GPS [26]. In this contemporary world, it is easy to use map information by linking GPS receiver to PC and PDA [26]. Application that use the location of the user are getting widespread as a result of the embedding of GPS receivers to mobile phones. Many devices are now included GPS chips to analyze signal and capture the user's current position with high accuracy. In a big conference whose privacy is not an issue, people gather to exchange ideas with each other are likely to release their location information. The location server can easily get and track the user location when the user register to join the conference. Therefore, it becomes economical to manage location detection and tracking of such big conferences using server-centric mode. The use and availability of positioning system is one of the major technological advances for the development of location-based applications [26].

Location based service address various problems in different domain areas including the following [9], [10], and [11]: identify the nearest emergency center for those who are in emergency, identify the nearest hospital for patients, identify the nearest ATM for card banking customers, and show the shortest route with less traffic congestion for the community.

Three steps carried out in using Location based services [26]. The first step is to send location information for remote parties for example, location tracking application. The second step is to make communication decision using location information for example, user agent may automatically disable instant messaging when driving. The third step is initiating communication action as location change happens. This may be when a person's user agent gets a location notification indicating the person enters a room, the user agent may automatically turn on the light of the room [26].

A location-based research conducted by Enayat et al. [10] have employed mobile device, SDLC, GIS, Google map; various system models i.e. use case diagram, sequence diagram; and survey questionnaire has been used to evaluate the system. According to the authors [10], the evaluation result shows that 95% of the respondents express this system as useful and important, whereas, 50% of the respondents express the system as critical to emergency use [10].

The work of Swapnil et al. [9] on location-based service employed Google map, Google web services, GIS and 3-tier system architecture such as client side, business layer and back end. According to the authors, they have used JQuery, Phonegap (Apache Cordova), Html 5, JavaScript, CSS for the development of the client side. They used PHP for server-side implementation and MYSQL for backend to store data. The haversine equation is used to compute the distance between all nearest search places from user current geolocation by using geo-coded address form emergency database search current address and provide all services under required distance [9].

Nilima et al. [27] conducted research on location-based services (LBS) by integrating data mining. They employ Apriori and Frequent Pattern-Growth (FP) algorithms and made comparison with respect to the time required for the execution of frequent large dataset discovery. They develop a location aware recommender system using FP-Growth and Haversine formula [27].

2.2.1 Steps in determining the nearest ATM

There are two steps to determine the nearest ATM to user's location [8]. The first step is defining the nearest ATM distance, whereas the second step is identifying the optimal path to reach to the nearest ATM destination.

2.2.1.1 Defining the nearest ATM distance

Distance between two points can be computed with different techniques available, our main aim is to pick up a proper technique from the available ones. With respect to defining the nearest ATM for the user, three shortest distance determination algorithms such as Euclidian [28], City-block [29] and Haversine algorithms are experimented [30]. Finally, of the three experimented algorithms, one optimal algorithm that fit the objective of this study is considered in the integration of the prototype application.

Euclidean distance computes the root of square difference between co-ordinates of pair of objects, whereas, city-block distance or Manhattan distance computes the absolute differences between coordinates of pair of objects [28]. On the other hand, Haversine calculates by considering the length of two points on the surface of the earth based on latitude and longitude [28].

City Block Distance

City Block Distance between any two points a and b with n dimensions is easily calculated as shown in equation 2-1 [29]:

$$\sum_{i=1}^{k} |aj - bj| \quad \dots\dots\dots\dots\dots\dots (2.1)$$

Equation 2-1: City-block distance

The City block distance is instead calculated as the distance in x plus the distance in y, which is similar to the way we move in a city (like Manhattan) where we have to move around the buildings instead of going straight through [29]. This distance is always greater than or equal to zero. For identical points, the measurement would be zero and for the points that show little similarity, the measurement is high. It is important to notice here that the influence of a large difference in a single dimension is dampened for most of the times [31].

Euclidean distance

The Euclidean distance between two points, a and b, with k dimensions is calculated as [29].

$$= \sqrt{\sum_{i=1}^{n} (qi - pi)^2} \quad \dots\dots\dots\dots\dots\dots (2.2)$$

Equation 2-2:Euclidean distance

Euclidean (and squared Euclidean) distances are usually computed from raw data, and not from standardized data. One advantage of this method is that the distance between any two objects is not affected by the addition of new objects to the analysis, which may be outliers [29]. However, the distances can be greatly affected by differences in scale among the dimensions from which the distances are computed. For example, if one of the dimensions denotes a measured length in centimeters, and you then convert it to millimeters (by multiplying the values by 10), the resulting

Euclidean can be greatly affected (i.e., biased by those dimensions which have a larger scale), and consequently, the results of cluster analyses may be very different. Generally, it is good practice to transform the dimensions so that they have similar scales [29].

Haversine formula

Haversine requires four variables such as (lat1, lng1, lat2, lng2) to calculate straight line distance. It is a method of knowing the distance between two points by considering that the earth is not a plane but is a plane of a degree of curvature and has a radius of 6,367.45 km” [30]. Figure 2.1 shows Spherical triangle solved by the law of Haversine.

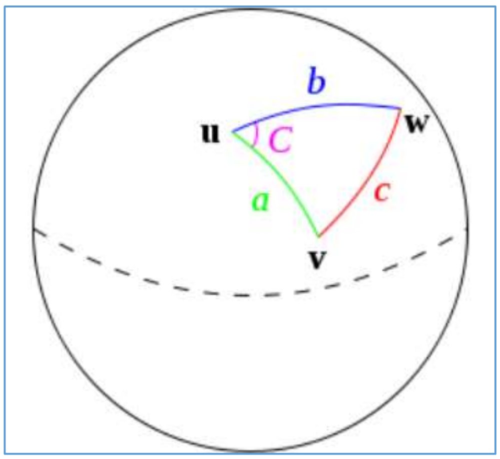

Figure 2-1: Spherical triangle solved by the law of Haversine [25]

The working of Haversine algorithm is to establish a direct distance between points that can be stretched in a triangular form where a, b, and c are the distances to be calculated. The following formulas (equation 2-3) are the way to perform Haversine algorithm [30]:

$$hav\left(\frac{d}{r}\right) = hav(\varphi 2 - \varphi 1) + \cos(\varphi 1)\cos(\varphi 2)\, hav(\lambda_2 - \lambda_1) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

Equation 2-3: Haversine distance

Where: hav() is the haversine function, with d is the distance between the two points (along a great circle of the sphere; see spherical distance), and r is the radius of the sphere. Parameters φ1, φ2: latitude of point 1 and latitude of point 2, in radians, and λ1, λ2 are longitude of point 1 and longitude of point 2, in radians.

On the left side of the equal sign, d/r is the central angle, assuming angles are measured in radians (note that, φ and λ; can be converted from radians to degrees by multiplying by 180/π as usual). The following formula is used to solve by applying the inverse Haversine (if available) or by using the arcsine (inverse sine) function [30]:

$$d = 2r.arcsin\left(\sqrt{sin^2\left(\frac{\Delta \text{latt}}{2}\right) + \cos(latt1)\cos(latt2)\,sin^2\left(\frac{\Delta long}{2}\right)}\right) \ldots \ldots (2.4)$$

Equation 2-4:inverse Haversine formula

To perform distance calculation using Haversine formula, we need the location of each data event. The latitude, longitude, and the radius area of the event are needed for calculating distance. This information will be entered into database [27].

2.2.1.2 Identifying the optimal path to the nearest ATM destination

With respect to shortest path/route findings, various informed and uninformed search algorithms are discussed [32] and [33] such as A* search, Dijkstra's and greedy search.

A* search

As noted by Norvig [32], the most widely-known form of best-first search is called A* search (pronounced "A-star search") in path finding. It is the process of finding a path between multiple points, called "nodes". To determine the shortest route, it evaluates nodes using cost function f(n) that combines the past path-cost function, g(n) and a future path cost function, h(n). f(n)=g(n)+h(n). Since g(n) gives the path cost from the start node to the current node n, and h(n) is the estimated cost of the cheapest path from the current node to the goal, we have f(n) that provide estimated cost of the cheapest solution through n [32]. It enjoys widespread use due to its performance and accuracy.

Dijkstra's algorithm

Dijkstra's algorithm solves the single-source shortest-paths problem on a weighted, directed graph, G = (V, E) for the case in which all edge weights are nonnegative [33]. Dijkstra's algorithm maintains a set S of vertices whose final shortest-path weights from the source s have already been determined. The algorithm repeatedly selects the vertex u ∊ V - S with the minimum shortest-path estimate, adds u to S, and relaxes all edges leaving u [33].

Breadth-first search

One of the simplest algorithms for searching a graph and the archetype for many important algorithms that has similar concepts with Dijkstra's single-source shortest-paths algorithm and Prim's minimum-spanning tree algorithm is breadth-first search algorithm [33].

Breadth-first search finds the edge of G to "discover" each vertex that is reachable from s given a graph G = (V, E) and a distinguished source vertex s. Breadth-first search computes the distance (smallest number of edges) from s to each reachable vertex. Breadth-first search also produces a "breadth-first tree" with root s that contains all reachable vertices. For any vertex v reachable from s, the simple path in the breadth-first tree from s to v corresponds to a "shortest path" from s to v in G, that is, a path containing the smallest number of edges. The algorithm works on both directed and undirected graphs [33].

2.3 Predicting active ATM

Explosive data growth in every day from business, society, science, engineering, and medicine requires a powerful tool to transform such data into organizational knowledge. This necessity has led data mining to be born [34].

Numerous authors had done work to discover interesting knowledge using data mining. Data mining refers to extracting or "mining" knowledge from large amounts of data [34].

Active ATM, also called in-service ATM or up ATM which gives usual service for customers. In this case, failed ATMs or out of service ATMs are ignored as it doesn't give usual service for the customer. Active ATM status information is obtained by undertaking various pre-processing activities of ATM data and designing predictive model.

Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically [34]. The next section discusses various models and tasks in data mining.

2.3.1 Knowledge discovery process model

The knowledge discovery process (KDP) seeks new knowledge in some application domain. It is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. The process generalizes to non-database sources of data, although it emphasizes databases as a primary source of data. As shown in figure 2-2, it consists of many steps (one of them is DM), each attempting to complete a particular discovery task and each accomplished by the application of a discovery method. Knowledge discovery concerns the entire knowledge extraction process, including how data are stored and accessed, how to use efficient and scalable algorithms to analyze massive datasets, how to interpret and visualize the results, and how to model and support the interaction between human and machine. It also concerns support for learning and analyzing the application domain. Figure 2-2 shows KDP model.

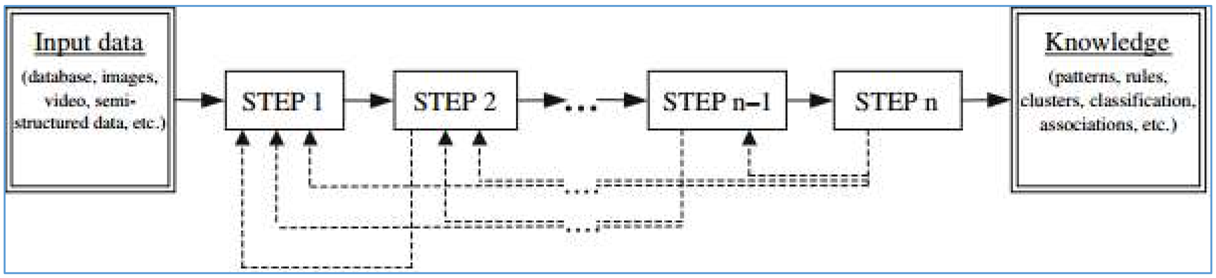

Figure 2-2: Sequential structure of the KDP model [30]

Although the models usually emphasize independence from specific applications and tools, they can be broadly divided into those that consider industrial issues and those that do not.

2.3.2 Data mining tasks

Data mining involves many different algorithms to accomplish different tasks [35]. All of these algorithms attempt to fit a model to the data. The algorithms examine the data and determine a model that is closest to the characteristics of the data being examined [35]. Data mining algorithms can be characterized as consisting of three parts [35]:

- ✓ Model: The purpose of the algorithm is to fit a model to the data.
- ✓ Preference: Some criteria must be used to fit one model over another.
- ✓ Search: All algorithms require some technique to search the data.

The model that is created by data mining can be either predictive or descriptive in nature [35]. Predictive data mining task uncover unknown pattern based on predictor dataset [35]. On the other hand, descriptive data mining characterizes the general properties of the data. The revealed pattern will be a model. Predictive models determine the desired application domain future trend decision. Predictive models are supervised learning because the class label identified before data analysis. Descriptive models are unsupervised learning because there is no class label considered for the data analysis. Clustering of data, Summarization, Association rules, and sequence discovery are considered as descriptive.

Classification algorithm

Classification is referred to as supervised learning because an example data set is used to learn the structure of the groups, just as a teacher supervises his or her students towards a specific goal. While the groups learned by a classification model may often be related to the similarity structure of the feature variables, as in clustering, this need not necessarily be the case [19]. In classification, the example training data is paramount in providing the guidance of how groups are defined [19]. Given a data set of test examples, the groups created by a classification model on the test examples will try to mirror the number and structure of the groups available in the example data set of training instances [19].

The goal of classification is to construct a model using the historical data that accurately predicts the label (class) of the unlabeled information [36].

The Data Classification processes involves two steps [35], building the classification model and using the model for classification. The first step is the learning phase. In this step the classification algorithms build the classifier. The classifier is built from the training set made up of database instances and their associated class labels. Each instance that constitutes the training set is referred to as a category or class. Then, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data instances if the accuracy is considered acceptable.

Various classification algorithms use different techniques for finding relations between the predictor attribute's value and the target attribute's value in the built data [36]. This study considers classification algorithms such as Decision Tree and Rule Induction.

Decision tree

Decision trees are a classification methodology, wherein the classification process is modeled with the use of a set of hierarchical decisions on the feature variables, arranged in a tree-like structure [37]. The decision at a particular node of the tree, which is referred to as the split criterion, is typically a condition on one or more feature variables in the training data [37]. The split criterion divides the training data into two or more parts.

Decision tree induction is the learning of decision trees from class-labeled training tuples [34]. A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node [34].

During tree construction, attribute selection measures are used to select the attribute that best partitions the tuples into distinct classes [34]. When decision trees are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data [34]. In rule based, if the condition (i.e., all the attribute tests) in a rule antecedent holds true for a given tuple, we say that the rule antecedent is satisfied (or simply, that the rule is satisfied) and that the rule covers the tuple. A rule R can be assessed by its coverage and accuracy. That is, a rule's coverage is the percentage of tuples that are covered by the rule (i.e., their attribute values hold true for the rule's antecedent). For a rule's accuracy, we look at the tuples that it covers and see what percentage of them the rule can correctly classify.

Tree induction follow a top-down approach, which starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built [34].

- The algorithm is called with three parameters: D, attribute list, and Attribute selection method. D is referred as a data partition. Initially, it is the complete set of training tuples and their associated class labels. The parameter attribute list is a list of attributes describing the tuples. Attribute selection method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class. This procedure employs an attribute selection measure such as information gain or the Gini index. Whether the tree is strictly binary is generally driven by the attribute selection measure. Some attribute selection measures, such as the Gini index, enforce the resulting tree to be binary. Others, like information gain, do not, therein allowing multiway splits (i.e., two or more branches to be grown from a node). Decision tree algorithm presented next.
- The tree starts as a single node, N, representing the training tuples in D (step 1).
  - Algorithm: Generate decision tree. Generate a decision tree from the training tuples of data partition, D.

- Input:
  1. Data partition, D, which is a set of training tuples and their associated class labels;
  2. attribute list, the set of candidate attributes;
  3. Attribute selection method, a procedure to determine the splitting criterion that "best"
  4. partitions the data tuples into individual classes. This criterion consists of a
  5. splitting attribute and, possibly, either a split-point or splitting subset.
- Output: A decision tree.
- Method

(1) create a node N;
(2) **if** tuples in D are all of the same class, C, **then**
(3) return N as a leaf node labeled with the class C;
(4) **if** attribute list is empty **then**
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, attribute list) to **find** the "best" splitting criterion;
(7) label node N with splitting criterion;
(8) **if** splitting attribute is discrete-valued **and**
multiway splits allowed **then** // not restricted to binary trees
(9) attribute list attribute list − splitting attribute; // remove splitting attribute
(10) **for each** outcome j of splitting criterion
// partition the tuples and grow subtrees for each partition
(11) let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12) **if** Dj is empty **then**
(13) attach a leaf labeled with the majority class in D to node N;
(14) **else** attach the node returned by **Generate_decision_tree**(Dj, attribute list) to node N;
endfor
(15) return N;

- If the tuples in D are all of the same class, then node N becomes a leaf and is labeled with that class (steps 2 and 3). Note that steps 4 and 5 are terminating conditions.
- Otherwise, the algorithm calls Attribute selection method to determine the splitting criterion. The splitting criterion tells which attribute to test at node N by determining the "best" way to separate or partition the tuples in D into individual classes (step 6). The splitting criterion also tells which branches to grow from node N with respect to the outcomes of the chosen test. More specifically, the splitting criterion indicates the splitting attribute and may also indicate either a split-point or a splitting subset. The splitting criterion is determined so that, ideally, the resulting partitions at each branch are as "pure" as possible. A partition is pure if all the tuples in it belong to the same class. In other words, if the tuples are split up in D according to the mutually exclusive outcomes of the splitting criterion, we hope for the resulting partitions to be as pure as possible.

- The node N is labeled with the splitting criterion, which serves as a test at the node (step 7). A branch is grown from node N for each of the outcomes of the splitting criterion. The tuples in D are partitioned accordingly (steps 10 to 11) [34].

J48 is the implementation of C4.5 in WEKA. C4.5 is a suite of algorithms for classification problems in machine learning and data mining [34].

Rule induction

Rules are a good way of representing information or bits of knowledge [34]. A rule-based classifier uses a set of IF-THEN rules for classification [34]. An IF-THEN rule is an expression of the form

IF condition THEN conclusion.

The "IF" part (or left side) of a rule is known as the rule antecedent or precondition. The "THEN" part (or right side) is the rule consequent. In the rule antecedent, the condition consists of one or more attribute tests that are logically ANDed. The rule's consequent contains a class prediction.

Algorithm: Sequential covering. Learn a set of IF-THEN rules for classification [34].

Input:

- D, a data set of class-labeled tuples;
- *Att_vals*, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

Rule_set ={}; // initial set of rules learned is empty

for each class c **do**

    repeat

        Rule = **Learn One Rule** (D, Att_vals, c);

        remove tuples covered by *Rule* from D;

        Rule_set = *Rule_set* + *Rule*; // add new rule to rule set

    until terminating condition;

endfor

return *Rule_Set*;

PART is the implementation of rule induction algorithm in WEKA. PART algorithm combines the divide-and-conquer strategy (the top-down approach) for decision tree construction with the separate-and-conquer approach for rule learning. The separate-and conquer strategy first builds a rule and then removes those instances that the rule covers. These consecutive activities continue recursively for the remaining instances until none are left which generates sets of rules called 'decision lists' or ordered set of rules. On the other hand, in the partial decision tree, a pruned decision tree is built for part of the training instances, the leaf with the largest coverage is made into a rule, and the tree is discarded. Using partial decision trees in conjunction with the separate-and-conquer methodology adds flexibility and speed. A partial decision tree is an ordinary decision tree that contains branches to undefined sub trees. During the generation of such a tree, construction and pruning operations are integrated in order to find a "stable" sub tree that cannot be simplified further. Once this sub tree has been found, tree building ceases and a single rule is read off [38].

2.4 Recommender system

Recommender system aimed at suggesting relevant items to users seeks to rate or predict the "preference" a user would give to an item [39]. It is also a subclass of information filtering system that predict the user preference and propose an item that match or close to the user needs [39].

Ever since the popularization of web-based transactions, it has become increasingly easy to collect data about user buying behaviors. This data includes information about user profiles, interests, browsing behavior, buying behavior, and ratings about various items. It is natural to leverage such data to make recommendations to customers about possible buying interests [37].

In the recommendation problem, the user–item pairs have utility values associated with them. Thus, for n users and d items, this results in an $n \times d$ *matrix* D of utility values. This is also referred to as the utility-matrix. The utility value for a user-item pair could correspond to either the buying behavior or the ratings of the user for the item [37]. Typically, a small subset of the utility values is specified in the form of either customer buying behavior or ratings. It is desirable to use these specified values to make recommendations [37]. The nature of the utility matrix has a significant influence on the choice of recommendation algorithm [37]:

1. Positive preferences only: In this case, the specified utility matrix only contains positive preferences. For example, a specification of a "like" option on a social networking site, the browsing of an item at an online site, or the buying of a specified quantity of an item, corresponds to a positive preference. Thus, the utility matrix is sparse, with a pre-specified set

of positive preferences. For example, the utility matrix may contain the raw quantities of the item bought by each user, a normalized mathematical function of the quantities, or a weighted function of buying and browsing behavior. These functions are typically specified heuristically by the analyst in an application-specific way. Entries that correspond to items not bought or browsed by the user may remain unspecified [37].

2. Positive and negative preferences (ratings): In this case, the user specifies the ratings that represent their like or dislike for the item. The incorporation of user dislike in the analysis is significant because it makes the problem more complex and often requires some changes to the underlying algorithms [37].

Generally, recommendation system models are classified as content-based recommendation and collaborative filtering [37].

In content-based recommendations: the users and items are both associated with feature-based descriptions. For example, item profiles can be determined by using the text of the item description. A user might also have explicitly specified their interests in a profile. Alternatively, their profile can be inferred from their buying or browsing behavior [37].

Collaborative filtering, on the other hand, is the leveraging of the user preferences in the form of ratings or buying behavior in a "collaborative" way, for the benefit of all users. Specifically, the utility matrix is used to determine either relevant users for specific items, or relevant items for specific users in the recommendation process. A key intermediate step in this approach is the determination of similar groups of items and users. The patterns in these peer groups provide the collaborative knowledge needed in the recommendation process [37].

I. This study design location aware active ATM recommender system for banking service. ATM and radius profile developed by collecting data using questionnaire (refer appendix IV Customer/User preference section). The data is collected using questionnaire from domain expert due to the fact that there is no data prepared for this purpose. Due to the nature of the data collected for this study, the types of recommender that fit for this study is content-based recommender. Hence, this study considers content-based recommender.

2.5 Related work

There are different researches that attempt to predict ATM failure, and also to design location aware ATM services. These studies are reviewed and presented as follows.

2.5.1 Foreign Works

Cheong et al [40] attempt to forecast ATM failure using decision analytics methodologies to perform better ad-hoc ATM failure forecasting and plan the field service engineers to repair the ATM machines. This paper reports the work in analyzing past daily ad-hoc ATM failures, forecasting ad-hoc ATM failures using Stepwise Autoregressive, Exponential Smoothing and Holt-Winters Additive model. Using the forecasted results to optimize the number of field service engineers to deploy in each geographical zone, to minimize the number of daily unattended ad-hoc ATM failures. They show that, the optimization model ensures that the least number of engineers are deployed in each zone on each day. However, to maintain a consistent number of engineers for a 2-week schedule, they recommend to deploy the maximum number of engineers in each zone within the 2 weeks. According to the authors, the resulting surplus engineer idle hours is reduced, and it represents a cost savings of 28.6% when compared with the bank's current practice.

Rachburee et al. [41] predict the amount of failure symptoms and compare the performance between artificial neural network and support vector machine in ATM spare part inventory forecasting. According to the authors, this study uses three years' data from January 2013 to March 2016 of one hundred thousand ATM incident log data. They perform data cleaning and pre-processing task to filter out relevant data such as preventive and corrective ATM log data for the study. They use data exploration method to identify the trends and seasonality to help identify working and non-working days' (i.e. Saturday and Sunday) data which in this case they have removed as it is not useful for their study. They also use time series moving average data set and establish a windowing to define the time period such as 5 and 7 days which is weighted by Hann function. They separate the data set into two groups such as training data set which is composed of 2013 – 2015 data and testing data set which is composed of 2016 data. They have set parameters in the rapidminer (V.7.1) tool such as 500 training cycle with learning rate of 0.3 and a momentum of 0.2. The result of this study shows that SVM has higher average accuracy of 82.24% than ANN i.e. SVM for regression generated the highest prediction accuracy rate of 92.7% expense parts (components). The authors work also shows that SVM for regression is a satisfied method for predicting the amount of failed spare parts. The authors recommend association rule techniques to improve predictive ATM maintenance as a future work.

Gugapriya et al. [26] has carried out a research attempt to identify nearest ATM center location [26]. This study employs Android, GPS and Google Map. According to the researcher, this study provides a new way to access the bank transaction by android mobiles. The researcher also stated, it makes easy to access the banking transaction in ubiquitous access by identifying Nearest ATM Centers by using GPS along with transaction [26]. However, this study has limitation with respect to finding in-service ATMs and route selection to reach to the destination.

The work of Mario et al. [42] focus to give android users the ability to find ATMs quickly. This work employs Android as programming language, Google Map API to present location on map, XML to create android user interface, Parse.com to store collected bank data as well as geo-coordinates of ATMs location locally in JSON format and Genymotion emulator to test the application in emulator and in physical devices. The paper illustrates that, Bank and ATMs geo-location data is obtained from official 20 banks documents and websites [42]. According to the paper the key findings are: enable finding of ATMs quickly, provide shortest path to selected ATM on foot or on car or on public transport options with tips where money can be withdrawn free of charge [42]. The authors recommend voice search on Croatian language and offline map for future work [42].

Rajib et al. [6] conducted a research to find the nearest ATM so as to address ATM location problem faced by the customer when the need for cash is high during emergency or when tourists who is new to the area faces cash shortage. This study employ android, OpenStreetMap, and GIS platform. According to the authors, this work allows customers to see list of ATMs by bank, recommend the nearest ATM along with the shortest path on the map, show marker on the map along with textual information, and allow volunteer person to add new ATM location. The researchers evaluate the system by comparing with the existing actual road distance [6]. Few of the limitation pointed out by the researchers are: from the result of the research evaluation, the system has accuracy problem, usability by the user and shortage of developers are problems observed on the OpenStreetMap API, OpenStreetMap API allows editable map which could result in accepting or adding redundant map information, high speed internet connection requirement to use OpenStreetMap API. Moreover, this work is not address ATM availability condition information to the customer.

A research carried out by Faried et al. [7] attempted to construct a model for search application using mobile device which implement location based augmented reality (AR) method. This work employ android, web, Google map and GIS platform. This study also uses secondary data to capture ATM and bank details as well as the system is designed as client-server architecture. This study come up with a model that allows customer to search ATM location in the neighborhood, allow navigation and augmented reality, provide ATM availability condition information [7]. The author recommends shortest path algorithm for searching location and also the mechanism to prevent the fake information from the user for further study.

Bharath et al. [16] conducted a research to help customers by providing nearby ATMs information to help customer withdraw cash without delay. This study employs a separate hardware called "ATM Locator device". According to the authors, the device recommends the nearest ATM along with ATM availability using map on the device as well as in text form. It also suggests alternative road when traffic congestion occurs [16]. However, the limitation of this work is: the device doesn't support local Wi-Fi internet connection, requires configuration and installation of utility software which could be difficult for the customer, requires training about the device hence this this could be an extra cost for the candidate bank who is interested to implement.

2.5.2 Local Works

Fasika [15] attempt to find reasons for ATM out of service. This study uses three months of 14 ATM monitoring system data from Dashen Bank S.C. This study performs data pre-processing activities such as construct new attributes from the existing attributes, remove irrelevant attributes to make the data suitable for the data mining. This research uses WEKA tool and experiment using J48 decision tree, PART rule induction, Naïve Bayes and Multi-layer perceptron, Neural Network algorithms to discover knowledge. The result illustrates that 73% of failure is hardware such as dispenser, card reader and network device failure, whereas 27% failure is software failure which associated with transaction and application. This study shows J48 decision tree algorithms has better performance compared to the others and used to discover knowledge to be consumed by prototype system. In this study prototype application is developed using Java to help users know ATM failure based on the knowledge discovered by J48 model. This study performs evaluation on the prototype system and the result shows that the system is effective, efficient and satisfactory and the respondent agreed to accept and use it. This study shows the possibility to integrate the result of WEKA tool with prototype application and also recommends to try with more datasets, and making ATM not to run empty of cash proactively by constructing a predictive model are some of the future work recommendation.

2.6 Research Gap

Different researches have been conducted on location-based ATM services. Those researches allow customer to get the nearest ATM on the map [6], suggest shortest road route or path on the map [16], identify in-service ATMs [7], navigation and augmented reality [7], suggest alternative road when traffic congestion occurred [16], see list of ATMs by banks and allow to add new ATM location [6]. However, those works have limitation with respect to ATM status prediction information. Therefore, adopting directly those research outputs is not fit for Ethiopian banks location-based ATM services.

As to the researcher knowledge, there is no study conducted on location aware ATM recommender system that provides the nearest active ATM status prediction information for ATMs of Ethiopian banks. Hence, the card banking customers are challenged in finding the nearest active ATM along with condition of the ATM information. As a result, the customer and visitor's loss time, effort and money advantages. Besides, the banks may loss loyal customers due to inconvenience of ATM service which have impact on the revenue and face inefficient resource allocation that could incur bank for additional cost.

It is therefore the aim of this study to design location aware active ATM service recommender for Ethiopian banks.

Chapter 3

Methods and Approaches

3.1 Overview

In this study, an attempt is made to develop location aware active ATM recommender for customers focused banking sector. This chapter begin by architecture discussion in section 3.2. Next, in section 3.3 active ATM prediction discussed; determining nearest ATM elaboration followed in section 3.4; and shortest path determination elaboration presented in section 3.5.

3.2 Architecture of prototype system

In this study, an architecture is developed that recommends the nearest active ATM as per the customer preference. There are three key steps involved to recommend the nearest active ATM as per the customer preference such as predict active ATM, determine or find the nearest active ATM as per the customer preference, and determine or find the shortest path from the customer location to the nearest active ATM. Figure 3-1 shows the architecture of the study.

This architecture is developed by integrating different architectures which are adopted from data mining [34], from shortest straight-line distance [27] and optimal path finding [42] domain areas.

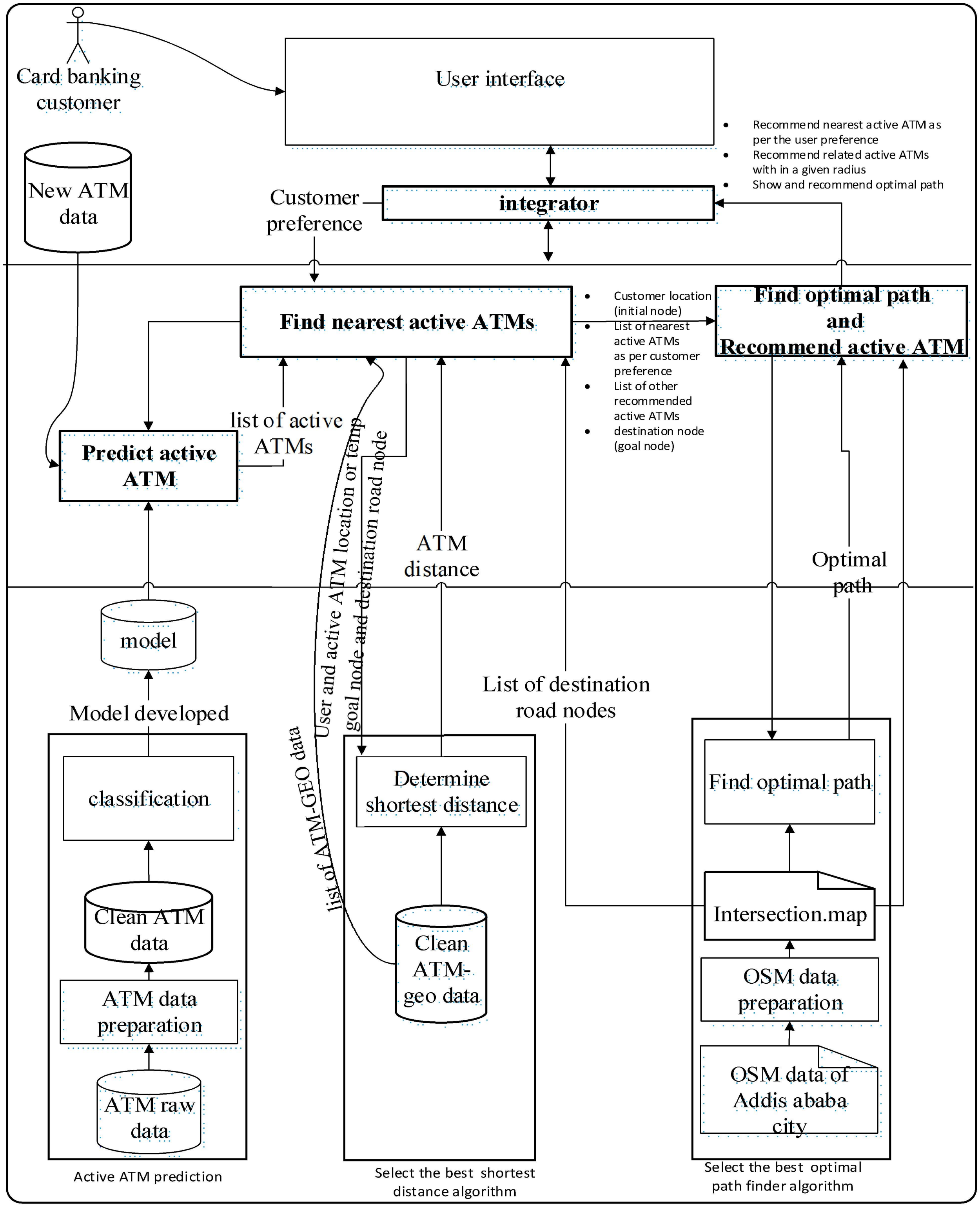

Figure 3-1:architecture of prototype system

Card banking customer is a bank customer who have information gap related to nearest active ATM as per his/her preference in the surrounding of the customer. The customer can use smart phone or personal computer to request and view nearest active ATM information via the user interface (UI) installed on the devices.

User interface (UI) bridges the customer and **Integrator** component by accepting input of the customer preference and showing an outputs information. During request it passes user location from mobile device or IP address from PC or user entered location input along with the customer preference (ATM type and radius) and submit to the **Integrator** component. It receives output from **Integrator** component and shows recommend nearest active ATMs as per the user preference, recommend related active ATMs with in a given radius, show and recommend optimal path on GMAP from user location to recommended active ATM. **GMAP** is the main input and output component of the user interface.

Integrator component is the key components where the user interface is connected with other components of the system. It accepts input from the user interface and send it to the **find nearest active ATM** component for further processing. It also receives output from the **find optimal path and recommend active ATM** component and send to the user interface..

Find nearest active ATMs component in the architecture is used to find and select the nearest active ATMs as per the customer preference. To do so, the search process involves the following steps:

1. Receive customer location, customer preferences from **integrator** component.
2. Receive list of active ATM data from **predict active ATM** component
3. Receive list of clean ATM-geo data
4. Merged active ATM data with clean ATM-GEO data
5. If the request is to find the nearest active ATM from user location to merged active ATM location, it sends customer location data and merged active ATM data to **determine shortest distance** component and receive the distance of each ATM.
6. Select one active ATM temporary goal node from the user preference.
7. Send selected active ATM location data and list of destination road nodes data to **determine shortest distance** component and receive the distance of each nodes.

8. Send the result to **Find optimal path and recommend active ATM** component.

Find optimal path and recommend active ATM component send recommendation information i.e. those active ATMs as per the user preference, related active ATMs in the given radius, and path to the **Integrator** component. It involves the following steps:

1. Receive user location, all merged active ATMs along with their distance, destination nodes with their distance and customer preference
2. Pick three active ATMs with the smallest distance that satisfy the user preference (ATM type and radius)
3. Select five other related active ATMs that resemble the user preference.
4. Select nearest node for the selected active ATM which serve as final goal node.
5. Send user location, selected nearest node, and intersection.map file data to **find optimal path** component and receive the path.
6. Send recommended active ATM as per the user preference, related recommended active ATMs, and path information to the **integrator** component.

Predict active ATM component in the architecture is responsible to predict active ATM by fetching new ATM data. It then forwards those active ATMs to Find **nearest active ATMs** component.

ATM raw data is collected for the purpose of active ATM prediction. Three-month data from September 01,2019 to November 31,2019 is used. A total of 45,343 data is collected from internal i.e. database and external i.e. different reports filled by field ATM maintenance engineer. The two data i.e. internal and external data are merged together to serve the intended purpose.

ATM data preparation is the process of cleaning noisy ATM data from the corpus. Filling missing value and feature selection and reduction are applied during data preparation. The total number of clean data that is ready for experiment after data preparation is 44,105. This activity is performed by using WEKA, excel, and java program.

Clean ATM data is a data which is used for active ATM prediction using classification algorithms. It is generated by **ATM data preparation** component of the architecture.

New ATM data is a data which we use for the prototype. It is used by **predict active ATM** component for predicting active ATM purpose.

Classification component carryout testing using J48 and PART algorithms to develop better active ATM prediction model. These algorithms use 44,105 cleaned ATM data. A model is developed using J48 classification algorithm to as it is selected as the optimal one compared to PART.

Model in the architecture is the rule or knowledge output generated from data mining classification algorithm experiment. The **model** contains the knowledge or rule along with the predicted ATM status information. To develop this model eight different experiments are carried out using J48 and PART algorithms. The final model which is integrated with the prototype application is constructed using J48 algorithm and 44,105 ATM cleaned data with WEKA tool.

Clean ATM-geo data in the architecture is a data that contains information about the ATM physical location. 121 ATM-geo data are collected using the commercial GoogleMap software. This data is used in the experiment of determining the nearest active ATM and prototype application.

Determine shortest distance in the architecture test Euclidian, city-block and Haversine algorithms. This experiment is carried out with the objective to select the optimal algorithm among them and integrate it to the prototype application. 20 clean ATM-geo data are selected randomly and employed for the experiment. Time and distance are used to evaluate them. This experiment result shows that Haversine has better performance than the others, hence selected for the integration.

OSM data of Addis Ababa city in the architecture is a map file which contains different data such as building, lake, drainage, road, football field etc. This data is downloaded from online OSM data provider. In this study, only Addis Ababa city road data is filtered for the purpose of optimal route selection experiment and for the prototype application consumption. Different data preparation is made to clean the data.

OSM data preparation in the architecture is the process of filtering out city road data that is relevant for this study. Data cleaning is done to make the data suitable for the experiment tool.

This process generates more than 107,000 road intersection data which key to carry out experiment using road network algorithms. QGIS, excel and java program tools are used for cleaning.

Intersection.map data in the architecture is the main data generated by **OSM data preparation** component. It is used for experimentation of finding the optimal route algorithms. It is also used by the prototype application. More than 107,000 cleaned intersection record is prepared. QGIS, excel and java program tools are used for preparation. **List of destination node** data is also genereated from Intersection.map file which is useful to find nearest road node (final goal node in routing) for the selected active ATM. In the data preparation duplicate nodes are removed to increase efficiency of the algorithm while searching the node. This data is useful to find the nearest road node (final goal node) for the recommended active ATM. Java program is used for preparation.

Find optimal path component experiment three algorithms such as A star, Dijkstra's and Breadth-first and select the best one. More than 107,000 cleaned intersection data is used form **intersection.map** file in the experiment. The end result shows that A star algorithm has better performance than the other two, and therefore, it is selected for integration in the prototype application.

3.3 Classification algorithms for predicting active ATM

In this study, a model is developed to predict the condition of the ATM and identify active ATMs. This model is the main input of **find nearest active ATMs** component. For constructing the predictive model, a total of 45, 343 ATM data are collected. By applying data preparation such as filling missing values and feature selection and reduction, a clean data is prepared. After the data preparation process, 44,105 clean data are generated which is suitable for the experiment. WEKA is used in for classification experiment, clustering the corpse and resampling the experiment data. Excel is used to filter missing values, and java program are used for data preparation while merging internal and external ATM data which we use for prediction.

The prediction activity is experimented using classification algorithms. Two classification algorithms such as J48 decision tree and rule induction PART are used for prediction experiment. These algorithms are selected because literatures vastly use them for similar purpose and the researcher is familiar to work with them. WEKA is employed to experiment the prediction using 44, 105 cleaned ATM data. Those algorithms are explained next.

J48 decision tree is a supervised learning algorithm. Given all attribute-valued dataset where instances are described by collection of attributes and belong to one of a set of mutually exclusive classes. J48 decision tree learn by mapping from attribute values to classes that can be applied to classify new, unseen instances [43]. Algorithm 3-1 presents how J48 decision tree works during the classification process.

The generic description of how C4.5 works is shown in Algorithm 3.1. All tree induction methods begin with a root node that represents the entire, given dataset and recursively split the data into smaller subsets by testing for a given attribute at each node. The subtrees denote the partitions of the original dataset that satisfy specified attribute value tests. This process typically continues until the subsets are "pure," that is, all instances in the subset fall in the same class, at which time the tree growing is terminated.

Algorithm 3-1:J48 algorithm

Input:
  an attribute-valued dataset D of ATM data
Output: Tree
Method:
    Tree = {}
    if D is "pure" OR other stopping criteria met **then**
        terminate
    end if
    for all attribute $a \in D$ **do**
        Compute information-theoretic criteria if we split on a
    end for
    a_{best} = Best attribute according to information theory criteria
    Tree = Create a decision node that tests a_{best} in the root
    D_v = Induced sub-datasets from D based on a_{best}
    for all D_v **do**
        $Tree_v$ = check_C4.5(D_v)
      Attach $Tree_v$ to the corresponding branch of Tree
    end for
return Tree

The other classification algorithm that is used for prediction is PART. Rule induction algorithms generate a model as a set of rules logically ANDed together to form the rule antecedent ("IF" part) and the rule consequent ("THEN" part). The antecedent consists of the attribute values from the branches taken by particular path through the tree, while the consequent consists of the class value for the target attribute given by the particular leaf node [38]. Algorithm 3-2 presents how APRT rule identification works.

A basic sequential covering algorithm is shown in Algorithm 3.2. Here, rules are learned for one class at a time. Ideally, when learning a rule for a class, C, we would like the rule to cover all (or many) of the training tuples of class C and none (or few) of the tuples from other classes. In this way, the rules learned should be of high accuracy. The rules need not necessarily be of high coverage. This is because we can have more than one rule for a class, so that different rules may cover different tuples within the same class. The process continues until the terminating condition is met, such as when there are no more training tuples or the quality of a rule returned is below a user-specified threshold. The Learn One Rule procedure finds the "best" rule for the current class, given the current set of training tuples.

Algorithm 3-2:PART algorithm *[34]*

Input:
D, a data set of class-labeled tuples;
Att_vals, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

```
Rule_set = {}; // initial set of rules learned is empty
for each class c do
    repeat
        Rule = Learn_One_Rule(D, Att_vals, c);
        remove tuples covered by Rule from D;
        Rule_set = Rule_set + Rule; // add new rule to rule set
    until terminating condition;
endfor
return Rule_Set;
```

3.4 Determining the nearest active ATM

Once all active ATMs identified, the next task is to determine the nearest ATM to the customer location. To this end we apply algorithms that determine the distance between each active and the current user location using ATM-GEO data and user preferences. Of this result, the one with small distance meets user requirement is selected as the nearest active ATM (temporary goal node). To calculate such distance, distance measure algorithms such as Euclidian distance, City-block distance and Haversine are tested using 20 records of ATM-geo data, user preference ATM and user location. The details of each algorithm discussed next.

The city-block distance computes Manhattan distance which is the absolute difference between the customer location and ATMs location using equation 2.1. The city block distance is always greater than or equal to zero. This procedure is called an absolute block or as known as city block distance [44]. Algorithm 3-3 describes how city block distance is determined.

Algorithm 3-3:City-block distance algorithm

Input:
    Vector one
    Vector other
    distance
Output: distance

Method:
    distance += absolute value of (one[dimension1] – other[dimension1])
        + absolute value of (one[dimension2] – other[dimension2])
return distance

The other algorithm used in this study is Euclidian distance. Euclidian distance is the straight-line distance between two points. The Euclidian distance between customer location and ATM location is the length of the line segment connecting them. Here under algorithm 3-4 shows how Euclidean distance works.

Algorithm 3-4:Euclidian distance algorithm *[45]*

Input:
Point one
Point other
distance
Output: distance

Method:
distance += square root of ((other [dimension1] – one[dimension1]) ×
((other [dimension1] – one [dimension1])) +
square root of ((other [dimension2] – one[dimension2]) ×
((other [dimension2] – one[dimension2]))
return distance

The third distance measure experimented in this study is Haversine. According to Haversine [30], The length of two points on the surface of the earth is calculated based on latitude and longitude. To calculate Haversine, it requires information related to latitude and longitude (lat1, lng1, lat2, lng2). It is a method of knowing the distance between two points (customer and ATM location) [30]. Algorithm 3-5 depicts the way Haversine finds the distance between two points.

Algorithm 3-5:Haversine algorithm [21]

Input:
Point one
Point other
R, dLat, dLon, a,c,d

Output: distance

Method:
R=6371
dLat= other.dimension1-one.dimension1 × PI/180
dLon= other.dimension2-one.dimension2 × PI/180
a=sin(dLat/2) × sin(dLat/2) + sin(dLon/2) × sin(dLon/2)
b=invese_sin(square root of(a), square root of(1-a))
d=r*c
return d

3.5 Determining the shortest/optimal path

The third step in the architecture is finding or determining the optimal path. Route selection algorithms has been tested/checked to choose the best or optimal one. For this experiment OSM road intersection data of Addis Ababa city is used.

The data is prepared to make it suitable for the experiment software. More than 107,000 road intersection data of Addis Ababa city are prepared and is used by routing algorithms. QGIS, java program and excel are the main tools that is used to prepare the data.

The next task after the data preparation is to carry out experiment using three-road network route selection techniques such as A star, Dijkstra's and Breadth first search and select the best algorithms among them. This study uses these three algorithms because literatures vastly discussed them which tells that well-known in the area as well the researcher familiar to work with them.

Four key activities involved in this study while determining the optimal path from customer location to the ATM location:

1. Read initial node (user location) and temporary goal node (recommended active ATM) from **find nearest active ATM** component.
2. Read road intersection data
3. Read unique road intersection data which helps to find the nearest road node for the recommended ATM (final goal node).
4. Finding optimal path using A star, Dijkstra's, and Breadth-First search algorithms.

How each of the algorithms used for optimal route finding works is discussed in detail below.

A* search

To determine the shortest route using A star algorithm, it evaluates nodes using cost function f(n) that combines the past path-cost function, g(n) and a future path-cost function, h(n), f(n)=g(n)+h(n) [32].

Algorithm 3-6:A-star algorithm

Input:

start
goal

Output: optimal path

Method:

```
// The set of nodes already evaluated
  closedSet := {}
  // The set of currently discovered nodes that are not evaluated yet.
  // Initially, only the start node is known.
  openSet := {start}
  // For each node, which node it can most efficiently be reached from.
  // If a node can be reached from many nodes, cameFrom will eventually contain the
  // most efficient previous step.
  cameFrom := an empty map
  // For each node, the cost of getting from the start node to that node.
  gScore := map with default value of Infinity
  // The cost of going from start to start is zero.
  gScore[start] := 0
  // For each node, the total cost of getting from the start node to the goal
  // by passing by that node. That value is partly known, partly heuristic.
  fScore := map with default value of Infinity
  // For the first node, that value is completely heuristic.
  fScore[start] := heuristic_cost_estimate(start, goal)
  while openSet is not empty
     current := the node in openSet having the lowest fScore[] value
     if current = goal
        return reconstruct_path(cameFrom, current)
     openSet.Remove(current)
     closedSet.Add(current)
     for each neighbor of current
        if neighbor in closedSet
           continue              // Ignore the neighbor which is already evaluated.
        // The distance from start to a neighbor
        tentative_gScore := gScore[current] + dist_between(current, neighbor)
        if neighbor not in openSet       // Discover a new node
           openSet.Add(neighbor)
        else if tentative_gScore >= gScore[neighbor]
           continue
        // This path is the best until now. Record it!
        cameFrom[neighbor] := current
        gScore[neighbor] := tentative_gScore
        fScore[neighbor] := gScore[neighbor] + heuristic_cost_estimate(neighbor, goal)
```

Dijkstra's algorithm

Dijkstra's is another algorithm which is used for route selection. "Dijkstra's algorithm solves the single-source shortest-paths problem on a weighted, directed graph G = (V, E) for the case in which all edge weights are nonnegative" [33]. "Dijkstra's algorithm maintains a set S of vertices whose final shortest-path weights from the source s have already been determined.

Algorithm 3-7:Dijkstra's algorithm

```
Input:
        Graph
        source
Output: optimal path

Method:
        create vertex set Q
        for each vertex v in Graph:
           dist[v] ← INFINITY
           prev[v] ← UNDEFINED
           add v to Q
        dist[source] ← 0
        while Q is not empty:
           u ← vertex in Q with min dist[u]
           remove u from Q
           for each neighbor v of u:         // only v that are still in Q
              alt ← dist[u] + length (u, v)
              if alt < dist[v]:
                 dist[v] ← alt
                 prev[v] ← u
           end for
     return dist[], prev[]
```

Breadth-first search

The other algorithm is Breadth-first search. Breadth-first search finds the edge of G to "discover" each vertex that is reachable from s given a graph G = (V, E) and a distinguished source vertex s. Breadth-first search computes the distance (smallest number of edges) from s to each reachable vertex. Breadth-first search also produces a "breadth-first tree" with root s that contains all reachable vertices. For any vertex v reachable from s, the simple path in the breadth-first tree from s to v corresponds to a "shortest path" from s to v in G, that is, a path containing the smallest number of edges. The algorithm works on both directed and undirected graphs [33].

Algorithm 3-8: BFS algorithm

Input:
G,
start_v
Output: optimal path

Method:

```
let Q be a queue
label start_v as discovered
Q.enqueue(start_v)
while Q is not empty do
   v := Q.dequeue()
   if v is the goal then
      return v
   for all edges from v to w in G.adjacentEdges(v) do
     if w is not labeled as discovered then
        label w as discovered
        w.parent := v
        Q.enqueue(w)
   end for
```

Chapter 4
Dataset Preparation

4.1 Overview

The raw format of real data is usually widely variable. Many values may be missing, inconsistent across different data sources, and erroneous. For the analyst, this leads to numerous challenges in using the data effectively [37]. Therefore, a data preparation phase is needed.

The data preparation phase is a multistage process that comprises several individual steps, some or all of which may be used in a given application. These steps are as follows:

1. Feature extraction and portability: The raw data is often in a form that is not suitable for processing. Examples include raw logs, documents, semi-structured data, and possibly other forms of heterogeneous data. In such cases, it may be desirable to derive meaningful features from the data. Generally, features with good semantic interpretability are more desirable because they simplify the ability of the analyst to understand intermediate results. Furthermore, they are usually better tied to the goals of the data mining application at hand. In some cases where the data is obtained from multiple sources, it needs to be integrated into a single database for processing. In addition, some algorithms may work only with a specific data type, whereas the data may contain heterogeneous types. In such cases, data type portability becomes important where attributes of one type are transformed to another. This results in a more homogeneous data set that can be processed by existing algorithms.
2. Data cleaning: In the data cleaning phase, missing, erroneous, and inconsistent entries are removed from the data. In addition, some missing entries may also be estimated by a process known as imputation.
3. Data reduction, selection, and transformation: In this phase, the size of the data is reduced through data subset selection, feature subset selection, or data transformation. The gains obtained in this phase are twofold. First, when the size of the data is reduced, the algorithms are generally more efficient. Second, if irrelevant features or irrelevant records are removed, the quality of the data mining process is improved. The first goal is achieved by generic sampling and dimensionality reduction techniques. To achieve the second goal, a highly

problem-specific approach must be used for feature selection. For example, a feature selection approach that works well for clustering may not work well for classification.

For the various tasks performed in this study, we collect data from different sources. This is followed by a preparation of section 4.2 domain area understanding, data understanding section 4.3 and data preparation in section 4.4.

4.2 Understanding of the problem domain

In this section ATM and OSM problem domain understanding is discussed in detail. Preliminary survey revealed that, Bank of Abyssinia is one of the private banks that provide ATM services throughout the country (refer appendix IV question no. 3).

Bank of Abyssinia was established on February 15, 1996 (90 years to the day after the first but defunct private bank was established in 1906 during Emperor Menelik II) in accordance with 1960 Ethiopian commercial code and the Licensing and Supervision of Banking Business Proclamation No. 84/1994. The bank has undergone several restructurings. Now a day the bank provides excellence domestic, international and special banking services to its esteemed and valuable customers. It also strives to serve all economic and services sectors via its ever-increasing branch networks throughout the country [46].

Since the focus of this study is on ATM service, digital banking services at Bank of Abyssinia given more emphasis in this discussion. In Bank of Abyssinia, digital banking is organized in chief level under direct supervision of the CEO of Bank of Abyssinia [18]. The main responsibility of this department is to make sure the digital services are operating as per the bank strategy. Digital banking organized in to four departments/directors i.e. Online Banking (Mobile and Internet Banking), Card Banking (ATM and POS), Agent Banking and Digital Banking Operations [18].

Each department composed of at least two managers whose responsibility is to supervise the day-to-day proper operation of their respective services such as ATM, POS, mobile banking, internet banking and agent banking [18].

Discussion with experts revealed that, Digital Banking Operation department is responsible to ensure that, card banking services are working as per the standard of the bank such as 24/7. The same is true for the other three departments. For example, with respect to cash withdrawal failure dispute, Digital Banking Operation department is responsible to investigate the customer claim and made corrective actions if the customers complaint is found correct (refer appendix IV question no. 7).

The first issue with this study is determining ATM status for suggesting nearby active ATM for customers. To this end, we raise the following questions what kind of data relevant for this study? What are the variables for active ATM indicator? How to collect the data and why we collect it?

In this study, ATM data that is useful for the experiment is identified from two sources i.e. internal and external sources. Internal source of data is a data stored automatically in the ATM system database whenever the ATM performs any activity. This data includes cash withdraw, balance checking, and fund transfer. On the other hand, external source of ATM data is a data that describes the ATM external behavior such as ATM deployment environment, exististance of generator, exististance ups, working status of ups. This type of data is collected by ATM maintenance engineers on a daily basis when they supervise the ATM daily.

The second issue is identifying nearby ATM to the customer location. To ensure this issue, we need geo-location data of the ATM. In Bank of Abyssinia, ATMs are deployed near to bank branches, commercial center/shopping mall, super markets, hotels throughout Ethiopia. For this study 121 ATMs deployed in Addis Ababa are considered. Accordingly, we collected geo-location data of ATMs from Bank of Abyssinia.

The other problem dealt in this study is identifying optimal road route from customer's location to ATM location. For this paper, we considered Open Street Map (OSM) data useful for road routing in Ethiopia. OSM is a collaborative project to create a free editable map of the world. The creation and growth of OSM has been motivated by restrictions on use or availability of map information across much of the world, and the advent of inexpensive portable satellite navigation devices. OSM has grown to over 2 million registered users, who can collect data using manual survey, GPS devices, aerial photography, and other free sources. The data from OSM is available for use by Facebook, Craigslist, OsmAnd, Geocaching, MapQuest Open, JMP statistical software, and Foursquare to replace Google Maps, and more unusual roles like replacing the default data included with GPS receivers [47].

As noted [47], though there are different OSM data formats, in this study we use Shape file format as it contains different information relevant for this study such as road, building, lake, rivers, sewerages, football fields etc.

4.3 Understanding of the data

NCR and Diebold ATM types are vastly used in Bank of Abyssinia. Those ATM are deployed in the outline city as well as in Addis Ababa city.

Currently Bank of Abyssinia uses a system called "Select" to manage ATM and POS service. This system records every activity of the ATM and POS. In consultation with the domain expert, the researcher identifies the relevant data for this study. These data are obtained from internal (Select system/database) and external (reports) sources. From the internal source, fifteen attributes are selected. The below table 4-1 summarizes the attributes considered for this study.

Table 4-1: selected attributes for prediction

No	Attribute	Data type	Description
1	Deployment name	String	The name of the ATM where it is deployed and is known in the database
2	Category	String	Indicates whether ATM deployed in hotel, branch, mall etc.
3	Expose to rain	String	Specify whether the ATM expose to moisture or rain
4	Year of service	Number	Indicates how long the ATM give service
5	On schedule preventive maintenance	String	Determines whether the ATM get preventive maintenance on time/schedule
6	Manufacturing standard	String	Determines whether the ATM is lobby or through the wall and its manufacturers.
7	Deployment standard	String	Ensures whether the ATM is deployed as per the manufacturing standard (outside or inside)
8	Number of third part application installed	Number	Shows how many third-party applications is installed in the ATM
9	Ups exist	String	Indicates weather the ATM ha ups or not
10	Ups work	String	Shows whether the ATM ups functional or not
11	Generator exist	String	Shows whether the ATM has generator or get power from generator or not
12	Generator start automatically	String	Specify whether the generator start automatically as power interruption happens or when power shortage occurred

13	Transaction volume	Number	Show the number of transactions executed.
14	Transaction amount	Number	Show the amount of the transaction executed
15	Transaction date	Date	Show the date of transaction.

The other data that this study use is ATM-geo location. ATM geo-location data for 121 ATMs located in Addis Ababa are collected from commercial Google Map using PC connected to fast internet connection. This data contains ATM physical location data (latitude and longitude) as well area name where the ATM is deployed. This data is useful in the experiment of determining the nearest ATM and in the prototype application. Selected attributes are presented in table 4-3.

Table 4-2: ATM geo-location data attributes

Attribute	Data type	Description
Latitude	String	Latitude of the deployed ATM
Longitude	String	Longitude of the deployed ATM
Deployment name	String	The name of the ATM and known in the database.
Purpose	String	Is the ATM type that is developed for user/customer preference (refer appendix IV user/customer preference section).

This study experiments shortest route-finding algorithms using real world road data. To carry out this experiment, the need for geo-spatial data is important, hence GIS expert is consulted and identified the type of geo-spatial data that fit for this experiment.

Based on the expert recommendation, vector data that contains points and lines fit for this experiment. Therefore, vector data of Addis Ababa city road data has been obtained from OSM since it is free to use [11].

There are various sources of data for shortest route selection algorithms experimentation in virtual map environment. Google Map and OpenStreetMap are the popular one among others to accumulate virtual map routing data. However, Google Map is a commercial IT firm that provide data with payment, whereas, OpenStreetMap is an open source data which any one can use it for free. Therefore, experiment data for this study is obtained from OSM as it is free to use [47].

Depending on the user purpose, one can download OSM data of the whole world or small portion of the world like city or football area. Planet, ZAPI and bbbike.org are the popular OSM data provider (mirror/server) in different file formats.

Downloading the whole world OSM data take long time to download as well as to process since the data is huge. Therefore, we extract and download small portion of the OSM data (Addis Ababa city) for our purpose from online (https://extract.bbbike.org). Figure 4-1 shows OSM road network data of Addis Ababa city which is used for this study.

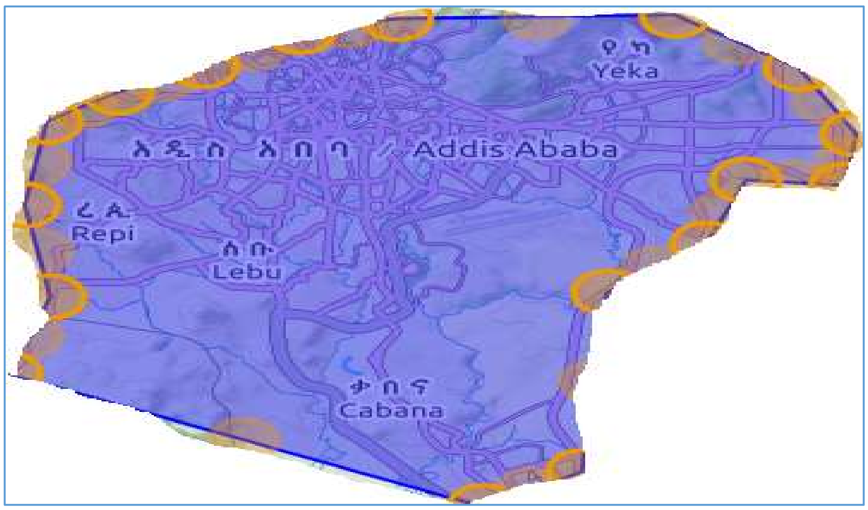

Figure 4-1: road route OSM data of Addis Ababa City

The downloaded shape file contains different data such as building, water, sea, road, football field etc. Since routing is one of the focus of this study, the important data for routing purpose is road data of Addis Ababa city. Selection of road route data is discussed in data preparation section 4.4.

As per the expert recommendation the relevant data for routing in Addis Ababa City road is identified and presented in table 4-5. Extraction of such data from OSM is discussed in the data preparation section 4.4.

Table 4-3: road routing data attributes

Attribute	Data type	Description
Latitude	String	Latitude is defined as an imaginary line joining points on Earth's surface that are all of equal distance north or south of the equator.
Longitude	String	Longitude is defined as the angular distance east or west of the prime meridian that stretches from the North Pole to the South Pole and

		passes through Greenwich, England. Longitude is measured in degrees, minutes, and seconds.
Name of the place	String	Name of the place is defined as name of the place or area recognized on the map by reverse geocode.
Road type	String	Road type is defined as Road type such as primary, resident, secondary, motorway etc.

4.4 Data preparation

This section is key as it helps to prepare data to make suitable for the experiment. Both ATM and OSM data preparation is discussed in this section.

ATM data pre-processing task which involves cleaning the data that is useful to predict whether the ATM is active (in service) or inactive (out of service). In this process data of ATM which is obtained from internal (database) and external (reports) are prepared.

Internal ATM data are those that are stored in the database automatically whenever any activity is done in the ATM. These include transaction amount, terminal, transaction volume, transaction date, transaction status and deployment date. This type of data in most cases is recorded whenever the ATM perform any activity.

External ATM data are those that describe the condition of the ATM (over all ATM status). This type of data is recorded and reported by ATM maintenance engineers on a daily basis during supervision. These include preventive maintenance schedule, deployment standard, generator existance, generator start automatically, ups exist, ups work are external data that is used in this study in consultation with domain expert.

The pre-processing steps for ATM data that is shown in figure 4-2.

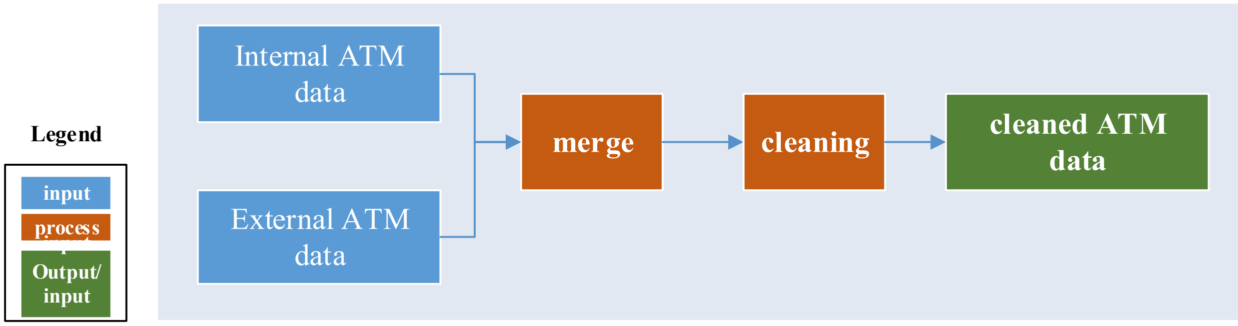

Figure 4-2: ATM data pre-processing steps

ATM data pre-processing activity has four steps. The first step performed is to capture internal ATM data from Bank of Abyssinia ATM system database. Three-month data from September 01, 2019 to November 31, 2019 are collected for 121 ATMs using SQL script and the total record is more than 45,000. Sample internal ATM data is presented in table 4-5.

Table 4-4:internal (database)ATM data (BOA, 2019)

TER_ID	MER_NAME	LOCATION	TRANS_AMT	TRANS_COUNT
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	6400	6
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	12200	12
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	33100	37
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	20900	22
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	23900	28
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	12100	13
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	7500	7
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	14400	12
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	19000	19
00300001	NEGADRAS BRANCH - WEST	NEGADRAS BRANCH ATM01 ADDIS ABAB ET	55100	33

The second step is to capture external ATM data from Bank of Abyssinia ATM field maintenance engineer report. Three months external ATM data from September 01,2019 to November 31, 2019 for 121 ATMs are collected. Table 4-6 presents sample external data.

Table 4-5: external ATM data (BOA, 2019)

deploymentname	category	exposetorain	yearofservice	onschedulepreventivemaintenance	Type	manufacturingstandard
SEALITE MIHIRET BRANCH ADDIS ABAB ET	branch	no	1	yes	TTW	Ncr through the wall
SENGA TERA BRANCH NCR ADDIS ABAB ET	branch	no	1	no	Lobby	Ncr through the wall
SHEWA ROBIT BRANCH NCR ADDIS ABAB ET	branch	no	1	yes	TTW	Ncr through the wall
SHIROMEDA BRANCH ATM01 ADDIS ABAB ET	branch	no	3	yes	TTW	Ncr through the wall
SIDIST KILO BRANCH ATM ADDIS ABAB ET	branch	no	4	yes	TTW	Ncr through the wall
SIGNAL BRANCH NCR ATM0 ADDIS ABAB ET	branch	no	1	yes	Lobby	Ncr through the wall

The third step is to merge internal and external ATM data using terminal id attribute which is common for both sources of data and is merged using java program (the program is presented in appendices III). The merged data is about 45,000 and shown in Table 4-7.

Table 4-6: merged ATM data (BOA, 2019)

terminalid	merchantid	deploymentname		txnvolume	txndate	creationdate	city	txnamount	txnstatus
12700002	000061012710707	3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	7.0	30-Nov-2019	10-Jan-2019	ADDIS ABABA	600.0	Do Not Honor
12700002	000061012710707	3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	1.0	30-Nov-2019	10-Jan-2019	ADDIS ABABA		Issuer Or Switch Inoperative
12700002	000061012710707	3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	23.0	30-Nov-2019	10-Jan-2019	ADDIS ABABA	22900.0	Approved or completed successfully.

In the fourth step, data cleaning activity carried out. In this step relevant data is identified and maintained as per the recommendation of domain expert and unnecessary data i.e. terminal id and merchant id has been removed from the corpus. Some of the activities of data cleaning are: for empty values found for some of transaction amount attribute. In consultation with domain expert, those values are replaced with zero. Special characters are identified and removed. To achieve the data cleaning task, Microsoft excel is used. Sample cleaned ATM data presented in table 4-8.

Table 4-7: cleaned ATM data (BOA, 2019)

deploymentname		category	txnvolum	txndate	service ye	txnamour	txnstatus	onschedu
3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	hotel	7	Saturday	0.916667	600	Do Not Hc	yes
3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	hotel	1	Saturday	0.916667	0	Issuer Or !	yes
3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	hotel	23	Saturday	0.916667	22900	Approved	yes
3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	hotel	3	Friday	0.916667	2000	Do Not Hc	yes
3DAYS HOTEL NCR - EAST ADDIS ABAB	ET	hotel	1	Friday	0.916667	500	Incorrect I	yes

For the ATM data, besides the above discussed data preparation, this study also uses clustering to further prepare the data. Since the data is large, it is time taking to label the class using experts while preparing training data, hence the need for clustering using data mining is crucial. A clustering method is applied to label the class into active and inactive for the training data. Using WEKA tool, the training data is classified into two clusters using expectation-maximization (EM). EM is selected because it is one of the most commonly used algorithms in the literature for similar purpose. Detail learning the pattern of the clusters is done in collaboration with the domain expert. Accordingly, they recommend cluster0 is inactive and cluster1 is active.

The other data pre-processing discussed in this section is OSM data. There are various software's that helps to pre-process OSM data. ArcGIS and QGIS are the popular one among others. ArcGIS is a commercial powerful software to pre-process geo-spatial data, whereas, QGIS is a free open source software anyone can use to pre-process geo-spatial data for free.

QGIS (previously known as Quantum GIS) is a free and open-source cross-platform desktop geographic information system (GIS) application that supports viewing, editing, and analysis of geospatial data [48]. Hence, this study uses QGIS to pre-process the OSM data.

QGIS functions as geographic information system (GIS) software, allowing users to analyze and edit spatial information, in addition to composing and exporting graphical maps. QGIS supports both raster and vector layers; vector data is stored as either point, line, or polygon features. Multiple formats of raster images are supported, and the software can georeference images [48].

QGIS supports shape files, coverages, personal geodatabases, dxf, MapInfo, PostGIS, and other formats. Web services, including Web Map Service and Web Feature Service, are also supported to allow use of data from external sources [48].

In connection to this, figure 4-7 below presents basic steps followed in data pre-processing and the discussion of it is presented next.

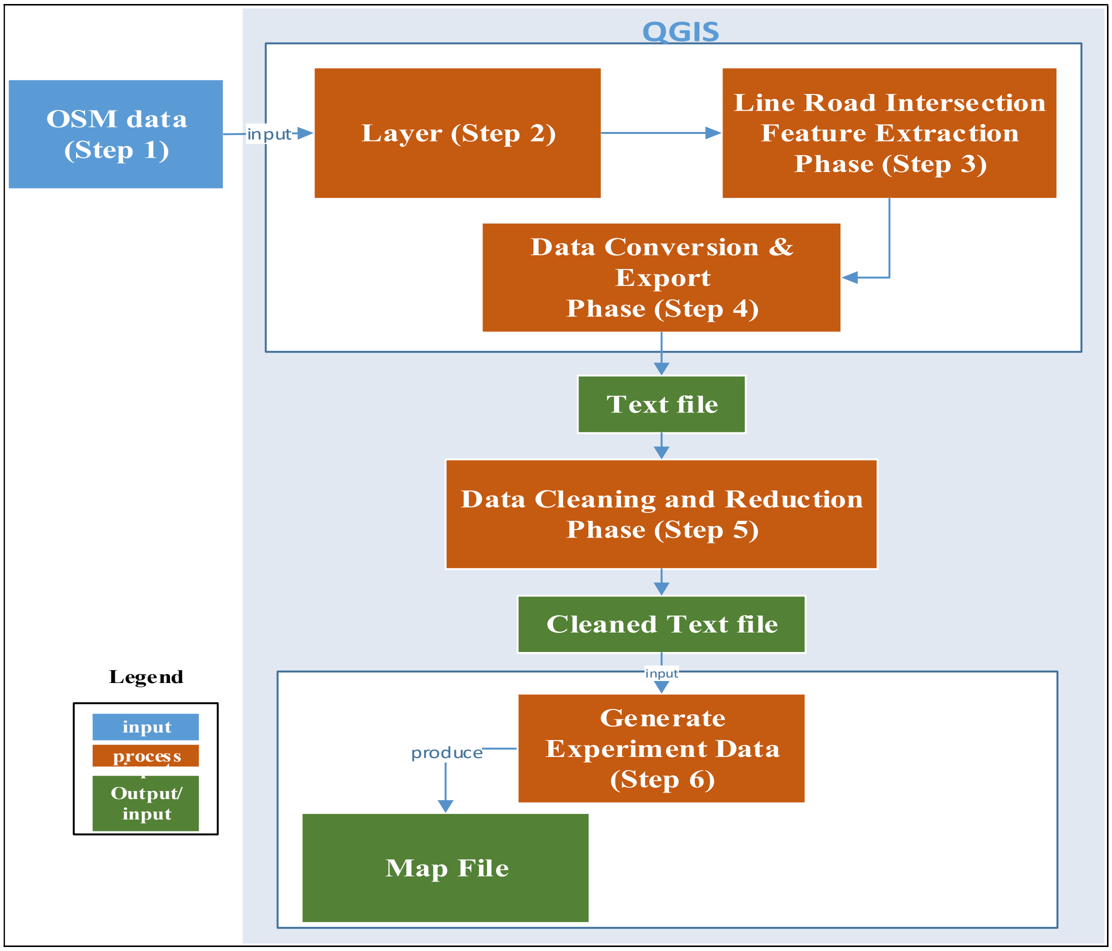

Figure 4-3: OSM data pre-processing steps

As shown in figure 4-7, OSM Data pre-processing has six important steps and all steps are elaborated in detail below.

Once, Addis Ababa city OSM shape file uploaded to QGIS in step1, the next is step2, the QGIS layer breakdown the shape file into various components (road, water, building, train road etc.). For the purpose of this work, we choose road data to further pre-process. What OSM shape file data looks like when viewed on QGIS layer shown in figure 4-8.

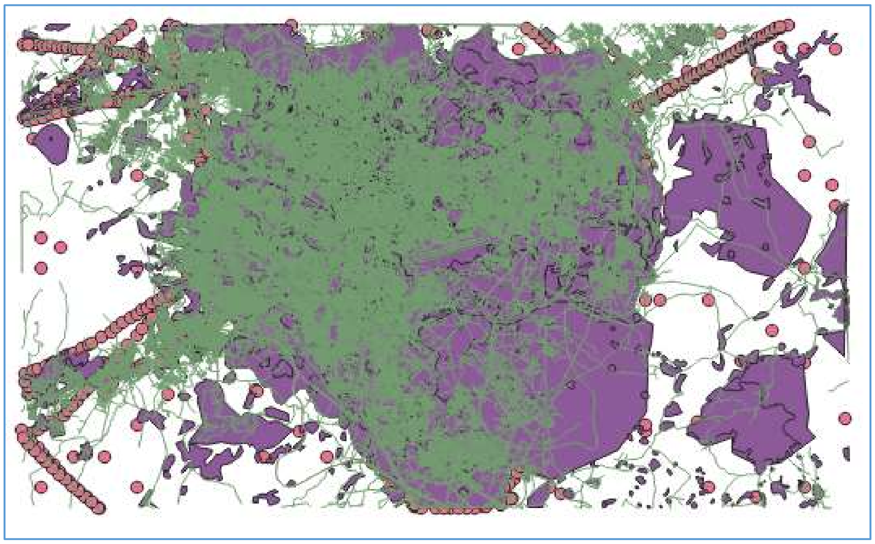

Figure 4-4: OSM data of Addis Ababa city when viewed in QGIS

In step3, line intersection feature extraction is done. This activity generates road intersection, polylines, lines and points. This activity generates road junction for Addis Ababa city using vector component of QGIS and presented in figure 4-9.

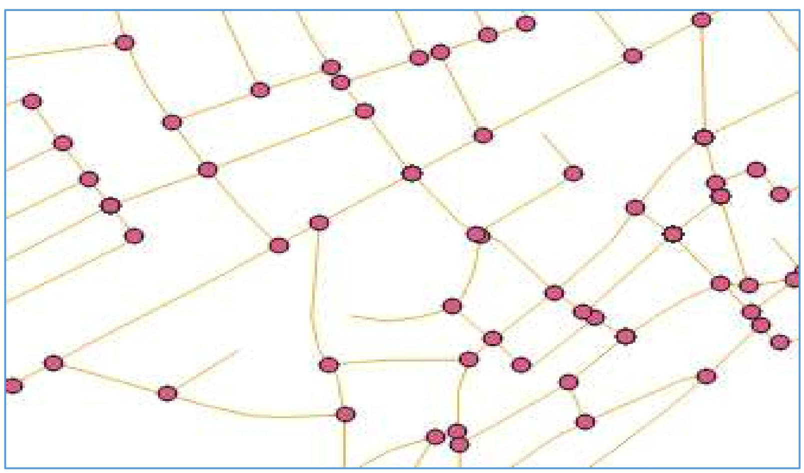

Figure 4-5: road line intersection

In step4, the road line intersection data generated in step3 is exported to text file format using vector line component of QGIS. The total road line intersection corpus data produced using QGIS is 107,963. Sample intersection data in text file format presented in figure 4-10.

```
//$DELIMITER "tab"
//$QUOTED-TEXT "no"
//$CHARSET ANSI
//$UNIT Distance:m
//$FORMAT 2
//$SYSCOORD {Type: 101}
//$FIELDS Class=intersection5100;Subclass=intersection5100;Kind=1;Fields=Private#Identifier Private#Class   Private#Subclass   Private#Name
Private#NbFields   osm_id  name   ref type   oneway  bridge  maxspeed   osm_id_2   Private#X   Private#Y
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   4677325 38.792880500   9.014703700
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   27880308   38.792554800   9.015543100
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   27880310   38.792700000   9.015219500
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   194467777   38.791220500   9.015205900
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   194467785   38.791687400   9.015322600
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   194467790   38.791567800   9.015289300
-1  intersection5100   intersection5100   intersection5100   8   4677317   residential 0   0   585074060   38.790660800   9.015059500
-1  intersection5100   intersection5100   intersection5100   8   4677323   residential 0   0   24291817   38.791699900   9.010753400
-1  intersection5100   intersection5100   intersection5100   8   4677323   residential 0   0   43217096   38.791949400   9.009773700
-1  intersection5100   intersection5100   intersection5100   8   4677323   residential 0   0   565808723   38.792081400   9.008622200
-1  intersection5100   intersection5100   intersection5100   8   4677323   residential 0   0   565829847   38.792081400   9.008622200
```

Figure 4-6: text intersection file exported from QGIS

The different types of road data for Addis Ababa city along with its total numbers generated from OSM shape file vector data are summarized in number in table 4-5.

Table 4-8:Addis Ababa city road type

Item	Frequency
residential	89,948
primary	6,579
Unclassified	5,382
Secondary	2,449
Tertiary	1,906
Service	1,378
Trunk	1,281
Primary link	931
Track	605
Path	559
Footway	462
Trunk link	186
Motorway	168
Secondary link	161
Motorway link	35
Steps	18
Construction	13
Total	107,963

In step5, the text file obtained in step4 converted into csv file format to make the data cleaning easier. Irrelevant fields and values removed using excel filter function. Only relevant fields i.e. primary, primary link, residential, secondary, secondary link, track, truck and truck link, construction, footway, motorway, motorway link, path, service, steps, tertiary and unclassified maintained. This step produces clean csv file for further pre-processing which is presented in table 4-10.

Table 4-9: CSV file after data reduction and cleaning

Private#Ic	Private#Cl	Private#Sı	Private#N	Private#N	osm_id	name	type	oneway	maxspeec	osm_id_2	Private#X	Private#Y
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79288	9.014704
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79255	9.015543
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.7927	9.01522
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79122	9.015206
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79169	9.015323
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79157	9.015289
-1	intersecti	intersecti	intersecti	8	4677317		residentia	0	1	4677317	38.79066	9.01506
-1	intersecti	intersecti	intersecti	8	4677323		residentia	0	1	4677323	38.7917	9.010753
-1	intersecti	intersecti	intersecti	8	4677323		residentia	0	1	4677323	38.79195	9.009774
-1	intersecti	intersecti	intersecti	8	4677323		residentia	0	1	4677323	38.79208	9.008622

After the data reduction step is done using excel software, we move on to data cleaning step and is done using java program.

In step6, the data from step5 is written into map file by reducing irrelevant attributes to satisfy the requirement of the experiment software. This task is performed using java program. For example, map file required to include latitude, longitude and name of the place attributes only. Table 4-11 shows sample map file.

Table 4-10: Sample map file format

```
Latitude     Longitude Latitude  Longitude   Name of the place   Road type
8.9567808 38.7637214 8.9556023 38.7636234 "Debre Zeit Road" primary
8.9556023 38.7636234 8.954252 38.7635111 "Debre Zeit Road" primary
8.954252 38.7635111 8.9538078 38.7634691 "Debre Zeit Road" primary
8.9538078 38.7634691 8.953329 38.7634047 "Debre Zeit Road" primary
8.953329 38.7634047 8.9532148 38.7633968 "Debre Zeit Road" primary
8.9726644 38.7607533 8.9727126 38.7609663 "" secondary
8.9727126 38.7609663 8.9727284 38.7610348 "" secondary
8.9727284 38.7610348 8.9727676 38.7614568 "" secondary
8.9727676 38.7614568 8.9727921 38.7618748 "" secondary
```

For the experiment one file having 54, 000 records have been prepared from the total 107,963 records, due to performance reason to work with the full data set. The reason not to experiment with the full data set i.e. 107,963 is due to experiment hardware resource limitation (8GB is the experiment machine but requires more than 8GM to run with the full dataset)

Chapter 5

Experimentation

5.1 Overview

In this chapter, we present the use of classification algorithms for building predictive model in section 5.2. Experiment nearest ATM and optimal path presented in section 5.3 and section 5.4 respectively.

ATM that exist in central, east and west Addis Ababa districts are selected using stratified random sampling is used. Using stratified sampling methods helps to select ATMs from each of the four districts of the bank in Addis Ababa. A total of 121 ATMs deployed in Addis Ababa city are considered for the experiment. This study collects 45,000 ATM records from Bank of Abyssinia database that covers from September 01,2019 to November 31,2019 for those ATMs whose geographic data considered in this study.

The procedures that is used while obtaining ATM data from Bank of Abyssinia ATM system are: first the relevant data is identified from the database. Second, using SQL script, the ATM data is retrieved and exported in excel format and maintained in the experiment PC for further processing.

The geo-location data for the stated number of ATMs are collected from Bank of Abyssinia using Google Map commercial since ATM deployment location data not maintained in digital format that contains latitude and longitude in the respective bank website. The data is collected from all Addis Ababa city districts (central, east and west).

5.2 Building predictive model

In this study, to predict the ATM working condition, two classification data mining algorithms such as J48 and PART are applied.

In this study, WEKA tool is used for data mining and knowledge discovery. The cross-validation technique was also used in this experiment. To examine the stability of the data mining models, 10-fold cross-validation was executed and the entire procedure was repeated 12 times.

To evaluate the performance of the data mining models, out of the total (45,343 record) corpus, the data were partitioned into a training set which accounts 70% (30,507) of the data set and a testing set which accounts the remaining 30% (14,836) of the data set.

The training set is used to construct a predictive model, whereas the testing set is used to evaluate the prediction accuracy of the model.

5.2.1 Clustering the data item

Before predicting the status of the ATM, the expert need to prepare training data whether the ATM is active or inactive. However, due to large corpus size, the expert unable to classify the training data manually and the need to cluster automatically is significant.

In this study EM algorithm is used for clustering. The clustering result using EM algorithm is shown in table 5-1 below.

Table 5-1: clustering result using EM

Cluster	Instances in number	Instances in %
Cluster 0(active)	20,964	72%
Cluster 1(inactive)	8,307	28%

The training data that is generated using WEKA by EM algorithm is saved in arff format to be used in the prediction of ATM status using classification algorithms which is discussed in detail in the next section.

In this study, WEKA tool is used for data mining learning. The cross-validation technique was also used in this experiment. To examine the stability of the data mining models, 10-fold cross-validation was executed and the entire procedure was repeated 12 times.

5.2.2 Experimenting classification algorithm

To construct a predictive model that determine ATM status, we use The J48 decision tree and PART rule induction classification algorithms. The experiment is made by interpolating parameter values such as un pruned and binary split parameters. Experiment result is summarized in table 5-2 below.

Table 5-2: experiment result

Experiments	Classification algorithm	Parameters		Accuracy
		Un Pruned	Binary Split	
1	J48	False	True	99.79 %
2	J48	True	False	99.73 %
3	**J48**	**True**	**True**	**99.81 %**
4	J48	False	False	99.73 %
5	PART	False	True	99.73 %
6	PART	True	False	99.72 %
7	PART	True	True	99.77 %
8	PART	False	False	99.73 %

As shown in table 5-2, J48 decision tree registered maximum of **99.81%** accuracy for unpruned true and binary split true, **99.81**. On the other hand, PART rule-induction register 99.77% for same parameter values. Hence, predictive model constructed using J48 to determine the status of the ATM. The confusion matrix is presented in appendix III.

```
=== Confusion Matrix ===

   a     b   <-- classified as
 4508    48 |    a = inactive
   79 39471 |    b = active
```

The above confusion matrix shown in table 5-3 that out of 44,106 instances 43,989 were classified correctly (99.81 %) which is the diagonal partition and two of them were classified incorrectly (0.19 %).

J48 decision tree rule learner with the specified scheme has resulted total of 56 rules. Out of these the rules which are highly predictive are selected and discussed as the finding of this study based on relevant to the domain. The following are selected best rules generated from the identified model.

Rule # 1: If txnvolume <= 34 and howmanythirdpartapplicationinstalled <= 3 and txnamount > 30950 and and txnamount <= 40800 and generatorstartautomatically = no and txnamount <= 37150 and txnvolume <= 29 and deploymentstandard = outside the hall and txnamount <= 35250, then the status of the ATM likely expected to be Active (60.0).

i.e.

If

Transaction volume is less than or equal to 34, **and**

Number of installed third party application is less than or equal to 3, **and**

Transaction amount is greater than 30,950, **and**

Transaction amount is less than or equal to 40,800, **and**

Generator does not start automatically, **and**

Transaction amount is less than or equal to 37,150, **and**

Transaction volume is less than or equal to 29, **and**

Deployment standard is outside the hall, **and**

Transaction amount is less than or equal to 32,250,

Then,

the status of the ATM is likely expected to be **Active** (60.0)

- The next rules can be described similarly.

Rule # 2: If txnvolume <= 34 and howmanythirdpartapplicationinstalled <= 3 and txnamount > 30950 and generatorstartautomatically = no and txnamount <= 37150 and txnvolume <= 29 and deploymentstandard = outside the hall and txnamount > 35250 and txnvolume <= 28, then the status of the ATM likely expected to be Active (10.0/1.0).

Rule # 3: If txnvolume <= 34 and howmanythirdpartapplicationinstalled <= 3 and txnamount > 30950 and txnamount <= 40800 and generatorstartautomatically = yes, then the status of the ATM likely expected to be Active (259.0).

Rule # 4: If txnvolume <= 34 and howmanythirdpartapplicationinstalled <= 3 and txnamount > 30950 and txnamount > 40800 and generatorstartautomatically = yes and txnamount <= 47100 and txnvolume <= 27, then the status of the ATM likely expected to be Active (21.0).

Rule # 5: If txnvolume <= 34 and howmanythirdpartapplicationinstalled <= 3 and txnamount > 30950 and txnamount > 40800 and generatorstartautomatically = yes and txnamount <= 47100 and txnvolume > 27 and txnamount <= 41900, then the status of the ATM likely expected to be Active (5.0).

Rule # 6: If txnvolume > 34 and txnamount <= 36150 and generatorstartautomatically = yes and txnvolume <= 44 and txnamount <= 32250, then the status of the ATM likely expected to be Active (190.0).

Rule # 7: If txnvolume > 34 and txnamount <= 36150 and generatorstartautomatically = yes and txnvolume <= 44 and txnamount > 32250 and txnvolume <= 40, then the status of the ATM likely expected to be Active (49.0/1.0).

Rule # 8: If txnvolume > 34 and txnamount <= 36150 and generatorstartautomatically = yes and txnvolume > 44 and txnamount <= 27850 and txnvolume <= 48, then the status of the ATM likely expected to be Active (17.0/3.0).

Rule # 9: If txnvolume > 34 and txnamount > 36150 and txnvolume <= 39 and generatorstartautomatically = yes and txnamount <= 38950, then the status of the ATM likely expected to be Active (22.0/3.0).

The model that is developed using J48 decision tree algorithm integrated with the nearest active ATM component of the prototype application by converting selected J48 rule into a logic that ordinary programming languages understood.

5.2.3 Evaluation and discussion

In this study, the condition of the ATM is predicted using data mining approach. 8 experiment had been done using classification i.e. J48 decision tree and PART algorithms using 43,343 data set. This prediction was used to examine the condition of the ATM that resides in Addis Ababa city.

The accuracy and consistency of the of the J48 were improved by applying data resampling to make the data balanced using WEKA. In addition, txn volume, no of third-party application, txn amount and generator start automatically are influential attributes which successfully predict the condition of the ATM. On the other hand, deployment name, ups exist, ups work, year of service, expose to rain, generator exist, manufacturing standard attributes has insignificant contribution on the ATM status prediction, hence removed from being considered in the prediction or construction the model.

The pruned and binary split of J48 model produce better performance, hence considered in the prototype-type application to present the discovered knowledge for the end user.

Additionally, WEKA have feature to visualize data mining metrics that is easy to understand. The other selected metrics that prove the experiment done as accuracy were precision, recall, and ROC. The following figures are screenshots taken from WEKA.

J48 decision tree is selected because it has better performance than PART algorithm. It is also easy to convert the rules, and their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. Therefore, the final model is developed using J48 decision tree to be integrated in the prototype system implementation.

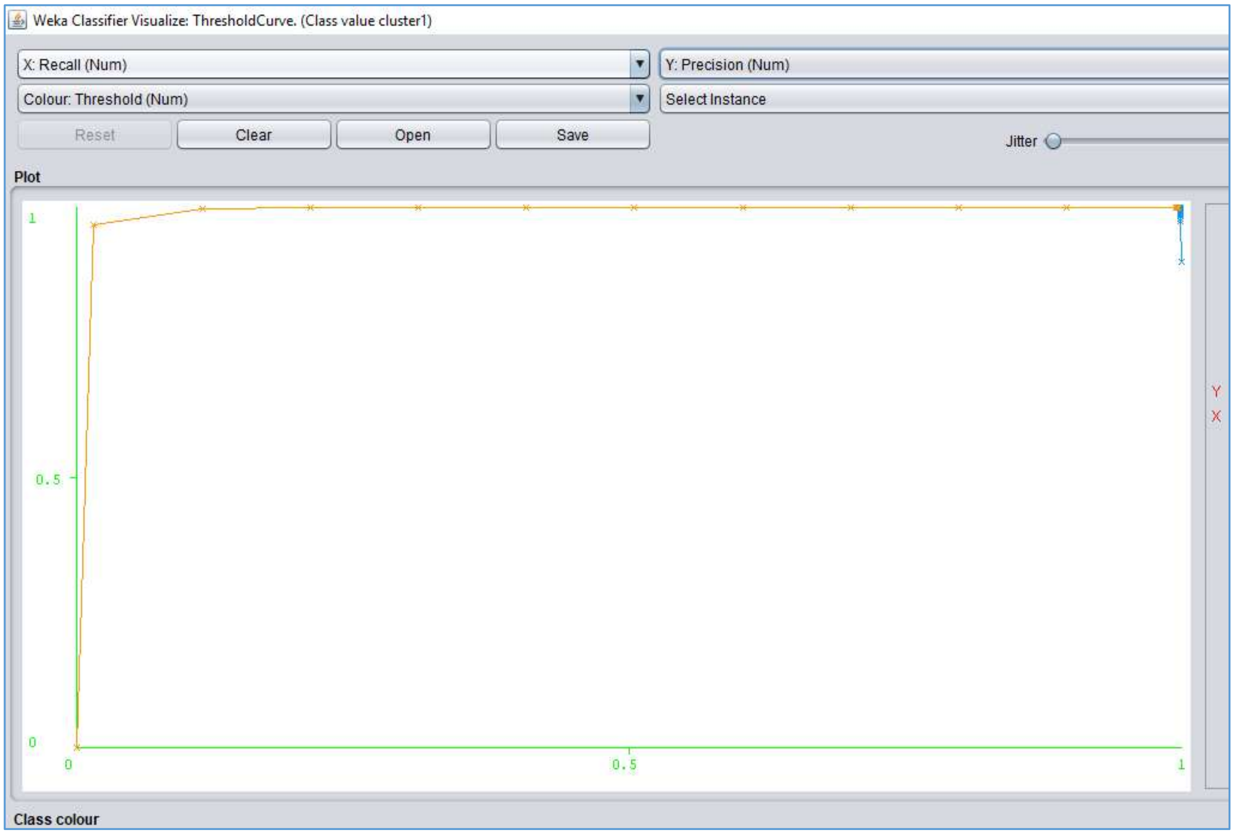

Figure 5-1: precision-recall curve of class active

Figure 5-1 precision-recall curves are important to visualize the classifier performances as accuracy. The aim is to observe whether precision-recall curve is towards the upper right corner of the chart. The P-R curve displayed with respect to class value active and x-axis is recall (true Positive Rate) and y a-axis precision.

5.3 Experimenting nearest ATM finder algorithms

Nearest ATM finder algorithms are useful to search the nearest active ATM in the user surrounding.

In this study, three straight line distance such as Euclidian, city-block and Haversine algorithms are experimented to select the optimal algorithm among them. 20 ATM-geo data is randomly selected and used in the experiment.

GoogleMaps software is used for the experiment and is selected because it is flexible to customize requirement and free to use since it is open source software. However, it only uses algorithms for identifying the nearest node. To test the selected algorithm, we compile the source code and integrate with GoogleMap for the experiment.

The start node for the three algorithms are set the same (9.0032545, 38.6953741). 20 ATM-geo data are randomly selected and employed in the experiment. To validate the algorithms performance or accuracy, a test data is prepared by collecting from commercial GoogleMap. Experiment result presented in table 5-4.

Table 5-3:nearest ATM finder algorithms experiment result

Start node	Goal node	**Euclidean distance**		**City-block distance**		**Haversine distance**		**Nearest target ATM**	**GoogleMap Commercial Test data**
		distance in km	Estimate time in ms	distance in km	time in ms	distance in km	time in ms		distance in km
8.9855316, 38.813478	8.9937685, 38.8112989	8.845	28	0.012	19	1.045	24	All the same	4.6
8.9639735, 38.8135406		**6.693**	**65**	**0.061**	**25**	**2.863**	**69**	**Euclidean and Haversine the same goal node(8.983819 ,38.796944), but city block differ (8.994591, 38.810951)**	• **6 for both,** • **5 for city**
8.935991, 38.8136378	**8.9579939, 38.7729788**	**0.002**	**20**	**0.061**	**23**	**5.112**	**25**	**Euc and Hav the same goal node(8.957640, 38.772594), but city differ (8.994591, 38.810951)**	• **31.4 for both,** • **12.4 for city**
8.9155557, 38.7158606	8.9489377, 38.7242697	0.001	19	0.041	25	3.806	20	All the same	6.9
9.0011571, 38.7346126	8.9917391, 38.7397776	9.173	26	9.173	23	1.062	10	All the same	6.9
9.0021177, 38.714262	8.988632,38.72339	2.699	30	2.699	35	1.820	7	All the same	1.9
8.9887289, 38.7125153	8.988632, 38.72339	1.239	23	1.239	29	1.223	36	All the same	2.3
8.964456, 38.7384339	8.9744539, 38.7443585	1.380	22	1.380	29	1.302	7	All the same	1.6

8.9425753, 38.768943	8.9576895, 38.7639959	2.378	31	2.378	20	1.713	3	All the same	2
8.9630125, 38.707904	8.9616333, 38.7138022	4.100	27	2.378	32	0.704	4	Euc and Hav the same, but city differ	1.2
8.9797933, 38.681965	8.9755517, 38.6817186	1.900	8	0.005	8	0.485	7	All the same	0.7
9.0028263, 38.7772268	8.999987,3 8.7809611	1.350	7	0.004	6	0.404	6	All the same	1.1
9.0079577, 38.7720414	8.9998955, 38.7699423	2.165	3	0.006	23	0.516	3	All the same	1.8
	Total	**41.925**	**309**	**19.437**	**297**	**41.925**	**221**		
	Average	**3.23**	**23.77**	**1.50**	**22.85**	**3.23**	**17**		

As presented in table 5-4, algorithms tested in this study gives the same target ATM 87% of the time. However, 13% of the cases City-block algorithm finds different and more nearest target ATM than the two algorithms, with big differences when checked with the test data of commercial Google map which is found in experiment 2&3 of table5-4. On the other hand, in all experiments, Euclidean and Haversine finds the same target ATM.

With respect to processing time Haversine register an average of **17 ms** which is the smallest compared to the other two which registered 22.85 for city-block and 23.77 **ms** Euclidean respectively. With respect to distance City-block register an average of **1.5 km** which is also the smallest compared to the other two which registered the same average result i.e. 3.23 km. Therefore, for this study City-block distance is selected for integration in the prototype application.

5.4 Experimenting optimal/shortest path selection algorithms

Optimal path recommends the optimal path from the customer location to the recommended active ATM. The need to experiment optimal path algorithms is to select the best algorithm and integrate to the prototype application.

GoogleMaps software is used for the experiment and is selected because it is flexible to customize requirement and free to use since it is open source software. However, it only uses algorithms for identifying the nearest node. To test the selected algorithm, we compile the source code and integrate with GoogleMap for the experiment.

In this experiment three algorithms i.e. Breadth First Search, A*(A star) and Dijkstra are tested. The algorithms are evaluated using time, distance and node visit.

Minimum distance is the length from customer location to ATM location and measured in terms of kilometer. Time implies how long it takes to process searching the shortest path from starting to destination location and measured in terms of millisecond. Node count is used to know the nodes selected to construct a path by counting the intersection point that the algorithm visits from the starting to the destination road node.

To carry out the experiment, one file having 54, 000 records are prepared from the total 107,963 records, due to performance issue to work with the full data set. We have not used the full corpus that contains all the data because of experimental hardware limitation.

Ten experiments had been conducted. The start and destination roads for the three algorithms are the same and taken randomly. For validation or testing purpose, the commercial Google map (GoogleMap) is used. The experimental result is shown in table 5-5.

Table 5-4: Experimentation and its result

Start road (lat, lon)	Destination road (lat, lon)	Distance in Kilometer			Processing Time in Millisecond			Nodes count			Distance in Kilometer (km) for Commercial GoogleMap
		A*	Dijk	BFS	A*	Dijk	BFS	A*	Dijk	BFS	
8.9842172, 38.6944694	8.9755517, 38.6817186	3.5	3.5	2.0	71	15	1509	45	45	25	2.3

8.9646147, 38.7034412	8.9616333, 38.7138022	1.5	1.5	1.1	9	6	696	1.5	26	23	2.7
9.0072133, 38.7358336	8.9999503, 38.7456549	1.7	1.7	1.3	7	6	103	18	18	15	2.9
8.9586337, 38.7927417	8.9579939, 38.7729788	4.2	4.2	2.1	212	16	4327	59	59	54	6.6
8.9185419, 38.7307886	8.9483348, 38.7363305	4.0	4.0	3.3	109	21	967	38	38	26	4.1
8.9597324, 38.716805	8.9489377, 38.7242697	1.9	1.9	1.4	32	7	1239	25	25	20	2.6
8.98545, 38.7996136	8.9950463, 38.7943467	1.6	1.6	1.2	13	9	633	22	22	22	1.7
9.0061795, 38.8137035	8.9937685, 38.8112989	1.6	1.6	1.4	8	7	306	31	31	31	1.7
9.0079059, 38.791364	8.999987, 38.7809611	2.0	2.0	1.4	40	5	841	41	41	29	2.1
9.0079857, 38.769385	8.9998955, 38.7699423	1.1	1.1	0.9	6	5	83	17	17	17	1.1
Total		**23.1**	**23.1**	**16.1**	**507**	**97**	**10704**	**297.5**	**322**	**262**	**27.8**
Average		**2.31**	**2.31**	**1.6**	**50.7**	**9.7**	**1070.4**	**29.75**	**32.2**	**26.2**	**2.78**

As shown in table 5-5, with respect to minimum distance, BFS register the smallest minimum average distance, **1.6 km** compared to the other two which register the same result i.e. 2.31 km.

With respect to processing time, Dijkstra's registered the smallest average processing time, **9.7 ms** compared to the other two and BFS registered the highest i.e. 1070.4 ms. Use of Dijkstra's improve resource performance.

With respect to node count, BFS register the smallest node count with an average of **26.2**, compared to the other two i.e. A* register 29.75 and Dijkstra register 32.2. Therefore, BFS has better result than the other two, hence it is selected for the integration of the prototype application.

Figure 5-2: road intersection points displayed on the Google map during experimentation

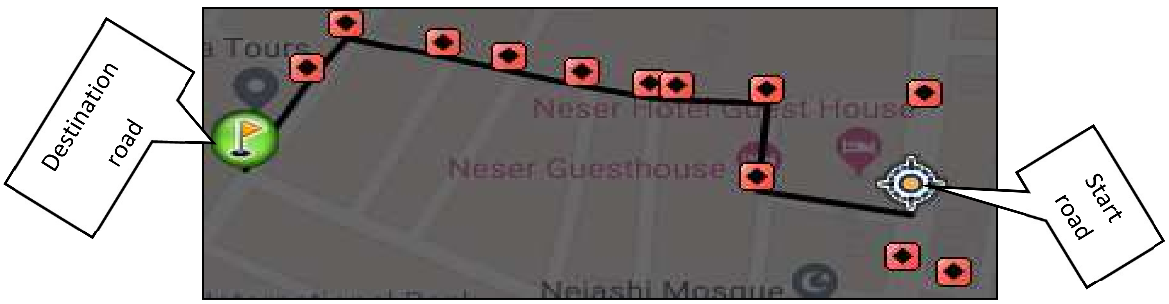

Figure 5-3: when the algorithm find & draw shortest path on Google Map

The diamond symbol in the Google Map shows road intersection point/nodes visited by the algorithms during shortest route searching

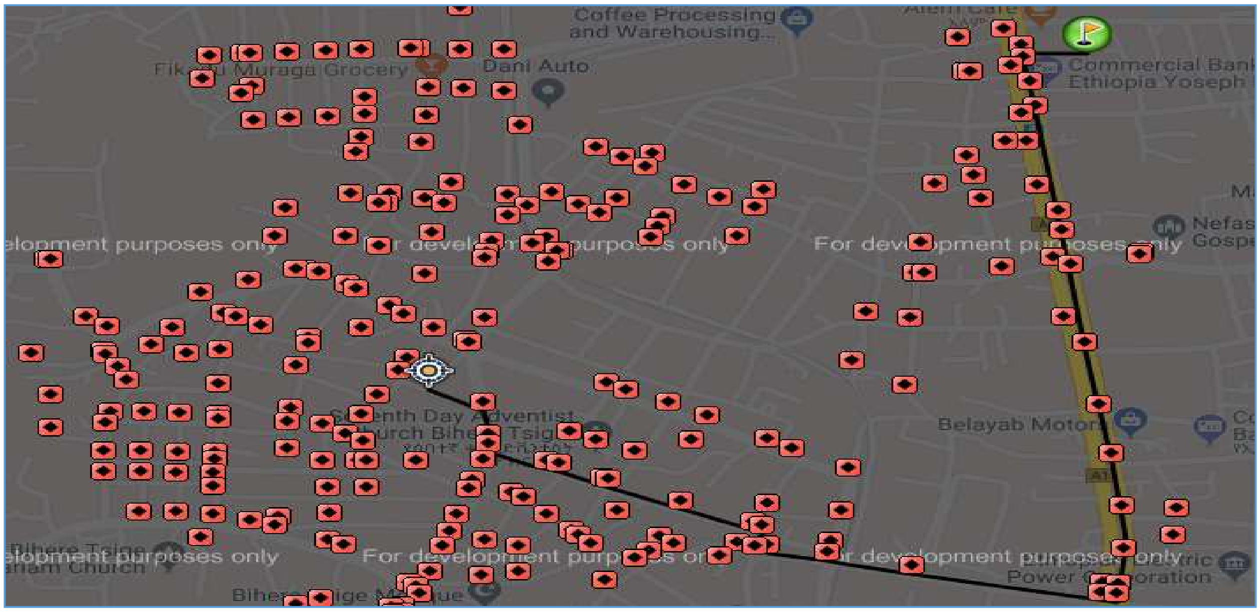

Figure 5-4: When the algorithm visits nodes/intersection points on google map

Chapter 6

Prototyping and evaluation

6.1 Overview

Integrating the model with the prototype application helps to utilize the knowledge generated by the model which increase the customer satisfaction and efficiency. For integration a java program is used and the below activity attempted.

6.2 Test data preparation

Data preparation is attempted to make suitable the data for the prototype application. The data is ATM data which is collected with the aim to test the prototype. Therefore, it needs some cleaning to make it fit for the prototype which makes only relevant attributes are maintained. The below table 6-1 presents sample ATM data for the prototype.

Table 6-1:data preparation for the prototype application (BOA, 2019)

deploymentname	category	txnvolume	txndate	txnamount	onschedulepreventivemaintenance	deploymentstandard	howmanythird	generator:
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	7	Saturday	600	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	1	Saturday	0	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	23	Saturday	22900	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	3	Friday	2000	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	1	Friday	500	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	1	Friday	2000	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	17	Friday	17700	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	1	Friday	3000	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	2	Thursday	4000	yes	outside the hall	3	no
3DAYS HOTEL NCR - EAST ADDIS ABAB ET	hotel	21	Thursday	28600	yes	outside the hall	3	no

The other activity attempted is rule conversion. The model that is developed by J48 classification algorithm is integrated with the prototype application. To do so, the rules generated by J48 algorithm is converted into a logic that ordinary programming languages understood. Only those rules which are recommended by the domain expert are selected and converted. Java is used to convert the rule developed by J48 rules. Snippet of converted code is presented in figure 6-1.

```
    public List<PredictActiveAtm> rules() throws IOException {
        List<PredictActiveAtm> lstActiveAtm = new ArrayList<>();
        for (PredictActiveAtm a : readersAll1()) {
            if ((a.getGeneratorstartautomatically1().equals("no"))
                    && (a.getTxnamount1() > 34050)
                    && (a.getTxnamount1() <= 36050)) {
                a.setFinal_status1("active");
                lstActiveAtm.add(a);
            }
            else if (a.getTxnamount1() > 30900 && a.getTxnamount1() <= 37150
                    && a.getGeneratorstartautomatically1().equals("no")
                    && a.getTxnamount1() > 34050
                    && a.getTxnamount1() > 36050) {
                a.setFinal_status1("active");
                lstActiveAtm.add(a);
            } else if (a.getTxnamount1() > 30900 && a.getTxnamount1() <= 37150
                    && a.getGeneratorstartautomatically1().equals("yes")) {

                a.setFinal_status1("active");
                lstActiveAtm.add(a);

            } else if (a.getTxnamount1() > 37150
```

Figure 6-1: rule converted from J48 model

The other task attempted while integrating the model with the application is merging ATM-geo data or ATM physical location data with active ATM data. Snippet of code is presented in figure 6-2 that shows merging active ATM with ATM-GEO data.

```java
public List<GmapNewDb> merge() throws SQLException, ClassNotFoundException, IOException {
    GmapNewDb hh = new GmapNewDb();
    PredictActiveAtm pp = new PredictActiveAtm();
    List<PredictActiveAtm> lstActiveAtm = new ArrayList<>();
    List<PredictActiveAtm> lstActiveAtm1 = new ArrayList<>();
    List<GmapNewDb> ll = hh.readAllAtmLocation();
    List<GmapNewDb> ll2 = new ArrayList<>();
    List<GmapNewDb> ll10 = new ArrayList<>();
    lstActiveAtm = pp.rules();
    lstActiveAtm1 = formatActiveAtm(lstActiveAtm);
    ll2 = formatAtmGeoData(ll);
    System.out.println("db connected successfully");
    for (PredictActiveAtm j : lstActiveAtm1) {
        for (GmapNewDb i : ll) {
            //this is where active ATM and ATM-GEO data merged
            if (i.getDeploymentNameNew().equals(j.getDeploymentNameNew())) {
                if (!ll10.contains(i)) {
//                      i.setCategory(i.getCategory());
                    i.setAtmStatus(j.getFinal_status1());
                    ll10.add(i);
                    System.out.println("unique dep name: " + i.getDeploymentName());
                }
            }
        }
    }
    System.out.println("total merged active ATMs after removing duplicated records: " + ll10.size());
```

Figure 6-2: merge active ATM with ATM-GEO data

The next activity attempted is to look for the nearest active ATM (temporary goal node or destination node) given start node. To do so first start node is inputted and then read merged ATM data that contains only active along the ATM geo-data. Since City-block is recommended as the optimal algorithm these data is supplied to it and the algorithm start to compute and finally return one ATM whose distance is small. Snippet of code is presented in figure 6-3 to show how city-block algorithms find nearest active ATM.

```java
public void setDestination(String yy, String rad) throws SQLException, ClassNotFoundException, IOException {
    if (pointLabel.getItem() != null) {
        GeographicPoint point1 = new GeographicPoint(9.130213, 38.7978143);
        MapApp mk = new MapApp();
        String selectedCategory = mk.getCbl().getValue();
        System.out.println("hi prference new !: " + selectedCategory);
        String lo = mk.getCbRaduis().getValue();
        System.out.println("hi raduis new !: " + lo);
        CLabel<geography.GeographicPoint> pointLabel1 = new CLabel<>("", null);
//          ObservableObjectValue<TestPerson> expected = new SimpleObjectProperty<>(person);
        pointLabel1.setItem(point1);//=point1;
        GeographicPoint point2 = new GeographicPoint();
        GeographicPoint point3 = pointLabel.getItem();
        GmapNewController u = new GmapNewController();
        lstGmapNew = u.getLstGmapNew();
        long startTime = System.currentTimeMillis();
        long end = 0;
        count_function(10000000);
        for (GmapNewDb g : lstGmapNew) {
            double h1 = point2.cityBlockDistance(point3.getX(), point3.getY(), g.getLatitude(), g.getLongitude());
            hm.put(g.getId(), h1);
        }
        end = System.currentTimeMillis();
        System.out.println("Counting to 10000000 takes " + (end - startTime) + "ms");
```

Figure 6-3: city-block algorithm when finding nearest active ATM

The other activity tried is to integrate BFS algorithm with the nearest active ATM finder. The algorithm takes input i.e. initial node, goal node, road intersection data, and unique road data. The algorithm finds optima path from customer location (initial node) to the recommended active ATM or destination node (goal node).

The other activity tried in the integration is incorporating the user preference i.e. ATM type and radius in the recommender prototype. To do this, the below activities tried:

- In the user interface a combo box is prepared to hold the user preference which automatically fetch data from database.
- Read all the ATM type and radius from database and automatically inserted into the combo's box.
- Listener or event is prepared to trigger when the customer selects preference.
- Recommend as per the user preference by showing in the user interface.

User preference is data is collected based on the survey which we collected by consulting domain experts as there is no data prepared for this purpose. This user preference data is prepared based on ATM service with respect to currency support (forex/local), ATM deployment category (branch, hotel, mall etc.) and the distance that the user wishes to get ATM service (radius). The user preference data then assigned or labeled for each of the 121 ATM-geo data in the database for simplicity.

After the integration is completed, the next activity performed is to conduct prototype system test using domain expert. For testing 8 test data is prepared that contains customer location and customer preference. The prototype test data and its result are presented in table 6-2.

Table 6-2:result of prototype system tested by experts

Test no	Customer location	Customer preference	Customer preference-based recommendation	Related recommendation
1	8.996604, 38.7343846	• Commercial Center • greater than 500-meter radius	SAN BUILDING BISRATE Gebreal, GOLLAGUL TOWER & MARATHON MOTORS	SARBET BRANCH, ADOT CINEMA, AFRICA UNION BRANCH, & LAFTO MALL
2	8.9804381, 38.7217838	• Branch normal • less than 500-meter radius	No item with the given criteria	SAN BUILDING BISRATE Gebreal, , LAFTO MALL, MEKANISSA BRANCH, & ADOT CINEMA
3	8.9554151, 38.7943621	• Cinema • greater than 500-meter radius	AMBASSADOR CINEMA, ADOT CINEMA & CINEMA EMPIRE ATM	SARIS ADDIS SEFER BRANCH, AIRPORT TERMINAL, SARIS BRANCH ATM01, SARIS BRANCH ATM02 & SARIS BRANCH ATM
4	9.0010025, 38.7386283	• University • greater than 500-meter radius	SIDIST KILO BRANCH ATM ADDIS ABABA	AFRICA UNION BRANCH, SARBET BRANCH, GUENET BRANCH, ADOT CINEMA & BALCHA BRANCH
5	8.9864037, 38.7362145	• Hotel Forex • greater than 500-meter radius	AZEMAN HOTEL, HILTON HOTEL & INTERCONTINENTAL HOTEL	MEKANISSA BRANCH, SARBET BRANCH, ADOT CINEMA, LAFTO MALL & SAN BUILDING
6	8.9467752, 38.7430882	• Mall • less than 500-meter radius	No item with the given criteria	LAFTO BRANCH, GOFFA MEBRATHAYILE BRA, LEBU BRANCH, SARIS BRANCH & SARIS BRANCH
7	8.9235639, 38.6875793	• Hospital • less than 200-meter radius	No item with the given criteria	ALEMGENA BRANCH, LEBU BRANCH, JEMO BRANCH, FURI BRANCH NCR & KARAKORE BRANCH
8	8.9063004, 38.7824927	• Branch normal • greater than 500-meters	KALITY MENAHERIA, KALITY BRANCH ATM01 & AKAKI BRANCH	KAFDEM PLAZA ATM01, KARAKORE BRANCH ATM01

The test result in table 6-2 shows that the prototype correctly recommends as per the user preference whenever the criteria is met. For example, test no. 1,3,4,5 & 8 met the user preference, hence, the prototype recommends as per the user preference. It also recommends related active ATMs that are not considered in the user preference. On the other hand, it also correctly recommends when user preference is not met. For example, test no 2, 6, and 7 can't met the user preference, no user preference-based recommendation, but the prototype recommends related active ATMs that are not with the user preference criteria. Therefore, the prototype correctly recommends active ATMs. Figure 6-4 shown the prototype recommends the customer after integration.

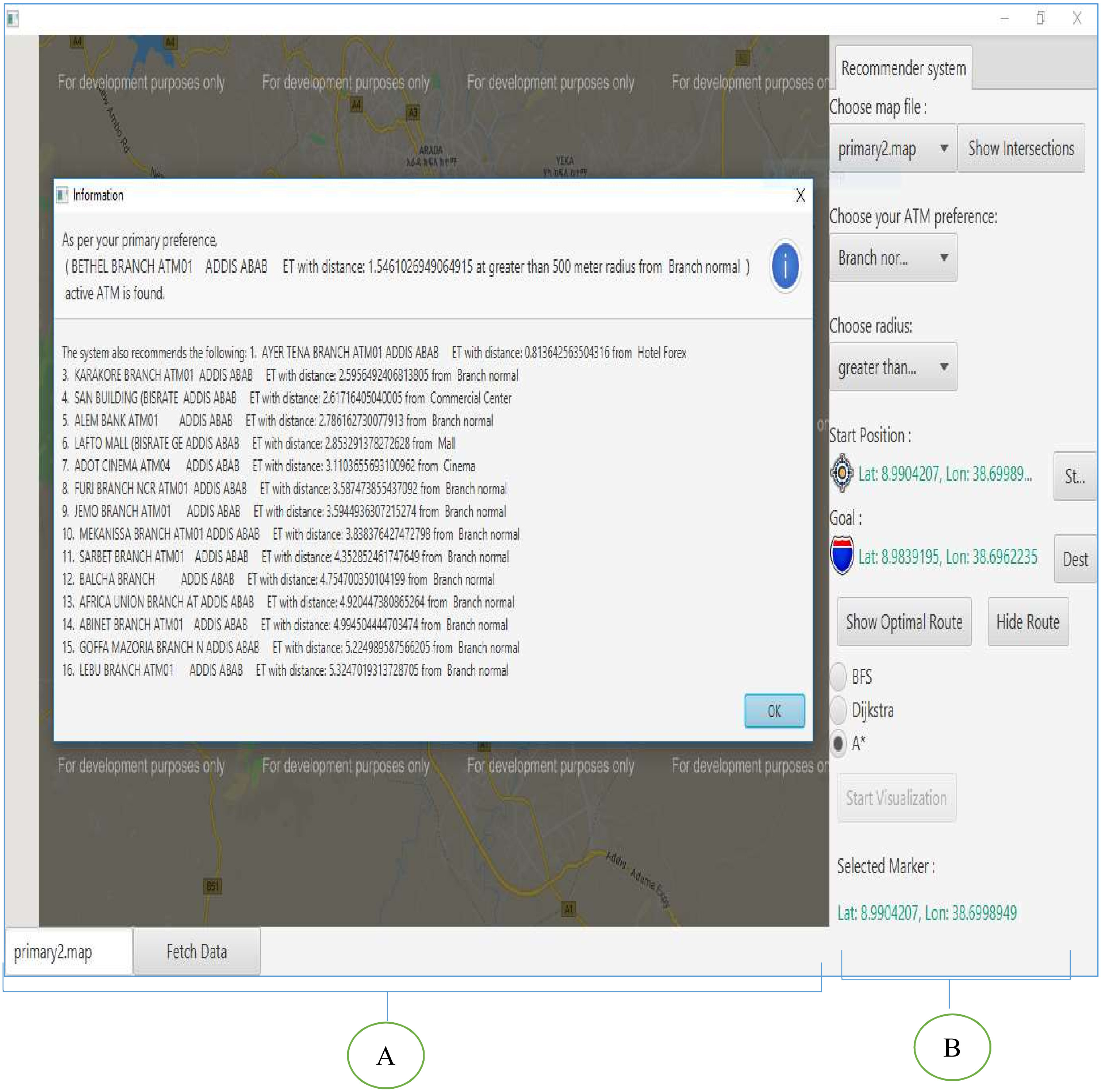

Figure 6-4: the prototype recommends primary preference & alternative active ATMs for customer

Figure 6-4 classified into two parts i.e. **A** and **B**. **A** shows the output area of the user interface whereas, **B** indicates input area of the user interface.

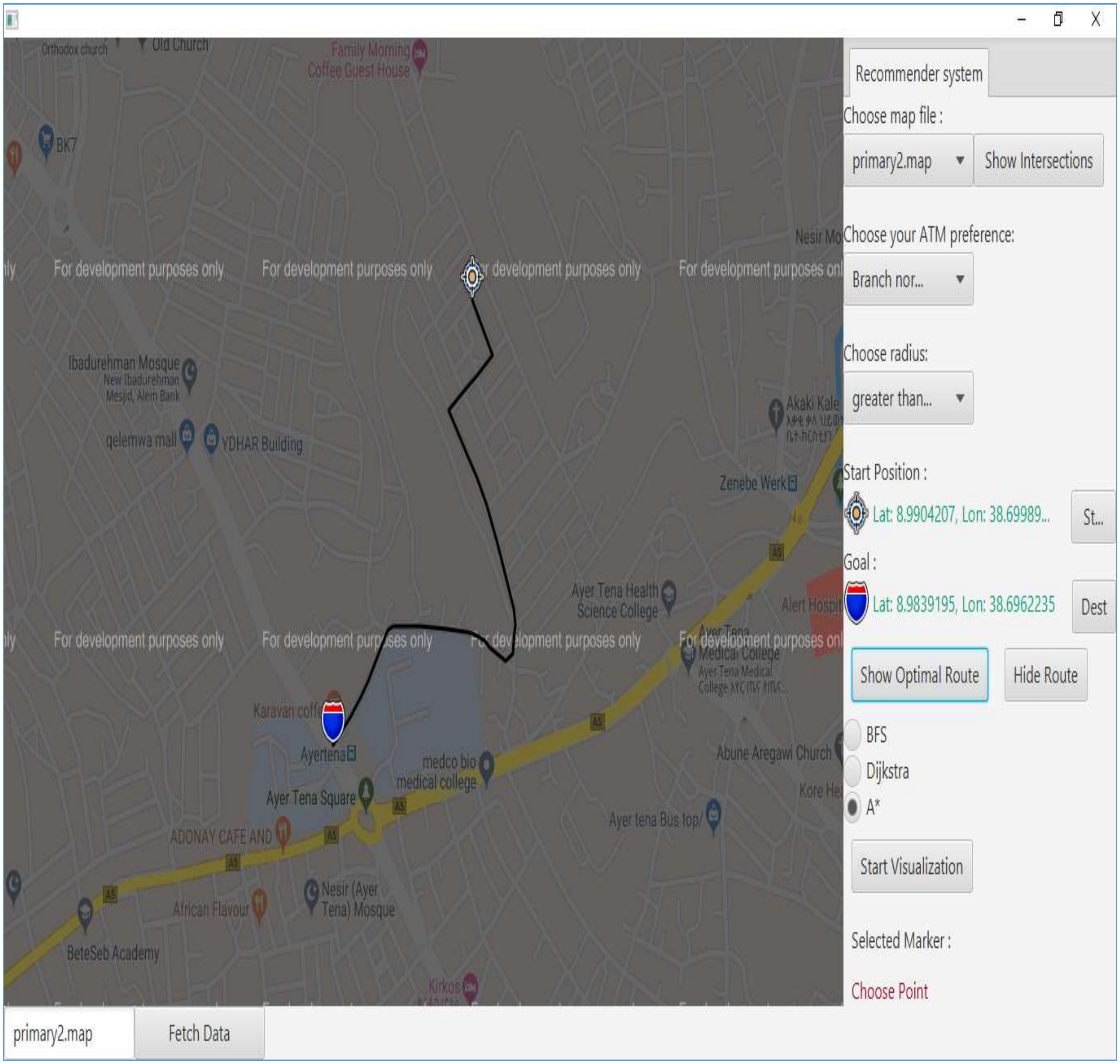

Figure 6-5:prototype recommend shortest path on GMAP

Figure 6-5 indicates the prototype recommend optimal path from customer location to the selected active ATM.

One of the interesting findings of this work is to predict active ATMs. The other finding is the system recommends as per the customer preference type as well as other related recommendation. The prototype:

- Predict active ATM
- Reject inactive ATMs from being considered in the recommendation.
- Find and recommend the nearest active ATM as per the customer preference.
- It also recommends related active ATMs in addition to the primary user preference.
- Find and recommend with radius.
- Find the shortest path from user to suggested active ATM.
- Shows route direction from start to destination road node on the map.

In conclusion the prototype performs good. The next sub-section discusses the evaluation of the prototype system.

6.3 Evaluation of the prototype system

Usability testing is done on the prototype application. 20 customers are involved in the testing. The evaluation is done from effectiveness, efficiency and user satisfaction perspective of ISO 9241-11 usability testing features [49].

A demonstration of the prototype has been given for the evaluators.

Guiding questionnaires with two parts has been prepared (see Appendices I). It is five level Likert scale (strongly agree (5), agree (4), neutral (3), disagree (2) and strongly disagree (1)) is used for the reply of the usability testing given questions.

There are different methods to analyze the questionnaire. The one this study used is average as it is easy to interpret the customer's feedback over the other similar data analysis methods. The result is presented below in table 6-3:

Table 6-3:Questionnaire result summary

Questionnaire						Result in average				
Question item	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1. Do you think the location aware active ATM recommender prototype essential to save cost for the customer?	7	5	0	0	0	0.54	0.46	0.00	0.00	0.00
2. Do you think the location aware active ATM recommender prototype essential to save time for the customer?	10	3	0	0	0	0.77	0.23	0.00	0.00	0.00
3. Do you think the location aware active ATM recommender prototype essential to save effort for the customer?	9	3	1	0	0	0.69	0.31	0.08	0.00	0.00
4. Do you think that the prototype can be used by any end user who have basic knowledge on smart phone/computer?	6	7	0	0	0	0.46	0.54	0.00	0.00	0.00
5. Do you think the response time for the operation fast enough?	3	7	3	0	0	0.23	0.54	0.23	0.00	0.00
6. Do you think the interaction to accomplish task is simple and complete with a few second?	7	5	0	1	0	0.54	0.38	0.00	0.08	0.00
7. Is location aware active ATM service recommender prototype easy to use?	7	6	0	0	0	0.54	0.46	0.00	0.00	0.00
8. Do you think the menu items are consistently located and work without failure?	4	6	3	0	0	0.31	0.46	0.23	0.00	0.00
9. Do you think the prototype is good enough product?	5	7	1	0	0	0.38	0.54	0.08	0.00	0.00
10. Do you think implementing this prototype in mobile based application more useful than the desktop version?	13	0	0	0	0	1.00	0.00	0.00	0.00	0.00

The guiding questionnaires in part two based on ISO 9241-11 usability testing features [49]. questions are classified into three. The questionnaires are prepared from effectiveness, efficiency and user satisfaction perspectives. The total number of questionnaires are ten. The result for the questionnaires are explained as follows:

Regarding cost, majority of the customers are satisfied on its cost saving. As a result, 54% of the respondents are strongly agree, whereas 38% of the respondent's agree. Regarding to time saving, the customers are satisfied that the prototype save time. As a result, 77% of the respondents strongly agree, whereas 23% of the respondents agree. With regard to effort, the customers are satisfied. Therefore, 69% of the of the respondents strongly agree, whereas 23% of the respondents agree and 8% of the respondent's neutral. With regard to usage, the prototype is 100% learnable by customers who have smart phone/computer know how. As a result, 46% of the respondents strongly agree, whereas 54% of the respondents agree. With regard to efficiency, majority of the customers are satisfied. Therefore, 23% of the respondents strongly agree, whereas 54% of the respondents agree and 23% of the respondents are neutral. Related to usability, the prototype is learnable by majority of the customers. Therefore, 92% of the respondents strongly agree, whereas the remaining 8% are unsatisfied. Related to failure, majority of the customers satisfied. As a result, 54% of the respondents strongly agree, whereas 38% of the respondents agree and 8% of the respondents are disagree. Related to the product quality, majority of the customers satisfied. Therefore, 54% of the respondents strongly agree, 53% of the respondents agree, the remaining 8% of the customers are average. The last question is related to improvements and 100 % of the customers would learn if it is upgraded to mobile application. Therefore, 100% of the respondents agree.

6.4 Discussion of result

To summarize, through this study, attempt has been made to answer research questions set on at the beginning.

Accordingly, the first research question was about the suitable attributes and algorithms for constructing ATM status prediction model. In responding to this, the study discusses the attributes and algorithms that fit for constructing a model to predict active ATM.

The second research question was the major consideration and requirements of active ATM recommender. Accordingly, the research discusses that there should be a mechanism to store LBS data, find nearest active ATM as per customer preferred radius, find the nearest active ATM as per the customer preference and also recommend related active ATMs, and find shortest path from the user location to the primarily recommended ATM and shows route on GMAP.

The third research question was about which prediction, straight line, optimal path algorithms work better. Accordingly, through successive experiment conducted, J48 decision tree, City-block and BFS algorithms are identified.

The fourth research question was to what extent location aware active ATM service recommender system works. Accordingly, it works in Addis Ababa city as data is collected from Addis Ababa and recommends as per the customer preferred ATM type and radius. It also works in other area as long as the data is provided.

Contribution of this study compared to other researches summarized and presented in table 6-4.

Table 6-4: contribution of this study

<table>
<tr><td>Other study</td><td></td><td>This study</td></tr>
<tr><td>Gugapriya*</td><td>• Problem: enhance existing banking services by moving toward mobile-banking
• Result: identifying Nearest ATM Centers by using GPS</td><td rowspan="4">• Problem: investigate challenges of card banking customer with respect to location aware nearest active ATM information gap.
• Approaches:
  • Predict active ATM: employ J48, PART algorithms; and select J48.
  • Determine nearest active ATM: experiment Euclidean, City-block and Haversine algorithms; and select Havesine.
  • Find optima path: experiment A star, Djikstra and BFS algorithms; and select A star.
• Result: Identify nearest active ATM, recommend nearest active ATMs as per the user preference (ATM type and radius), recommend other ATMs that also close to the customer without considering the user preference. show optima path from the user location to the recommended ATM.
• Recommendation: this study has limitation with respect to user need in how much effort</td></tr>
<tr><td>Mario*</td><td>• Problem: give android users the ability to find ATMs quickly.
• Result: enable finding of ATMs quickly, provide shortest path to selected ATM on foot or on car or on public transport options with tips where money can be withdrawn free of charge</td></tr>
<tr><td>Rajib*</td><td>• Problem: find the nearest ATM so as to address ATM location problem faced by the customer.
• Result: this work allows customers to see list of ATMs by bank, recommend the nearest ATM along with the shortest path on the map, show marker on the map along with textual information, and allow volunteer person to add new ATM location.
• Limitation: accuracy, usability, data duplication
• Recommendation: try with GoogleMap</td></tr>
<tr><td>Faried*</td><td>• Problem: construct a model for search application using mobile device
• Approach: augmented reality</td></tr>
</table>

	• **Result:** allows customer to search ATM location in the neighborhood, allow navigation and augmented reality, provide ATM availability condition information	the customer wants to get ATM service. Studying this issue can enhance the customer satisfaction.
Bharath *	• **Problem:** help customers by providing nearby ATMs information. • **Approaches:** hardware • **Result:** recommends the nearest ATM along with ATM availability with text, suggests alternative road when traffic congestion occurs, • **Limitation:** doesn't support local Wi-Fi internet connection, requires configuration and installation of utility software, requires user training	
Fasik*	• **Problem:** identify the reason for ATM failure • **Approaches:** J48 decision tree, PART rule determination, Naïve Bayes and Multi-layer perceptron, Neural Network algorithms • **Result:** dispenser, card reader, network device, software failure are major finding in this work.	

In conclusion, the contribution of this study compared to different researches i.e. [6], [16], [7], [6], [15] and [5] are:

I. With respect to approaches, this study upgrades the state of the arts by experimenting classification algorithms to construct better prediction model that determine active ATM, so as to identify the nearest active ATM for customers.

II. With respect to result, this study includes a recommender that recommends the nearest active ATM similar and/or related to user preference.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

Location based services (LBS) has the capability to provide information based on user's location.

Banking sector is one of the domain area that use location-based service to recommend the nearest ATM location for card banking customer and tourists who is new to the location/area and in need of cash for an immediate use in countries like Ethiopia where cash is the primary means for sales of goods and services. In Ethiopia for card banking customers and tourists, finding ATM is difficult. Even the discovered ATM might be inactive or out of service, this inhibits the customers to get the desired ATM service for an immediate use. This study is to predict active ATM for Ethiopian banking service and recommend for the customer based at their current location and based on their preference.

This study uses ATM data from Bank of Abyssinia to predict active ATM. For prediction, data mining technology is used. Using KDP, a model is developed to predict active ATM. To get the expected good result, 8 different experiments are conducted by measuring the accuracy using precision, recall, f-measure and ROC.

In this study, data understanding is done on the ATM data in consultation with the domain expert and then prepare the data set for data mining. 121 ATMs data from September 01, 2019 to November 31, 2019 is used for this experiment. Filling missing value and feature reduction is applied to clean the data.

In this study, two classification algorithms such as J48 and PART are used to experiment active ATM prediction using WEKA knowledge discovery tool. The result shows that J48 has better performance than PART, **99.81 %**. Therefore, the final active ATM prediction model is developed using J48 and is considered in the integration of the prototype application.

The attributes such as txnvolume, howmanythordpartyapplicationinstalled, txnamount, generatorstartautomatically and deploymentstandard have high significant in the prediction of the status of the ATM compared to the others.

The second experiment conducted in this study is straight line distance measurement algorithms using Euclidian, City-block distance and Haversine distance. Distance and processing time are used to evaluate algorithms. 15 different experiments are attempted to find the nearest active ATM by adjusting the start node for the three algorithms set the same. The result shows that Euclidian and Haversine algorithm find the same target whereas, but City-block sometimes find different target than the two. Based the result, City-block is selected for the integration in the prototype application.

In this study, the third experiment is carried out to find optimal path using Dijikstra, A*, and BFS algorithms. These algorithms use OSM data of Addis Ababa city road data. OSM data of Addis Ababa city is collected from online for road routing experiment. The data undergone different data preparation and more than 107,000 road intersection data is produced by the tool. But for the experiment 54,000 intersection data is employed. GoogleMaps is the tool selected to carry out the experiment. To evaluate those algorithms distance, processing time and node visited are used. The result shows that BFS register better result compared to the other two and therefore, BFS is selected as the optimal algorithm and considered in the integration of the prototype application.

The integration is successful. The result revealed that the application is able to recommend nearest active ATMs as per the customer preference ATM type as well as per selected radius. The system shows shortest route direction from start to destination road node on Google Map. The application drops those ATMs that are inactive during prediction from being considered in the recommendation.

In this study the following challenges are exhibited: One of the challenge in this study is unavailability of radius standard to determine the nearness of ATM the card banking customer. Though IP address-based location detection is attempted in this study the result obtained is more of a description of Addis Ababa city, rather than the specific location of customers.

7.2 Recommendation

This study explores active ATM prediction for location aware recommender system. By taking the findings of this study into consideration, with respect to future research directions, we recommend the extension of the work by considering the following:

In this study converting IP address to location is attempted to make the system recommends based on the PC IP address. But, the location obtained is not the exact location of the PC. Therefore, it is recommended to improve the conversion of IP to exact location to enhance usability of PC based location aware recommender system.

Due to lack of standard for radius, the radius used in the recommender prototype is not standard and it is recommended to define radius standard for the recommender to enhance the customer efficiency.

This study has limitation with respect to user need in how much effort the customer wants to get ATM service. It is recommended to study the customers need with how much effort (distance) they could find and use the ATM service would enhance customer satisfaction.

The other limitation is checking with mobile apps due to foreign currency payment requirement for android service. Formulating project, getting funding and testing using android is recommended to increase usability of the services.

Extending this work for blind customers to recommend and show route using voices in local language (Amharic and other local languages) is recommended.

One of the limitation of this study is doing optimal path without traffic data. It is therefore, recommended to experiment and see the effect by incorporating traffic data for Addis Ababa city road is recommended.

One of the challenge in this study is not making automatic knowledge discovery, hence, it is recommended to make the knowledge discovery automatic.

Reference

[1] V. Bhambri, "Application of Data Mining in Banking Sector," InternatIonal Journal of Computer Science and Technology, vol. 2, no. 2, p. 4, 2011.

[2] Gümüş, Sefer, Sudi Apak, Tuğbay Burçin Gümüş, Hande Gülnihal Gümüş, and Serpil Gümüş., "Ultimate Point in the Service Provided by the Banks to their Customers: Customer Satisfaction in the Common use of ATMs," Procedia - Social and Behavioral Sciences, 2015.

[3] Brian L. Dos Santos, Ken Peffers, "The Effect of Early Adoption of Information Technology: An Empirical Study," Journal of Information Technology Management, vol. IV, p. 14, 1993.

[4] Sujata Rao, Hrushikesh Mane, "ATM Availability Management System," IOSR Journal of Computer Engineering (IOSR-JCE), p. 11.

[5] Gugapriya, A., J. Vaitheki, and S. Kaviyarasi., "Mobile Banking With Location Tracking Of Nearest ATM Center Using GPS," International Journal of Innovative Technology and Research, vol. 1, no. 3, 2013.

[6] Rajib Chandra Das, Parijat Prashun Purohit, Tauhidul Alam and Mahfuzulhoq Chowdhury, "Location Based ATM Locator System Using OpenStreetMap," IEEE, 2014.

[7] I. Faried, "Application Model for Information ATM Location Using Location-Based Augmented Reality," International Journal of Progressive Sciences and Technologies (IJPSAT), vol. 10, no. 1, p. 4, 2018.

[8] Wen-Chen HU, Naima KAABOUCH, Hung-Jen YANG, "A Location-Based Service Using Geometric Location Methods to Unite Mobile Users," SYSTEMICS, CYBERNETICS AND INFORMATICS, vol. 14, p. 6, 2016.

[9] Swapnil R. Rajput, Mohd Sohel Deshmukh, Karbhari V. Kale, "Cross-platform Smartphone Emergency Reporting Application in Urban Areas using GIS Location based and Google Web Services," International Journal of Computer Applications (0975 – 8887), p. 7, 2015.

[10] D. O. A. N. Enayat Ali, "DEVELOPMENT MOBILE APPLICATION TO IMPROVE MEDICAL SERVICE (USING GOOGLE MAPS)," International Journal of Advanced Research in Computer Science, 2017.

[11] K.-T. Chang, Introduction to geographic information systems, Boston: Boston: McGraw-Hill Higher Education, 2006.

[12] Seema Vanjire, Unmesh Kanchan, Ganesh Shitole, Pradnyesh Patil, "Location Based Services on Smart Phone through the Android Application," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, no. 1, p. 6, 2014.

[13] "Annual Report," Bank of Abyssinia, Addis Ababa, 2019.

[14] B. Gezahegn, Assessment of customer Satisfaction with ATM banking: Empirical evidence from Selected Commercial Banks in Ethiopia, Addis Ababa Uiniversity, Faculty of Business and Economics, Department of Management, EMBA Program, 2015.

[15] F. Wondimu, Designing a model for identifying the reason for ATM out of service:, Addis Ababa: Addis Ababa University, 2019.

[16] Bharath, Vamsi, "ATM LOCATOR," International Journal of Students' Research In Technology & Management, p. 3, 2015.

[17] Steven M. Ross, Gary R. Morrison, EXPERIMENTAL RESEARCH METHODS, 2014.

[18] The Five Years Strategic Plan, Addis Ababa: Bank of Abyssinia, 2019.

[19] C. C. Aggarwal, Data Mining: The Textbook, Springer, 2015.

[20] R. Joshi, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures," 09 09 2016. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

[21] Pooja Singal, R.S.Chhillar, "Dijkstra Shortest Path Algorithm using Global Positioning System," International Journal of Computer Applications , vol. 101, p. 7, 2014.

[22] "A new time-dependent shortest path algorithm for multimodal transportation network," in The 8th international conference on ambient systems, networks and technologies (ANT 2017), Casablanca, 2017.

[23] A. Alem, APPLICATION OF CASE-BASED REASONING IN LEGAL CASE MANAGEMENT: AN EXPERIMENT WITH ETHIOPIAN LABOR LAW CASES, Addis Ababa: Addis Ababa University, 2014.

[24] Y. CHANIE, APPLICATION OF CASE BASED RECOMMENDER SYSTEM IN INVESTMENT SECTOR AND INVESTMENT ACTIVITY SELECTION TO NEW INVESTORS: IN THE CASE OF ETHIOPIA., Addis Ababa : Addis Ababa Universtiy, 2013.

[25] "ISO, Ergonomics of human-system interaction," ISO, Geneve, 2018.

[26] Gugapriya, A., J. Vaitheki, and S. Kaviyarasi., "Mobile Banking With Location Tracking Of Nearest ATM Center Using GPS," International Journal of Innovative Technology and Research, vol. 1, no. 3, p. 253–255, 2013.

[27] Nilima B.Umate, Prof . V.G. Bhujade, "Design of Targeted Advertising for Location Based Services and User Profile Based Recommendation," International Journal of Innovative Research in Computer and Communication Engineering, no. 9, 2017.

[28] D. Kaur, "A Comparative Study of Various Distance Measures," International Journal of Computer Trends and Technology , vol. 17, no. 3, 2014.

[29] "Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab," International Journal of Computer Science and Information Technologies, vol. Vol. 5 (2), 2014.

[30] Hartono, Subhan, Mhd Furqan, Andysah Putera Utama Siahaan, and Wirda Fitrian, "Haversine Method in Looking for the Nearest Masjid," International Journal of Recent Trends in Engineering and Research, 2017.

[31] Sunil Kumar Prabhakar, Harikumar Rajaguru, "Code Converters with City Block Distance Measures for Classifying Epilepsy from EEG Signals," ScienceDirect, 2016.

[32] S. J. R. a. PeterNorvig, Artificial Intelligence A ModernApproach Second Edition, Prentice Hall, 2003.

[33] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest,Clifford Stein, Introduction to algorithms, Library of Congress Cataloging-in-Publication Data, 2009.

[34] Jiawei Han,Micheline Kamber,Jian Pei, Data Mining Concepts and Techniques, 225 Wyman Street, Waltham, MA 02451, USA: Elsevier., 2012.

[35] M. H. Dunham, Data mining: Introductory and advanced topics, Pearson Education India, 2006.

[36] Madhavi, S., S. Abirami, C. Bharathi, B. Ekambaram, T. Krishna Sankar, A. Nattudurai, and N. Vijayarangan, "ATM Service Analysis Using Predictive Data Mining," International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 8, no. 2, 2014.

[37] C. C. Aggarwal, Data Mining: The Textbook, Heidelberg, New York, Dordrecht, London: Springer International Publishing Switzerland , 2015.

[38] Ng, Andrew Y., and Michael I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes., California, 2002..

[39] Bancu, Cristian, Monica Dagadita, Mihai Dascalu, Ciprian Dobre, Stefan Trausan-Matu, and Adina Magda Florea, "ARSYS--Article Recommender System," in IEEE, In 2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2012.

[40] CHEONG, Michelle L. F, "Ad-Hoc Automated Teller Machine Failure," Research Collection School Of Information Systems, Singapore Management University Institutional Knowledge at Singapore Management University, 2015.

[41] Rachburee, Nachirat, Samatachai Jantarat, and Wattana Punlumjeak., "Time series analysis for fail spare part prediction: case of ATM maintenance.," Journal of Telecommunication, Electronic and Computer Engineering (JTEC) , vol. 9, no. 2-6, pp. 49-52, 2017.

[42] Mario Lončar, Jasmin Abou Aldan, Marina Ivašić-Kos, "Mobile application for finding ATMs," MIPRO, 2015.

[43] Hindong Wu, Vipin Kumar, The top ten data minign algorithms in data mining, USA: Taylor & Francis Group, LLC, 2009.

[44] Purbaningtyas, R., and A. Arizal., "Nearest Excellent Potential Location Using Distance," 2019.

[45] "euclidean distance," revoledu, 2017. [Online]. Available: https://people.revoledu.com/kardi/tutorial/Similarity/EuclideanDistance.html. [Accessed 20 January 2020].

[46] "Bank of Abyssinia," Bank of Abyssinia, [Online]. Available: https://www.bankofabyssinia.com/. [Accessed 10 October 2019].

[47] Zhang, Liming, and Dieter Pfoser, "Using OpenStreetMap point-of-interest data to model urban change—A feasibility study," PloS one, vol. 2, p. 14, 2019.

[48] G. Maly, "Open Source Series Part 2: Spatial Analysis with QGIS (Article 1/2)," Digital Acceleration, February 2017. [Online]. Available: https://dai-global-digital.com/open-source-series-spatial-analysis-with-qgis.html. [Accessed 19 March 2019].

[49] I. STANDARD, ISO, Ergonomics of human-system interaction, Geneve: ISO, 2018.

Appendices

Appendices I: Usability Testing Questionnaire (End User)

The purpose of this system is to save cost, time and effort by providing location aware active ATM recommender system in real time for card banking customers. Hence, this questionnaire is prepared with the purpose to capture end user opinion on this system.

I. Background information

1. Gender:

Male	Female
☐	☐

2. Educational Qualification?

MSc.	BSc.	Diploma	Certificate	Other
☐	☐	☐	☐	☐

II. Prototype system:

The following questionnaire are application usability tests related to location aware active ATM service recommender system. Please indicate your agreement by marking "√" in the boxes.

No	Question	Strongly Agree	Agree	Neutral	Disagre	Strongl
1.	Do you think the location aware active ATM recommender prototype essential to save cost for the customer?					
2.	Do you think the location aware active ATM recommender prototype essential to save time for the customer?					
3.	Do you think the location aware active ATM recommender prototype essential to save effort for the customer?					

4.	Do you think that the prototype can be used by any end user who have basic knowledge on smart phone/computer?					
5.	Do you think the response time for the operation fast enough?					
6.	Do you think the interaction to accomplish task is simple and complete with a few second?					
7.	Is location aware mobile based active ATM service recommender prototype easy to use?					
8.	Do you think the menu items are consistently located and work without failure?					
9.	Do you think the prototype is good enough product?					
10.	Do you think implementing this system in mobile based application more useful than the desktop version?					

Please write any other comment about the location aware active ATM service recommender system?

Appendices II: J48 algorithm prediction result

=== J48 Run information ===

Instances: 44106

Attributes: 8

txnvolume

txndate

txnamount

onschedulepreventivemaintenance

deploymentstandard

howmanythirdpartapplicationinstalled

generatorstartautomatically

Cluster

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

```
------------------

txnvolume <= 34
|  howmanythirdpartapplicationinstalled <= 3
|  |  txnamount <= 30950: active (38755.0/4.0)
|  |  txnamount > 30950
|  |  |  txnamount <= 40800
|  |  |  |  generatorstartautomatically = no
|  |  |  |  |  txnamount <= 37150
|  |  |  |  |  |  txnvolume <= 29
|  |  |  |  |  |  |  deploymentstandard = outside the hall
|  |  |  |  |  |  |  |  txnamount <= 35250: active (60.0)
|  |  |  |  |  |  |  |  txnamount > 35250
|  |  |  |  |  |  |  |  |  txnvolume <= 28: active (10.0/1.0)
|  |  |  |  |  |  |  |  |  txnvolume > 28: inactive (3.0)
|  |  |  |  |  |  |  deploymentstandard = inside
|  |  |  |  |  |  |  |  txnvolume <= 18: active (4.0/1.0)
|  |  |  |  |  |  |  |  txnvolume > 18: inactive (4.0)
|  |  |  |  |  |  txnvolume > 29
```

```
|   |   |   |   |   |   |   txndate = Saturday: inactive (8.0/1.0)
|   |   |   |   |   |   |   txndate = Friday: inactive (6.0/1.0)
|   |   |   |   |   |   |   txndate = Thursday
|   |   |   |   |   |   |   |   txnamount <= 32600
|   |   |   |   |   |   |   |   |   txnvolume <= 32: active (6.0)
|   |   |   |   |   |   |   |   |   txnvolume > 32: inactive (2.0)
|   |   |   |   |   |   |   |   txnamount > 32600: inactive (4.0)
|   |   |   |   |   |   |   txndate = Wednesday: inactive (7.0/1.0)
|   |   |   |   |   |   |   txndate = Tuesday
|   |   |   |   |   |   |   |   txnvolume <= 30
|   |   |   |   |   |   |   |   |   txnamount <= 33550: active (2.0)
|   |   |   |   |   |   |   |   |   txnamount > 33550: inactive (2.0)
|   |   |   |   |   |   |   |   txnvolume > 30: inactive (10.0)
|   |   |   |   |   |   |   txndate = Monday: active (12.0)
|   |   |   |   |   |   |   txndate = Sunday
|   |   |   |   |   |   |   |   txnamount <= 31800: active (3.0/1.0)
|   |   |   |   |   |   |   |   txnamount > 31800: inactive (7.0)
|   |   |   |   |   txnamount > 37150
|   |   |   |   |   |   txnvolume <= 21
|   |   |   |   |   |   |   deploymentstandard = outside the hall: active (2.0)
|   |   |   |   |   |   |   deploymentstandard = inside: inactive (2.0)
|   |   |   |   |   |   txnvolume > 21: inactive (44.0)
|   |   |   |   generatorstartautomatically = yes: active (259.0)
|   |   |   txnamount > 40800
|   |   |   |   generatorstartautomatically = no: inactive (60.0)
|   |   |   |   generatorstartautomatically = yes
|   |   |   |   |   txnamount <= 47100
|   |   |   |   |   |   txnvolume <= 27: active (21.0)
|   |   |   |   |   |   txnvolume > 27
|   |   |   |   |   |   |   txnamount <= 41900: active (5.0)
|   |   |   |   |   |   |   txnamount > 41900: inactive (14.0/2.0)
|   |   |   |   |   txnamount > 47100: inactive (49.0)
|   howmanythirdpartapplicationinstalled > 3: inactive (718.0)
txnvolume > 34
|   txnamount <= 36150
```

```
|   |   generatorstartautomatically = no
|   |   |   txnamount <= 27400
|   |   |   |   txnvolume <= 40
|   |   |   |   |   howmanythirdpartapplicationinstalled <= 3
|   |   |   |   |   |   txndate = Saturday
|   |   |   |   |   |   |   txnvolume <= 38
|   |   |   |   |   |   |   |   txnamount <= 25650: active (12.0)
|   |   |   |   |   |   |   |   txnamount > 25650: inactive (2.0)
|   |   |   |   |   |   |   txnvolume > 38: inactive (6.0)
|   |   |   |   |   |   txndate = Friday: active (20.0/1.0)
|   |   |   |   |   |   txndate = Thursday: active (18.0/1.0)
|   |   |   |   |   |   txndate = Wednesday
|   |   |   |   |   |   |   txnvolume <= 36: active (7.0)
|   |   |   |   |   |   |   txnvolume > 36
|   |   |   |   |   |   |   |   txnamount <= 22250: active (5.0/1.0)
|   |   |   |   |   |   |   |   txnamount > 22250: inactive (10.0/1.0)
|   |   |   |   |   |   txndate = Tuesday
|   |   |   |   |   |   |   txnvolume <= 38: active (16.0/1.0)
|   |   |   |   |   |   |   txnvolume > 38
|   |   |   |   |   |   |   |   txnamount <= 22100: active (2.0)
|   |   |   |   |   |   |   |   txnamount > 22100: inactive (2.0)
|   |   |   |   |   |   txndate = Monday: active (22.0)
|   |   |   |   |   |   txndate = Sunday
|   |   |   |   |   |   |   txnamount <= 23900: active (12.0)
|   |   |   |   |   |   |   txnamount > 23900
|   |   |   |   |   |   |   |   txnvolume <= 36: active (2.0)
|   |   |   |   |   |   |   |   txnvolume > 36: inactive (7.0/1.0)
|   |   |   |   |   howmanythirdpartapplicationinstalled > 3: inactive (10.0)
|   |   |   |   txnvolume > 40: inactive (132.0/4.0)
|   |   |   txnamount > 27400: inactive (490.0/9.0)
|   |   generatorstartautomatically = yes
|   |   |   txnvolume <= 44
|   |   |   |   txnamount <= 32250: active (190.0)
|   |   |   |   txnamount > 32250
|   |   |   |   |   txnvolume <= 40: active (49.0/1.0)
```

```
|  |  |  |  |  txnvolume > 40: inactive (22.0/4.0)
|  |  |  txnvolume > 44
|  |  |  |  txnamount <= 27850
|  |  |  |  |  txnvolume <= 48: active (17.0/3.0)
|  |  |  |  |  txnvolume > 48: inactive (16.0/2.0)
|  |  |  |  txnamount > 27850: inactive (93.0/2.0)
|  txnamount > 36150
|  |  txnvolume <= 39
|  |  |  generatorstartautomatically = no: inactive (134.0)
|  |  |  generatorstartautomatically = yes
|  |  |  |  txnamount <= 38950: active (22.0/3.0)
|  |  |  |  txnamount > 38950: inactive (93.0/7.0)
|  |  txnvolume > 39: inactive (2616.0)
```

Number of Leaves : 56

Size of the tree : 101

Time taken to build model: 0.53 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	43979	99.7121 %
Incorrectly Classified Instances	127	0.2879 %
Kappa statistic	0.9845	
Mean absolute error	0.0038	
Root mean squared error	0.0498	
Relative absolute error	2.0345 %	
Root relative squared error	16.3545 %	
Total Number of Instances	44106	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.989	0.002	0.983	0.989	0.986	0.985	0.997	0.991	inactive
	0.998	0.011	0.999	0.998	0.998	0.985	0.997	0.999	active
Weighted Avg.	0.997	0.010	0.997	0.997	0.997	0.985	0.997	0.998	

```
=== Confusion Matrix ===

    a     b   <-- classified as
  4508    48 |     a = inactive
    79 39471 |     b = active
```

Appendices III: Implementations

Merge internal and external ATM data source code:

```
public List<ExperimentData> merger1() throws IOException, InvalidFormatException, SQLException,
ClassNotFoundException {
        ArrayList<ExperimentData> lstExp102 = new ArrayList<>();
        ArrayList<ExperimentData> lstExp103 = new ArrayList<>();
        lstExp102 = (ArrayList<ExperimentData>) prepareSalarySlip13();
        lstExp103 = (ArrayList<ExperimentData>) prepareSalarySlip14();
        int a = 0;
        ExperimentData bb = new ExperimentData();
        ExperimentData cc = new ExperimentData();
        for (ExperimentData v1 : lstExp103) {
            for (ExperimentData b : lstExp102) {
                if (v1.getTerminalid().equals(b.getTerminalid())) {
                    bb = new ExperimentData();
                    cc = new ExperimentData();
                    bb = b;
                    bb.setAddress1(v1.getAddress1());
                    bb.setExposetorain(v1.getExposetorain());
                    bb.setYearofservice(v1.getYearofservice());
                    bb.setManufacturer(v1.getManufacturer());
                    bb.setOnschedulepreventivemaintenance(v1.getOnschedulepreventivemaintenance());
                    bb.setManufacturingstandard(v1.getManufacturingstandard());
                    bb.setDeploymentstandard(v1.getDeploymentstandard());
                    bb.setHasfollowup(v1.getHasfollowup());
                    bb.setEmployeetraining(v1.getEmployeetraining());
                    bb.setProperswitchintegration(v1.getProperswitchintegration());
                    bb.setThirdpartapplicationinstalled(v1.getThirdpartapplicationinstalled());
                    bb.setHowmanythirdpartapplicationinstalled(v1.getHowmanythirdpartapplicationinstalled());
                    bb.setUpsexist(v1.getUpsexist());
```

```java
                bb.setUpswork(v1.getUpswork());
                bb.setGeneratorexist(v1.getGeneratorexist());
                bb.setGeneratorstartautomatically(v1.getGeneratorstartautomatically());
                bb.setServicehour(v1.getServicehour());
                lstSalarySlip101.add(bb);
                a++;
                System.out.println(bb.getDeploymentname() + " my amounts ss :" + bb.getTxnamount());
            }
        }
    }
    for (ExperimentData e : lstSalarySlip101) {
        System.out.println("deployment name:" + e.getDeploymentname() + " amount: " + e.getTxnamount());
    }
    return lstSalarySlip101;
}
```

Appendices IV: Preliminary survey questionnaire of ATM service

The purpose of this survey is to collect information about bank, ATM service and customer preferences. This survey has three parts. Part I of the questions are prepared to collect respondents background information. Part II of the questions are prepared to collect ATM service information. Part III are prepared to collect user preference information which is used in the recommender system.

I. Background information:

- Gender:

Male ☐	Female ☐

- Educational Qualification?

MSc.	BSc.	Diploma	Certificate	Other
☐	☐	☐	☐	☐

II. ATM service:

It consists of eight question with fill in the blank and yes/no type. Please indicate your agreement by underlining on the YES/NO type or filling on the blank space.

1. Is the services and products provided for customers significantly differ among local banks in Ethiopia? **<u>YES</u>/NO**
2. Is finding ATM convenient/easy for customer in unfamiliar location to get ATM service for urgent use? **YES/<u>NO</u>**
3. How many local banks currently operating in Ethiopia? **<u>19 local banks in Ethiopia and Bank of Abyssinia is one of them.</u>**
4. How many local banks in Ethiopia adopt ATM service? **18, except National Bank of Ethiopia**
5. How many ATMs are currently deployed by Bank of Abyssinia in Ethiopia? **<u>270</u>**
6. How many ATMs are currently deployed by Bank of Abyssinia in Addis Ababa city? **<u>150</u>**
7. How is the digital banking organized in Bank of Abyssinia?
   <u>It is organized in chief level who report to the CEO. Under the digital chief four directors are created to report to the chief of IT.</u>

8. How many district exist in Addis Ababa city? **3**

Central, west and east

Any comment related to the ATM service are appreciated to write in the space provided below?

____

____

____

III. Customer/User preference:

Data related to customer preference are collected to prepare ATM profile which is one of the customer preferences used in this study. ATM profile which we use for customer preference in the recommender system is prepared in consultation with the domain expert. Those preferences are presented below:

- Branch normal – this ATM support only local currency transactions which are deployed in branch.
- Branch forex – this ATM support foreign exchange in addition to local currency transactions and deployed in branch.
- Mall – this ATM support only local currency transactions which are deployed in mall.
- Cinema – this ATM support only local currency transactions which are deployed in cinema.
- University – this ATM support only local currency transactions which are deployed in cinema.
- Commercial center - this ATM support only local currency transactions which are deployed in business center i.e. super market.
- Hotel Forex – this ATM support foreign exchange in addition to local currency transactions and deployed in hotel.
- Hospital – this ATM support only local currency transactions which are deployed in hospital.

Radius is the second user preferences used in this study. Radius which we use for customer preference in the recommender system is prepared in consultation with the domain expert.

- Radius less than 200-meter radius – this ATM found with 200-meter radius.
- Radius greater than 200-meter and less than 500-meter radius – this ATM found between 200-500-meter radius.
- Radius greater than 500-meter radius – this ATM found with radius greater than 500-meter.