# Aggregating VMT Within Predefined Geographic Zones by Cellular Assignment: A Non GPS-Based Approach to Mileage-Based Road Use Charging

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Brian James Davis

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Max Donath

April 2012

# Acknowledgements

I would be remiss not to thank those who helped me complete this thesis. This project would not have been possible without the generous help of others.

I would first like to thank my adviser, Professor Max Donath, for his role in my graduate career at the University of Minnesota. His guidance and expertise were the key factors in the success of this project. He was a consistent ally, and I greatly value and am grateful for the relationship we developed throughout the course of my graduate career.

I would also like to thank everyone in the Intelligent Vehicle Lab for their time and assistance, especially Dr. Craig Shankwitz, whose experience and feedback contributed greatly to my project. Additionally I owe a great deal of gratitude to Alec Gorjestani and Arvind Menon, whose selfless assistance was invaluable.

I am thankful for the time and assistance I received from the Mechanical Engineering Department staff, especially John Gardner.

I would also like to thank the following groups for providing their support:

> This project received funding support from the Intelligent Transportation Systems (ITS) Institute and Hennepin County.
>
> The statistical Consulting Service at the University of Minnesota and in particular Patrick Zimmerman helped with the statistical analysis of the results of this project.
>
> Essential data collection was conducted through the use of City of Minneapolis Traffic Control vehicles and drivers. This was organized through the assistance of John Scharffbillig, Clara Schmit-Gonzalez, and William Gauthier.
>
> Additional data collection was conducted by vehicles from Linder Bus Company in Hutchinson, MN. This was organized through the assistance of John Brunkhorst (McLeod County Engineer), Brian Mohr, and Rick Linder.
>
> Hardware support was provided by Matt Sharma and Multi-Tech Systems, Inc.

Finally, I want to thank my friends and family, who made my journey through grad school easier.

# Abstract

Currently, most of the costs associated with operating and maintaining the roadway infrastructure are paid for by revenue collected from the motor fuel use tax. As fuel efficiency and the use of alternative fuel vehicles increases, alternatives to this funding method are being considered that don't use fuel consumption as a surrogate for road use. Many systems have been proposed which are capable of assessing mileage based user fees (MBUF) based on the vehicle miles traveled (VMT) aggregated within predetermined geographic areas, or travel zones, in which the VMT is generated. Most of the systems capable of this use GPS. However, GPS has issues with public perception, commonly associated with unwanted monitoring or tracking and thus an invasion of privacy.

One method to mitigate these issues is to use a system that utilizes a cellular network based approach that can determine a vehicle's current travel zone, but does not determine a vehicle's position through the use of GPS. The approach proposed here is based on a k-nearest neighbors (KNN) machine learning algorithm focused on the boundary of such travel zones. This method has two main phases. In phase one, the training phase, data is collected near zone boundaries using a cellular modem and a GPS receiver. This hardware creates a database that pairs readings consisting of observable cell towers and the strengths with which they were received, with the travel zone in which the reading took place, as determined by the GPS receiver. Then in phase two, the operational phase, GPS is no longer needed as the system detects changes in the vehicle's travel zone by

comparing currently available cellular information with the database. This method, while capable of determining the travel zone, is incapable of determining a vehicle's precise location, which better preserves both the user's actual privacy and perceived privacy.

The work described here focuses on the design and evaluation of algorithms and methods that when combined, would enable such a system. The primary experiment performed evaluates the accuracy of the KNN algorithm at sample boundaries in and around the commercial business district (CBD) of Minneapolis, Minnesota. The results show that with the training data available, the algorithm can correctly detect when a vehicle crosses a boundary to within ±2 city blocks, or roughly ±200 meters. A means for handling this relatively small ambiguous region between travel zones is also presented. The findings imply that a cellular-based VMT system may indeed be a feasible method to aggregate VMT by predetermined geographic travel zones.

## Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

## 1.1 Project Introduction and Document Organization

In this document we describe the design and evaluation of technology that enables a system capable of assessing road use fees based on both the distance a vehicle travels and the jurisdictions or zones in which the travel occurs. The methods presented here are an extension of the paper, *Technology Enabling Near-Term Nationwide Implementation of Distance Based Road User Fees* (Donath et al. 2009), which proposed an alternative to GPS for collecting road user fees. The rationale for examining such an alternative is the perception that GPS based approaches encroach on the traveler's privacy. The method described determines vehicle miles traveled by calculating distances by numerically integrating vehicle speed obtained from the vehicle's data bus, accessed through the OBD-II port. The jurisdictions or zones in which the travel occurs is determined by taking advantage of the cellular modem typically used to wireless communicate road use data to the back office to also identify the travel zone.

The focus of our effort is the design and evaluation of a cordon based algorithm, which was able to determine the jurisdiction or zone in which the vehicle was traveling. The research described here also extends the prior work by performing a preliminary evaluation of the ability of a simple in-vehicle device to calculate the distance the vehicle has traveled.

The rest of this chapter discusses the motivation for mileage based road use charging and a number of methods by which this can be accomplished. It also describes the method we

propose to accomplish this as well as further information about how such a method might be implemented in a fully functioning road user charging system.

Chapter 2 contains the description and results of a preliminary experiment of how accurately the hardware can calculate the vehicle miles traveled from the current speed as available from the vehicle's on-board computer and accessed through the OBD-II port.

Chapter 3 presents the machine learning algorithm used to perform the zone determination.

Chapter 4 details the methods by which the machine learning algorithm was evaluated.

Chapter 5 presents the results of this analysis.

Chapter 6 discusses these results and presents a possible method for further mitigating the ambiguity between zones.

### 1.2 Motivation for Mileage Based Road Use Charging

Currently, most of the costs associated with operating and maintaining the federal roadway infrastructure are paid for through the Highway Trust Fund (HTF), a mechanism to ensure continued funding of federal and state highways. The HTF receives funding from several sources including taxes on heavy truck use as well as sales of trucks, trailers, and tires. However, most of the revenue is from the federal motor fuel tax. While federal highways are funded primarily by the HTF, state, county, and city roads are

funded locally by state fuel taxes as well as from these various jurisdictions' general funds.

The state and federal motor fuel taxes, commonly referred to jointly as the gas tax, have been the primary funding source for our roadways for the last 50 plus years. It has shown itself to be relatively inexpensive to manage and operate and is understandable to the general public. However, in recent years, a number of problems with the gas tax have been identified.

The most pressing issue with the gas tax is that it is no longer sufficient to support the growing needs of the nation's roadway infrastructure. This is mainly due to the fact that road use is growing faster than gasoline consumption. This is a result of increasing fuel efficiency in vehicles as well as a growing number of hybrid and plug-in electric vehicles.

Additionally, there are also issues with the fairness of the use of the gas tax to raise revenue. Assessing a uniform, per gallon tax on gasoline makes an assumption that every gallon of gasoline affects the road infrastructure in the same way. Consider a hybrid-electric vehicle that uses much less gasoline than a traditional vehicle, or a plug-in electric vehicle that uses no gasoline at all. These vehicles contribute to roadway wear and congestion in the same way as a traditional vehicle, but in the process, pay less. Gasoline usage is ultimately an approximation for the number of miles a vehicle travels.

These issues motivate a solution that is capable of assessing a road use fee that is fair both to roadway users and the jurisdictions that maintain them. One way to accomplish

this is the use of mileage based user fees (MBUF). This approach is based on the idea that instead of charging for road use indirectly through gasoline usage, the best way to charge drivers for their road use is to determine the vehicle miles traveled (VMT) and then based on that, assess a per mile fee.

There are a number of potential ways to implement MBUF ranging in complexity from "pen-and-paper" methods, to using automated GPS based hardware. These systems address the issues with the gas tax in that they can, but don't necessarily have to, charge vehicles the same rate based on their VMT. Road use fees should facilitate road use pricing that is fair for both the road users and the jurisdictions that maintain the roads.

In addition to determining VMT, it is also advantageous to determine in which jurisdictions, or travel zones, these miles are traveled. Here, a travel zone can be defined as an entire state, a county, a city, or just a portion of a city. This would allow for road use fees to be directly allocated to the jurisdictions in which the road use occurs. This would allow for jurisdictions that experience heavier road use to appropriately maintain their roads. It would also allow for fees to be varied based on the travel zone in which the VMT occurred. For example, it may be advantageous to use different per mile prices for rural, suburban, urban and congested commercial business district (CBD) roads. This idea is not unlike cordon pricing systems implemented in a number of major international cities (Larson and Sasanuma 2010). Under cordon pricing systems, drivers are assessed a fee based on each entry in to or each day spent in the cordon zone. However, what we propose extends those approaches to allow one to charge based on the miles travelled in the zone.

Depending on the complexity of the system, it may be possible to further vary road use fees based on direction of travel in to or out of the zone, time of travel, vehicle type, vehicle emissions, or fuel efficiency. The gas tax inherently rewards high fuel efficiency vehicles, but does not allow for more sophisticated pricing structures which would enable policy makers to incentivize driver behavior that reduces road wear, congestion, emissions, and increases transit use.

Additional advantages, drawbacks, and considerations for implementing an MBUF system as it pertains to Minnesota were identified by a taskforce led by the Minnesota DOT containing representatives from a number of sectors. (MBUF Policy Task Force 2011).

We note that the use of terms such as mileage based user fees (MBUF) and vehicle miles traveled (VMT) indicate miles as the unit of measurement. It would be more correct to say distance based user fees or distance traveled, however the terms MBUF and VMT are very popular and understood as a part of the vernacular shared by transportation professionals in the United States. It is acknowledged that internationally, metric specific or unit indifferent terminology is used.

### 1.3 Requirements for MBUF Systems

In designing a system capable of determining both a vehicle's location and VMT and then assessing user fees based on that information, a number of issues must be

considered. It is important to first determine the technical requirements for the system. This provides a set of guidelines for the design process and allows for informed decision-making where trade-offs between requirements are necessary. Additionally, determining and explaining these requirements also provides a means and vocabulary with which to evaluate and compare potential solutions.

Any MBUF system must be accurate. The accuracy of a system is simply a measure of how correct the system is in determining the distance a vehicle travels and where this travel occurs. Although an intuitive requirement, there is some ambiguity in determining the distance a vehicle travels. The physical distance over the ground that a vehicle travels is a logical choice to use in defining VMT. However, this distance, is not necessarily the same as the distance determined by the vehicle's odometer, which is the legal definition of the distance a vehicle travels.

If the system is sufficiently accurate, it will also necessarily be repeatable. This means that for identical trips, the system must have the same output (VMT, user charges, etc.) each time. Otherwise, identical trips could be assessed different fees, which would be confusing and unfair to users. Predictability allows for greater faith in the performance of the system.

The system should be informative. That is to say, the system must inform the driver about the charges they will incur. It would be unfair for drivers to not know the rates they face when using their vehicle. Additionally, this information would allow for informed decisions to be made regarding when and where drivers choose to travel.

Auditability is another characteristic to consider. Ideally, there would be a convenient way for drivers to review not only the charges they are assessed, but also the trips that caused them. Allowing for this provides a method by which system errors can be identified and corrected. Even in cases where the system is behaving correctly, users will trust the system more if they are able to confirm that the trips the system registers matches the trips that they actually made.

As with any system capable of assessing charges, security becomes an important consideration. MBUF systems must be enforceable and tamperproof. Enforceablility means that there is a logistically feasible way for road use charges to be assessed and collected. Furthermore, those who attempt to evade payment must face consequences. This includes physical and electronic tampering. It must be prohibitively difficult to physically open, disable, or otherwise modify the system's hardware. Likewise, the entity that stores user charges and other information must be secured against unauthorized remote access. If the system is tampered with somehow, it must be detectable so that the responsible parties can be determined.

It is also important to consider how well the system maintains user privacy. Clearly, the system must never share the users' location with anyone other than those using the information to calculate and assess the corresponding road use fees. Additionally, it should collect only the minimum information needed to perform its function.

These privacy constraints imply that the method by which the vehicle is located should not be any more precise than it needs to be. In this sense, precision refers to how small of

an area in which a vehicle is being located. This depends on the technologies used to locate the vehicle. GPS based methods might locate a vehicle to within 1 to 3 meters, whereas a pen and paper system includes no information at all about the location in which VMT occurred. The solution proposed in this document bridges the gap between these two extremes by developing a system that is only capable of determining the location of a vehicle to within a jurisdiction or travel zone, for example a city, county, or state. In this case, the system is not determining the exact location of the vehicle, but rather determining the zone in which the vehicle is traveling.

In addition to not violating the users' privacy, the system must also be both understandable and sufficiently merit the users' trust so that the users *believe* that their privacy isn't being violated. Public acceptance and understanding frequently trumps the technological and legal realities when determining the success of a system (Douma and Aue 2011). The disconnect between system privacy and perceived privacy is due to misunderstandings and confusion about how MBUF enabling technologies work and their limitations.

For example, it is a common misconception that GPS, by itself, can inherently track its users. Even though this isn't the case, it can be very difficult to shake the idea of Big Brother when dealing with GPS. Confusion about this is compounded by the ubiquity of smart phones, which due to their cellular modem are capable of sharing their users' location information. For this reason, systems that use GPS have an inherent challenge when dealing with the perception of privacy, especially in the United States. Thus, the approach taken here is to simply not acquire GPS based location in the first place. If a

system can calculate the distance traveled and the associated travel zone without GPS, then there is no need to use it, especially if user acceptance is problematic.

## 1.4 Methods for MBUF Collection

Many technologies have been proposed for congestion pricing. The U.S. D.O.T. describes a number of these in their primer on the subject (Chu 2008) ranging from simple methods such as paper based systems or gantry-based tolling to more sophisticated means such as GPS/GNSS or cellular-based systems. Sorensen et al. reviews the many alternatives that can realistically be applied in the near future (Sorensen et al. 2009) and in a second report (Sorensen et al. 2010) outlines what needs to be done next to move forward with their deployment. Between the Sorensen reports and the previous work performed in (Donath et al. 2009) a further review was unnecessary.

### 1.4.1 Paper Based MBUF

The pen and paper method is the simplest method to implement an MBUF system. Using this method, drivers report the number of miles they drive, to the agency responsible for assessing the road use fees. The agency then determines the cost associated with that VMT and then assesses the corresponding fee. To ensure compliance, periodic audits of vehicle odometers would have to be performed. This could occur at certified auto shops similar to the emissions inspections process in New York or California. Alternatively, these audits could be performed at government service buildings such as driver's license

stations or vehicle registration offices. Although simple to implement, it would still require additional oversight, and therefore additional resources, to administer.

The advantage of the paper-based method is that it's simple and understandable. However, this is also its weakness. Such a system lacks the ability to vary charging by travel zone, nor can it fairly attribute revenue to those jurisdictions in which the travel occurred. Additionally, charging by other factors such as time or direction of travel is not possible. This severely limits its ability to incentivize beneficial driver behavior. It also requires additional paperwork on the part of the drivers where a more automated electronic system might require little to no extra work.

### 1.4.2 GPS

The most common and simplest technology capable of aggregating VMT by travel zone is GPS/GNSS. Using a GPS receiver to provide accurate and precise location data about where the vehicle has been, the system can then determine the distance the vehicle traveled in each travel zone. This can then be used to assess the appropriate user fees. Additionally, it is also possible to consider time of travel and use different pricing for peak and off-peak times.

We note that the term Global Navigation Satellite System (GNSS), refers to all satellite based navigation technology, not just the Global Positioning System (GPS), which refers to the GNSS created and operated by the United States. Using the term GNSS would be more correct as it is not specific to any country's system and could refer to GPS, GLONASS (Russia), Galileo (Europe), or Compass (China). The term GPS will continue

to be used here due to its familiarity and understanding among the audience of this document. All GNSS based MBUF systems would operate similarly.

MBUF through GPS is well documented in the academic literature having been used successfully in a number of studies in both the US and in Europe. Of these, perhaps the most notable was the successful deployment of a GPS based VMT pilot program in Oregon (Whitty 2007). This program, conducted by the Oregon DOT, showed that MBUF was a viable option. They also addressed small-scale implementation issues regarding how such a system could successfully be phased in incrementally over time where some vehicles would have the system, but others would continue to use the gas tax. The issue of privacy was also addressed in that their system demonstrated how a GPS based system could perform with high accuracy and preserve user privacy.

Another successful implementation of GPS based MBUF is in European trucking. Germany and other European countries use GPS enabled hardware to determine the distance trucks travel within their borders (Broaddus and Gertz 2008). Here, aggregating the VMT by travel zone is very important, so that countries only charge truck operators for the VMT occurring within their borders, but not in other countries.

There is also technology available on the market that is similar to MBUF capable technology, but is used for fleet tracking and pay as you drive (PAYD) insurance. One product not unlike others is the ROVR™ - Real-time Onboard Vehicle Reporting developed by TransCore (TransCore 2012). This product is equipped with GPS, a GSM modem, and the unit itself plugs in to the OBD-II port of a vehicle. As explained in

greater detail later in this document, the on-board diagnostics (OBD-II) port is a connection to the vehicle's on-board data bus that allows for an external computer to read information from the vehicle such as speed, emissions data, engine state, among others. This combination of hardware allows the ROVR to determine the vehicle's location, speed, distance traveled as well as allowing the modem to transmit this information to a remote computer.

GPS's greatest advantage is that it generates a reasonably accurate and precise location of the vehicle. This provides a means to determine the vehicle's current travel zone, as well as source of vehicle position and speed from which the VMT may be calculated. Although there are advantages associated with using GPS, there are also drawbacks. Technical limitations such as location fix times and issues with urban canyons hurt the efficacy of such a system. Additionally, having access to vehicle location information is convenient for determining VMT aggregated by travel zone, but is ultimately unnecessarily precise. This, coupled with popular but incorrect conceptions of how GPS functions, negatively affects the perceived privacy of such a system which may be the largest challenge for such a system used to determine road user charges.

### 1.4.3 Cellular Based VMT System

The issues with GPS based MBUF systems motivate a solution that is more advanced than paper based systems but still allows for sophisticated road pricing options. The solution is to use cellular networks to determine the zone in which the vehicle is traveling. Everyone will agree that cellular-based location is not as precise as GPS, but

given the general concern about protecting one's privacy, this serves as an advantage. Furthermore, cellular-based MBUF systems are also not affected by the other issues associated with GPS based systems such as long fix-times and line of sight to sky requirements.

Cellular network based travel zone determination can be performed using two different approaches; multilateration and cellular assignment. Both methods rely on the modem's ability to communicate with nearby cell towers, but they use the resulting information in different ways.

The first method, multilateration (sometimes referred to as triangulation), is based on determining the difference in distances between the modem and multiple nearby cell towers. This process is called time difference of arrival (TDOA). These differences in distances are calculated by detecting the differences in the arrival times of precise timing information sent from the cell towers. These differences are then combined with a priori information about the locations of each cell tower, which then reduces determining the location of the modem to a geometry problem. This yields the modem's location (although it is still less precise and less accurate than GPS). To determine MBUF, the location can then be used similarly to locations provided by a GPS receiver.

The major issue with using this method is that it requires the exact location of all cell towers, information that is not freely available. Therefore determining a modem's location through multilateration is restricted to cellular carriers or those who have the resources to invest in purchasing or otherwise determining the towers' locations. For

example, when a call is placed from a cell phone to 911, the Enhanced 911 (E911) systems are able to provide the location of the caller. However in this case, the location is calculated and provided to the 911 call center, or Public Service Access Point (PSAP), by the cellular carriers. The call center cannot determine the caller's location by itself.

The other way to determine location information using a cellular modem is through cellular assignment. This is the approach we propose and describe in greater detail later in this document. The basis for such a system is the existence a of a cell identification (CID) code that uniquely identifies each cellular tower. A tower's CID can be "read" by any cellular modem in radio contact with that tower, and unlike data, voice, or SMS transmission, this communication is free. As a cellular modem physically moves within an area with cellular coverage, the towers with which it can communicate and the signal strength thereof, as measured by the received signal strength indication (RSSI), changes. It is our assertion that a machine learning algorithm can use the observability and strength of nearby towers to determine the current travel zone.

This is done in two phases. The first is a data collection, or training phase, consisting of physically traveling in areas where the system is to be deployed. While doing so, the system records the CIDs and RSSIs of the cell towers that the cellular modem can observe. Additionally, the system records the GPS-provided location of each reading. This is used to determine the travel zone in which each reading was made which is in turn used to associate CIDs with travel zones. The result of the first phase is a table, referred to as the training set that contains a number of readings. Each of these readings consists of pairs of CID and RSSI that were "visible" when the reading was taken. Each reading is

also associated with the location in which the reading was taken. In operation, a system would need only to save the travel zone in which each reading was taken. Here, the location itself was saved to facilitate boundary changes.

Collected data is then used in a second, operational phase that no longer uses GPS to determine the vehicle's location. In order to determine the vehicle's current travel zone, the system uses the table of previously collected data. It does so by determining what cell towers it can currently observe and then compares these to the training set. The algorithm determines the vehicle's current travel zone based on the travel zones in which the most similar previous observations were made. By relying on the training set in this way, the system doesn't require knowledge of the exact locations of the cell towers.

Under normal operation, the method is incapable of determining a vehicle's location with a precision smaller than the size of a travel zone. This characteristic allows for better privacy for system users. However, the enhanced privacy comes at a price. In using cellular assignment, additional considerations must be made that may not be necessary with other methods.

The system clearly requires training data wherever the travel zone must be determined. To reduce the amount of data needed, a machine learning algorithm may be selected that only requires that data be collected along zone boundaries. However, this means that care must be taken to ensure that a vehicle passing through a boundary can't do so without detecting the change in travel zones. Additionally, if the system is turned on for the first time in an area without training data, for example in the middle of a very large travel

zone, the installer or registration agency should be able to inform the system about its initial travel zone.

Another potential issue with using cellular assignment is that unlike GPS, the method to determine the travel zone is insufficient to determine the vehicle's VMT. Because of this, vehicle distance is determined through other means.

The simplest way to determine VMT is through the use of data from the vehicle's data bus accessible via the on-board diagnostics (OBD-II) port. This port is included on all vehicles made after 1996, and is generally accessible under the dashboard near the steering wheel. Figure 1 shows the location of the OBD-II port in a typical vehicle.

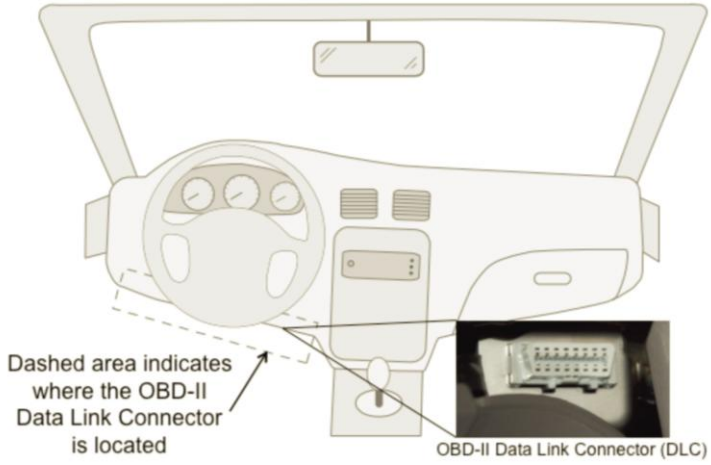


**Figure 1: Image and Location of OBD-II Connector**

The OBD-II port provides, along with other information, the vehicle's current speed. The speed can be numerically integrated to obtain distance which can then be matched with the respective travel zones as determined by the machine learning algorithm, thus aggregating VMT by travel zone. A representation of the process by which the calculated

VMT is combined with the current travel zone is shown in Figure 2.

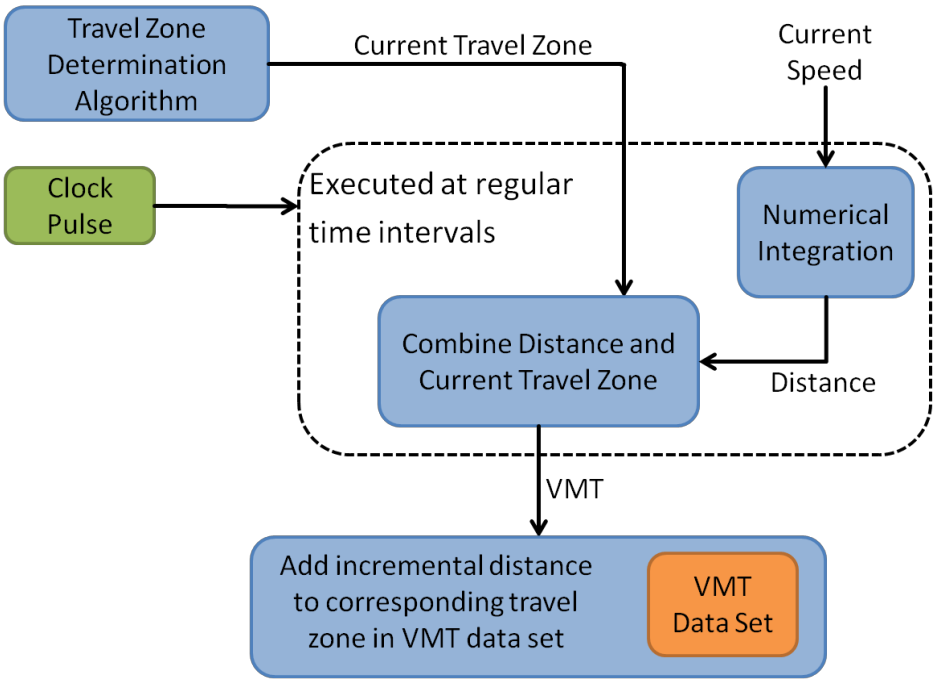


**Figure 2: Flow Diagram of MBUF Software**

### 1.5 Additional MBUF System Components

As described above, there are multiple enabling technologies that could be implemented in an MBUF system. We now discuss how the VMT and travel zone data is combined and converted to a road use fee. Once the VMT is aggregated for the different travel zones in which the vehicle has driven, this information can then be used to assess road user charges. Depending on the desired level of privacy, several alternatives exist.

In all cases, some form of data needs to be sent to a "back office" where billing and other administrative functions are based. In the method that preserves the most user privacy, the hardware installed in the vehicle keeps track of VMT per zone as well as current road

pricing data. It then pairs this information internally and only transmits the final user charge identified for that vehicle. This would remove the need for the hardware to ever share the driver's location with anyone, including the driver or the back office.

Alternatively, the system could assess road user charges by having the hardware send the VMT information to the back office, which would then calculate the charges using the rates stored in the back office database. Although this would involve transmitting more information about the vehicle's trips, this would allow for greater auditability in the event of a dispute.

In both these situations, data must be transferred from the MBUF hardware to the back office. This system was designed such that the cellular modem used to determine the current travel zone through cellular assignment can also be used to communicate with the back office. One way to do this would be through SMS text messaging. This is a convenient way to send information because the SMS protocol is mature and has features such as receipt acknowledgement. The disadvantage to using this method is that sending and receiving text messages can be expensive for a system that frequently sends messages to communicate its data. Additionally, depending on the system, the size of a text message might far exceed the amount of information that needs to be sent, but to send any data at all, an entire text message must be used and paid for.

Another way to facilitate communication between the in-vehicle hardware and the back office would be to implement a custom solution that sends information via the cellular data network. Although data on a per kilobyte basis can be expensive, such a system

could take advantage of how little information needs to be sent. This could be a more efficient usage of data transmission because only the data that needs to be sent would be paid for.

The issues of how the system communicates with the back office and what information it sends is an important and non-trivial consideration. However, such a discussion is beyond the scope of this document. Additional information about these considerations can be found in (Donath et al. 2009).

### 1.6 Project Scope

There are many important considerations to be made when designing, implementing, and evaluating MBUF systems. As touched on above, these range from policy decisions about how the fee structures are determined to technical issues about the nature of device to back office communication. Although these considerations merit additional discussion, this document has a limited focus. We will briefly describe a preliminary experiment designed to determine the feasibility of using numerically integrated speed from the OBD-II port to determine distances. This experiment was limited in scope and only served to compare the integrated speed distances to the distances recorded by the vehicle odometer.

Although the accuracy of the calculated distances is important, a more critical factor in determining the accuracy of the system is the ability of the machine learning algorithm to

correctly determine the zone in which the vehicle is traveling. The main focus of the work described here, was to develop and evaluate an algorithm capable of determining a vehicle's current travel zone through cellular assignment.

# Chapter 2: Pilot Study of VMT Calculated Based on OBD-II Speed

## 2.1 Introduction

Clearly, a major consideration of a system that aggregates VMT by travel zone is the accuracy with which VMT is calculated. In doing so, it is first necessary to identify factors that could affect the accuracy of the calculation. Vehicles generally determine their speed indirectly by measuring the angular velocity of the wheel axle. Numerical integration is used to calculate the VMT. Because of this, it's possible that the accuracy of the distances calculated will depend on tire pressure (which ultimately affects the effective wheel circumference). Additionally, the system accuracy may also be affected by driving conditions. Specifically, the driving profile of the vehicle may affect how closely the reported speed tracks the actual speed. System accuracy may also depend on the numerical integration technique used. Different algorithms may behave differently under dissimilar conditions. Lastly, vehicle type may also affect the accuracy of the system and should be considered.

The distance that the system calculates should match, as closely as possible, the actual distance the vehicle travels over the ground, or ground truth distance. However, it is more important that the calculated distance matches the odometer. Although the odometer may not be truly accurate as compared to the ground truth distance, it is the legal definition of how far a vehicle has traveled. For this reason, our experiment will examine the error between the calculated VMT and the vehicle's odometer, not the ground truth distance.

## 2.2 Methods

The experiment considers four factors: vehicle, driving conditions, tire pressure, and integration technique. Two vehicles were used in order to better understand vehicle-to-vehicle differences. Although, using only two vehicles doesn't give complete information about all vehicles, it does help to begin to understand these differences. Additionally, the vehicle factor was considered to be a block, an experimental design method used to account for variation between factors. This allowed the data collected on both vehicles to be used together to increase the ability to detect differences between the treatments, while still taking into consideration the differences between vehicles.

Two driving conditions were considered. The first was highway driving which consisted of an approximately 22-mile loop through the Minneapolis – Saint Paul metro area. The other was in the city, which consisted of an 8-mile route that included downtown Minneapolis and portions of the University of Minnesota campus. These two conditions attempted to generate two very different driving profiles. The highway profile consisting of mainly constant speeds, and the city profile consisting of frequent accelerations, decelerations, idling, and turns. Although it is noteworthy that the two routes have different distances, the way in which the error was calculated accounts for this by normalizing the difference between the odometer reading and the calculated distances. This is discussed further in Appendix A.

Three levels of tire pressure were evaluated. Both vehicles had a recommended tire pressure of 30 pounds per square inch. This pressure, along with 35 and 25 pounds per

square inch represented the largest range of tire pressures that could be used and still be safe and non-destructive to the tires or vehicles.

Two integration techniques were used: the Riemann sum method, and the trapezoid method. The integration technique factor has no random error associated with it, so all its levels (i.e. both methods) could be applied to each combination of the other three factors.

All trials were conducted in the course of a single night. This eliminated day-to-day variation. By running the trials at night, this greatly reduced the trip-to-trip variation due to traffic. At the beginning of each collection, the vehicle was warmed up and driven to ensure that the tire pressure would not change mid-trip due to temperature variation. Then before each trip, the assigned tire pressure was applied to each of the four tires. While at a stop, but with the vehicle running, the trip odometer was reset and the program collecting speed data was started. This program simply collected speed and timestamp information from the OBD-II port. The speed was later integrated offline using both of the techniques.

### 2.3 Results

A preliminary analysis of the data indicated that the integration technique has no effect on the system error. This is readily apparent, even without a formal analysis, because for each factor level combination of vehicle type, driving conditions, and tire pressure, the two techniques yield identical errors. Based on this observation, the remainder of the

analysis was performed with a single integration technique and only considered three factors instead of four.

The statistical package R was used to assist in the analysis by generating an analysis of variance table (ANOVA). The ANOVA table showed that no factor or factor interaction term was significant. This means that the vehicle type, the driving conditions, nor the tire pressure had a significant effect on the error of the calculated VMT as compared to the odometer.

The mean error over all the trials was calculated and a z-test was used to determine a 95% confidence interval on the mean. This yielded a mean error of 0.90% ± 0.2%. The maximum error from a trial was 1.5%. It is noteworthy that not only is the mean a positive value, but each trial's error is also positive. Due to the way the error was defined, positive error values correspond to the situation where the calculated VMT is less than the odometer-determined distance.

### 2.4 Discussion

Although the analysis implies that there is no significant difference between any of the treatments, it's important to consider that for the small sample sizes used here, this experiment may have simply failed to capture any differences. If there are differences between the treatments, they are minute and to detect smaller differences, one needs larger sets of data.

We did, however, identify some issues that ought to be considered before repeating this or conducting a similar experiment. Before the experiment, there was very little known about how such a system would perform. There is a strong indication that the integration technique has little or no effect on the system error. In order to detect differences due to the other three factors, the trip length must be longer so the issue of odometer resolution is mitigated. In addition to using longer trips, there should also be more trips, to again increase the ability of the experiment to detect smaller differences in the resulting errors.

In considering potential differences due to treatments, or a combination of each of the factors, as a source of random error and looking at the data as a collection of identical treatments, an overall estimation of VMT accuracy can be estimated to be 0.90% ± 0.2% at a 95% confidence interval. It is noteworthy the errors for every trial were positive, indicating a bias in the errors. This suggests that with further experimentation and better error characterization that it may be possible to correct for this bias by adding to or multiplying the calculated VMT by a correction term. Even without making any bias corrections, the worst trial had 1.5% error between the calculated VMT and the odometer. This is a good indication that calculating distances by integrating the speed from an OBD-II port is a viable method for determining VMT, and with additional experimentation, bias corrections could increase the accuracy.

# Chapter 3: Travel Zone Determination Algorithm

## 3.1 Introduction

As described above in Section 1.6, the primary focus of this project is to determine a suitable machine learning algorithm capable of determining a vehicle's current travel zone using cellular network information. Choosing a machine learning algorithm for this task proved to be a significant portion of the work associated with this project. In the end, an algorithm was selected, but not before considering a number of others. The process by which the current algorithm was selected as well as the motivations for selecting and then later rejecting other algorithms is detailed in Appendix C. This chapter documents the criteria by which these algorithms were evaluated as well as describing the algorithm ultimately selected.

## 3.2 Algorithm Considerations

Section 1.3 discusses several key requirements for an MBUF system to be viable. Clearly the algorithm used to accomplish the travel zone determination will greatly affect some of these criteria such as accuracy and precision. However, the algorithm may not affect other requirements at all such as for example, enforceability and how informative the system is to drivers. Therefore, in trying to maximize overall system performance, it is also necessary to maximize algorithm performance, especially in areas where the algorithm directly affects this overall system performance.

However, in determining a suitable algorithm for this application, simply finding the one

that maximizes accuracy and precision is insufficient. There are additional considerations that must be made to address the algorithm's implementation on cost effective hardware. This is especially important in the context of this project where all the software must run in real time on low power, low cost, physically compact hardware, which limits hardware specifications such as processing power and memory. Algorithms that have good performance in theory must be rejected if their implementation is not feasible.

The algorithm's computational expense is a major factor in determining whether or not it is suitable for implementation. A computationally expensive algorithm would be one that is very complex and requires many calculations and therefore clock cycles to perform. It would execute relatively slowly or require a fast processor. This is particularly important in this application where the algorithm may need to execute in real time on potentially limited hardware.

Another important factor to consider is the amount of memory the algorithm requires. This includes storage for intermediate calculations, parameterized or statistical data, or a training set. Clearly, the storage of data to support the algorithm's execution must not exceed the available storage, and must leave enough space for storing the calculated VMT, fee structure, accrued charges, etc.

These two characteristics must be balanced against each other and performance when choosing an algorithm.

### 3.3 K-Nearest Neighbors Algorithm

The machine learning algorithm selected to perform the travel zone determination is a simple k-nearest neighbors (KNN) approach. KNN is a supervised, non-parametric machine learning algorithm that relies on a large training set to determine which zone is associated with the current reading. The algorithm takes the reading, consisting of one or more CID/RSSI pairs and then looks through all members of the training set, then finds the k most similar readings. Here, similarity is defined by Euclidian distance when considering each reading as an n dimensional vector where n is the number of all possible CIDs. Using this definition, the k most similar readings are the k closest vectors (or nearest neighbors). The algorithm then looks at the travel zones from which each of the k-nearest neighbors originated. The travel zone that produced the most of the k nearest neighbors is determined to be the most likely travel zone for the current reading. Additional information about the KNN algorithm is available in most machine learning textbooks, for example (Bishop 2006).

Determining what k should be was addressed in a straight forward manner. Multiple k values were used in trial analyses (described in the next chapter) and the value leading to the best performance was selected. In this case, k was chosen to be 11.

In applying the standard KNN algorithm to the task of travel zone determination, a zone filtering step is conducted to every reading taken by the modem. Before performing the KNN algorithm, the software first determines if any of the cell IDs observed, correspond to only a single travel zone. If this is the case, the system can determine the vehicle is

traveling in the zone corresponding to that cell ID. This allows the system to only perform the KNN classification when needed. The zone filtering and its role in the zone determination algorithm is illustrated in Figure 3.

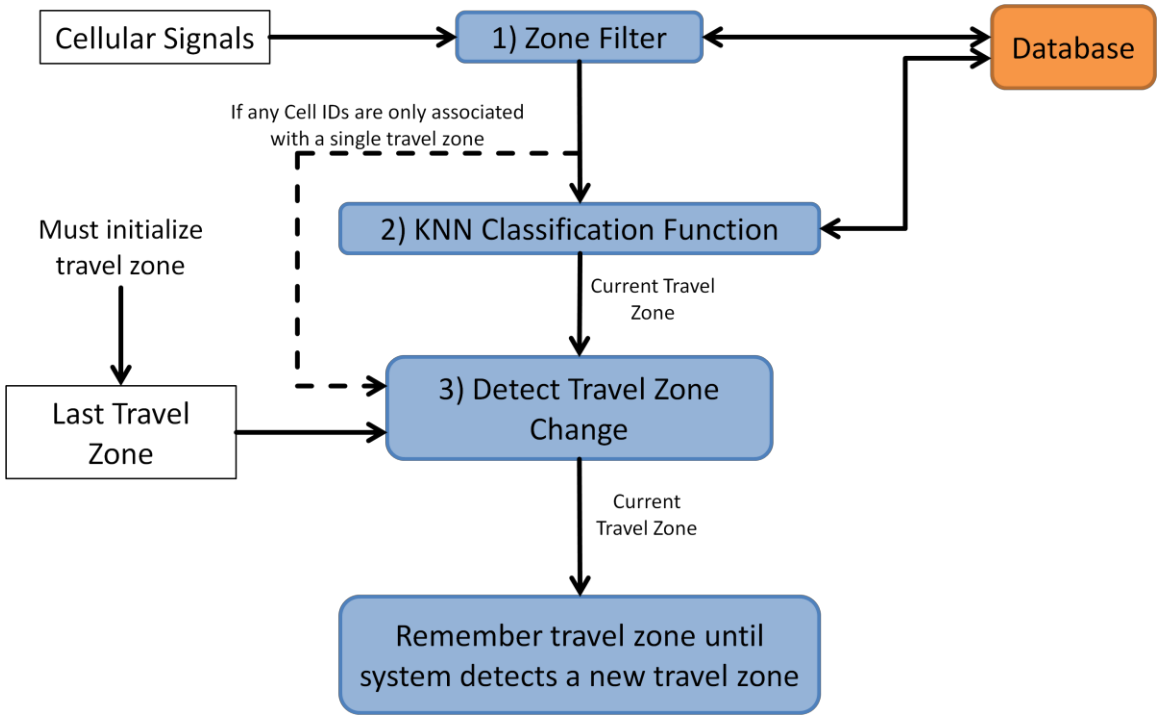


**Figure 3: Zone Determination Algorithm**

In applying the standard KNN algorithm to the task of travel zone determination, a pre-processing step was added to remove two types of unnecessary data from the training set. First, the data collection was conducted in such a way that often, data would be collected while a vehicle was stationary. This collection doesn't add useful information, and for the sake of reducing computing time, can be eliminated. To do this, observations made less than 4 meters from the last observation were omitted.

Second, due to the size of the cells, the area to which a single cell tower provides

coverage, the visible CIDs don't change drastically with small changes in position. This means that when a vehicle is near a travel zone boundary, the CIDs visible on one side of the boundary may not be significantly different from those seen on the other side. The issue here, and consequently the focus of this study, is to identify how far away the vehicle must be from the boundary in order to encounter sufficiently different CIDs.

In an attempt to assist the algorithm in making these travel zone determinations, it was found to be beneficial to remove training set data very near the boundary. Through trial and error 100 meters was determined to be a distance within which, there was no useful training data. This is to say that removing the data within ±100m of the edge of a travel zone increased the performance of the algorithm and as a result of reducing the training set, also reduced computation time.

# Chapter 4: Methods for Evaluating the Zone Determination Algorithm

## 4.1 Introduction

Upon the selection of the KNN machine learning algorithm, it was then necessary to develop and conduct a series of experiments to evaluate and document its performance. This chapter discuses this experiment including: the location examined and the rationale for its selection, the hardware used to conduct the evaluation, the data collection, and the analysis performed on the data.

## 4.2 Location

The location used for this experiment was the commercial business district (CBD), or the downtown area, of Minneapolis, Minnesota. One might consider the CBD as a congestion priced travel zone, which would be priced so that travelers would be encouraged to park their vehicles in parking facilities on the periphery of the zone as opposed to driving within the zone. Residents who live in the travel zone could be treated differently and would not be charged the congestion fee. Of particular interest is the evaluation of a cellular tower based approach in an area where GPS suffers issues due to the presence of urban canyons.

For reference, a downtown travel zone was created. This area is a 12 by 8 block (roughly 0.6 square miles) section of Minneapolis, more or less representing what is considered to be the downtown area. However, the exact boundaries of the corresponding travel zone,

as shown in Figure 4, are arbitrary and simply selected for purposes of the experiments described here.

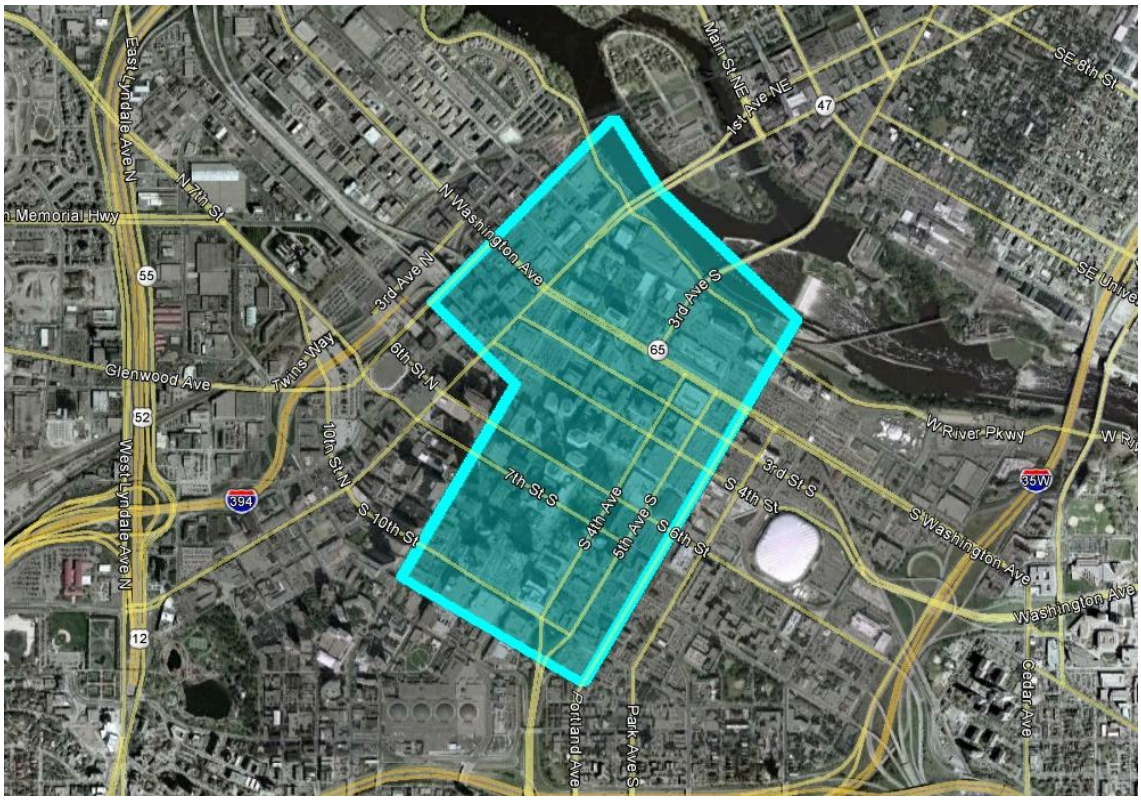


**Figure 4: Downtown Minneapolis CBD Travel Zone Considered in the Experiment**
Image: ©2012 U.S. Geological Survey, Sanborn. Map Data: ©2012 Google.

Note that the roads in the downtown area of Minneapolis form a grid, but this grid is not exactly aligned with the cardinal directions. In the context of downtown Minneapolis, what will be referred to as north will actually be "grid north" which is roughly a 30° clockwise rotation from true north.

For the purposes of evaluating the algorithm, three test areas were considered in and around the CBD. Test areas consisted of data taken along a given road marked in the

following figures with yellow (and in one case, orange) shaded rectangles. All available data within each test area were considered regardless of the original direction of travel or the street on which the reading was made. This data was then validated using a given boundary marked in the following figures with blue, green, white, or pink lines. Test areas were selected as they were representative of three variations of urban, commercial environments. The exact boundaries of the roads considered, as shown below, were selected based on the availability of data. The process by which the data was collected is discussed further in Section 4.4. The test areas used in this experiment were defined along Washington Ave. South, First Ave. North, as well as Eighth and Ninth Streets South. These three test areas are shown in Figure 5 along with the downtown boundary shown in Figure 4.

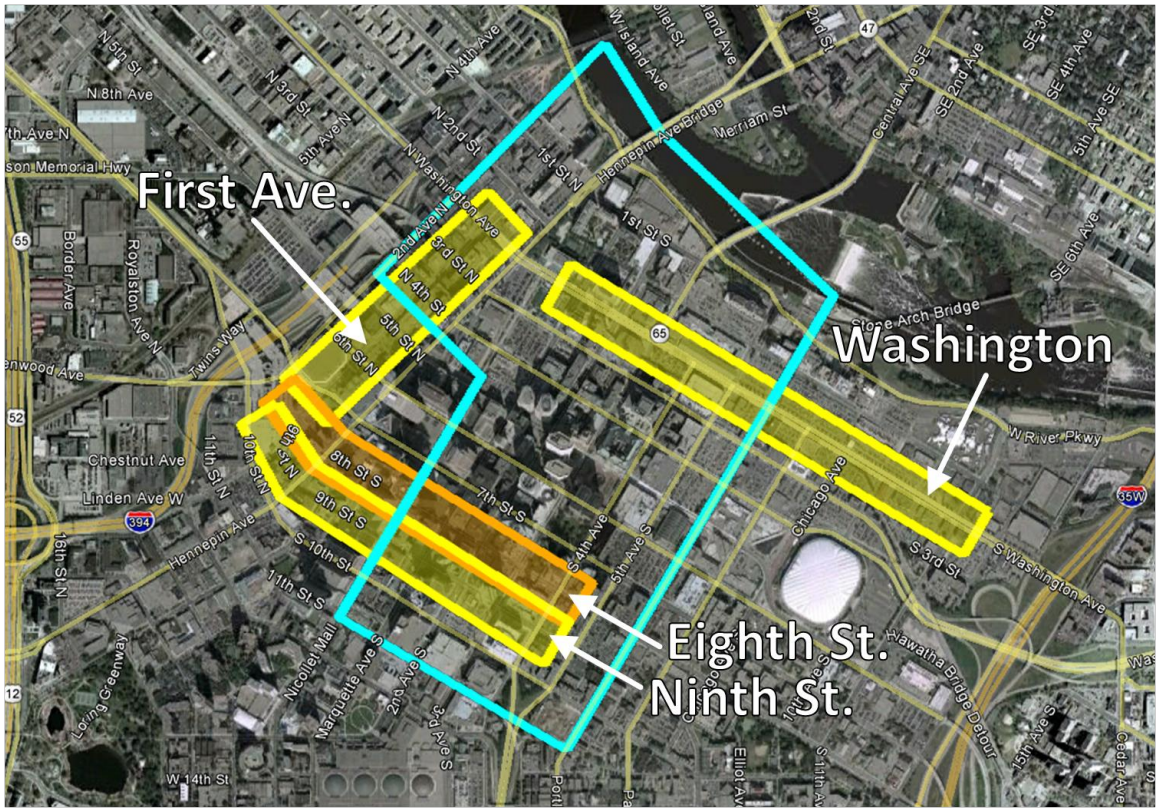


**Figure 5: Downtown Minneapolis Test Areas**
Image: ©2012 U.S. Geological Survey, Sanborn. Map Data: ©2012 Google.

### 4.2.1 Washington Ave

The first test area was a section of Washington Ave. South extending from Marquette Ave. South to 11th Ave. South and in the process, crossing through the downtown travel zone boundary shown in Figure 4 at Portland Ave. South. Additionally, this stretch of Washington Ave. South was also analyzed using different travel zone boundaries. This was done by repeating the analysis, but with the eastern edge of the downtown travel zone shifted to Chicago Ave. South and again with the edge of the zone shifted to 4th Ave. South. This corresponds to a two block shift of the boundary in the east and west

directions respectively. The boundary of the nominal downtown travel zone from Figure 4 is shown in light blue. The two alternate boundaries are shown in green and white. Figure 6 shows a close-up view of this test area with streets labeled.

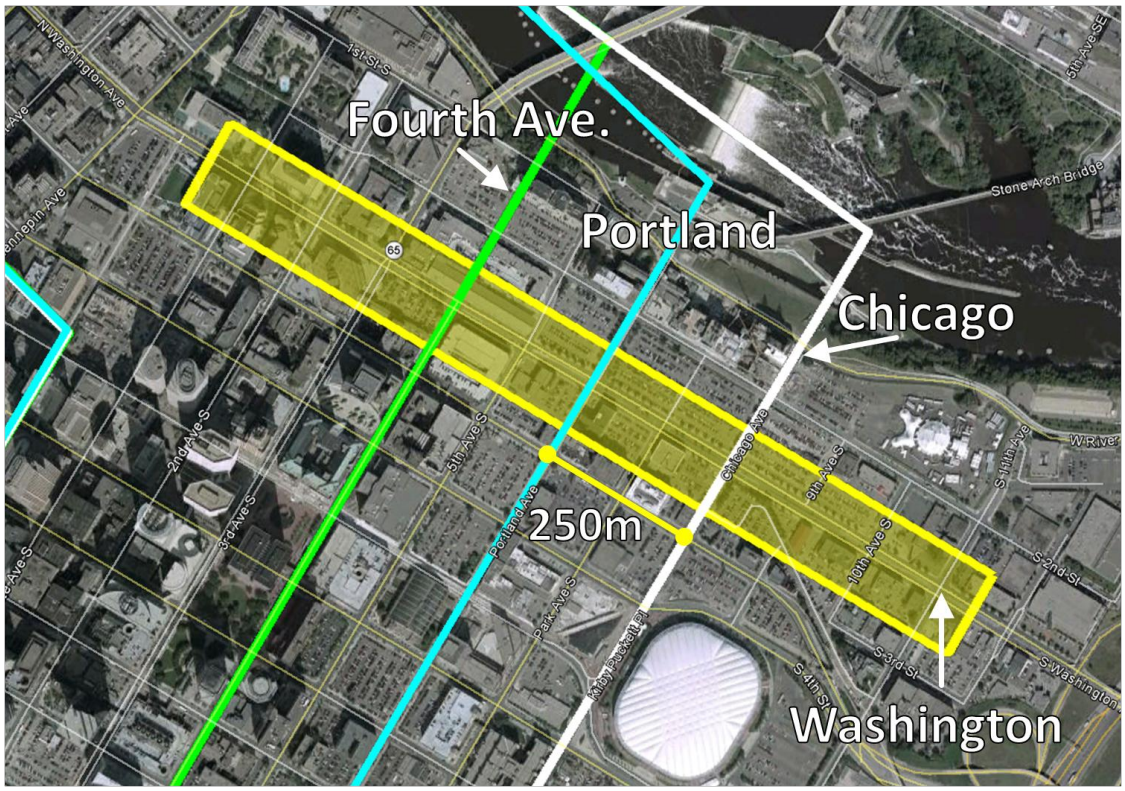


**Figure 6: Test Area 1 – Washington Ave (Showing All Alternate Boundaries)**
Image: ©2012 U.S. Geological Survey, Sanborn. Map Data: ©2012 Google.

### 4.2.2 First Ave

The second test area considered was a stretch of First Ave. South near the western edge of the downtown zone. This test area features a boundary that does not run along a street, but rather through a block, between 4$^{th}$ and 5$^{th}$ Streets South. This boundary was chosen as it roughly bisected the test area. The second test area is shown in Figure 7.

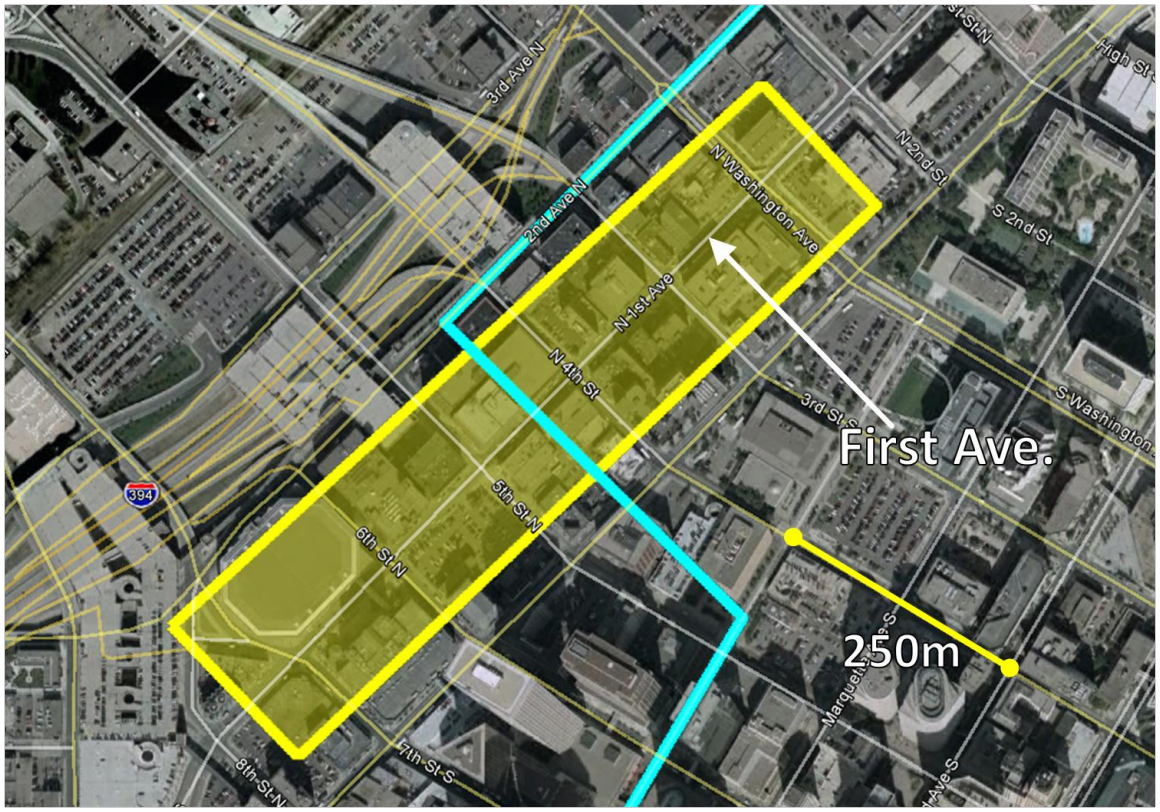


**Figure 7: Test Area 2 – First Ave**
Image: ©2012 U.S. Geological Survey, Sanborn. Map Data: ©2012 Google.

### 4.2.3 Eighth and Ninth Streets

The third test area considered was a collection of four intersections defined by test areas running along the east-west streets South 9$^{th}$ and South 8$^{th}$, shown in yellow and orange respectively as they intersect the nominal downtown boundary along Nicollet Mall (shown in teal) and an alternate boundary along Marquette Ave. (shown in pink). These test areas and zone boundaries are of particular interest because they are located in urban canyons. Each of the four intersections considered are marked with circles labeled A through D. This is shown in Figure 8. These are identified in order to discuss the results

for a given test area and boundary combination.

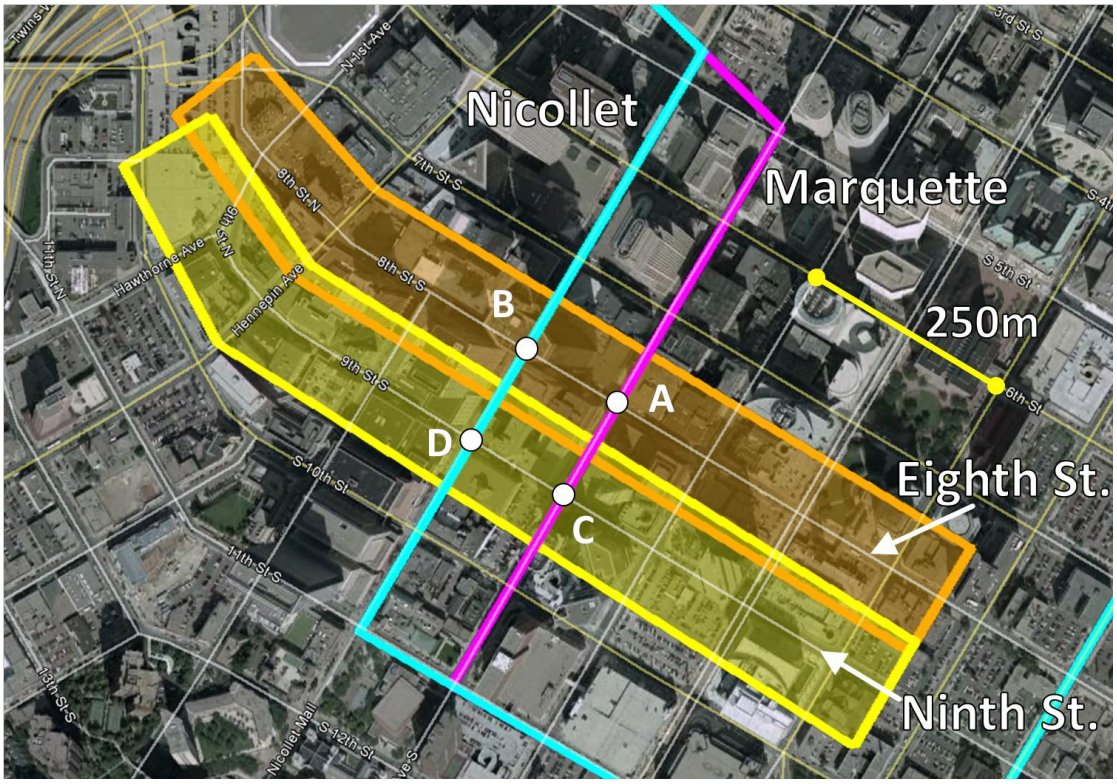


**Figure 8: Test Area 3 – Eighth and Ninth Streets**
Image: ©2012 U.S. Geological Survey, Sanborn. Map Data: ©2012 Google.

### 4.3 Hardware

To collect the data for the training phase of the experiment, an embedded computer was connected to a GPS receiver and a cellular modem. Additional information about the current hardware configuration and previous hardware solutions is provided in Appendix B. Figure 9 provides a visual representation of the hardware organization.

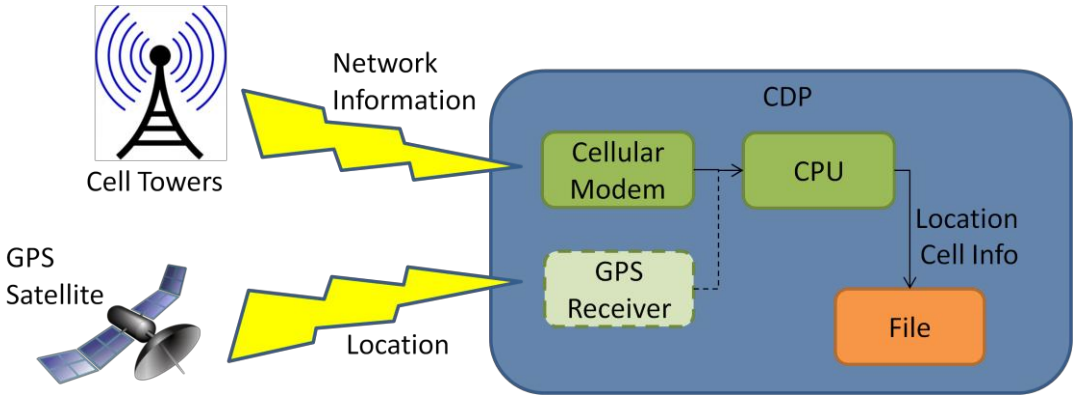


**Figure 9: Visual Representation of Hardware**

The software that ran on the CPU periodically queries the GPS receiver for the vehicle's location. While doing this, it also queries the cellular modem for the list of CIDs corresponding to the cell towers it can observe as well as the RSSI values associated with each of these towers. This data is then saved to a file for later analysis. Note: The GPS receiver was used only for training and testing the system. GPS would not be used in a final implementation of the MBUF system.

### 4.4 Data Collection

Data collection was performed by physically driving the areas of interest with the hardware that logs the data from the GPS receiver and cellular modem. The data was recorded into files, each corresponding to a single data collection trip consisting of a single vehicle and day combination.

To facilitate rapid data collection, four hardware units were installed in Minneapolis traffic control vehicles. Constraints prohibited prescribing specific routes for the vehicles

to follow, as they had their own routes to follow focusing their travel on roads with parking meters or other parking restrictions. However, vehicles could be selected whose daily routes were in or through useful areas. This yielded data that came not only from crossing the travel zone boundaries, but also from driving parallel to those boundaries. If we could have selected the routes for data collection, a more sophisticated and targeted method could have been used which would have significantly decreased the total amount of data taken. In this circumstance, data would have been collected exclusively near the travel zone boundaries.

### 4.5 Analysis Procedure

The goal of the experiment is to characterize the algorithm's accuracy. However, the performance of a supervised, machine learning algorithm such as KNN is inherently dependent on the quality of the training set. In this sense, the experiment evaluates not only the algorithm, but also the training set used.

The metric by which the algorithm accuracy is measured in this experiment is the percent accuracy of the algorithm as a function of distance from a boundary. For this experiment, percent accuracy is defined as the percent of correctly identified readings divided by the total number of readings. In considering distance from a boundary, the readings are discretized into groups based on how far they were taken from a boundary.

The data was analyzed using cross-validation (Bishop 2006), where the data is split into a

number of groups and then all but one group is combined to create the training set. The one remaining group is then analyzed as the test set. This constitutes a single fold of the validation. The process is then repeated until every group serves as the test set. For each fold, the algorithm is applied to each reading in the test set. The algorithm-determined travel zone was then compared to the actual travel zone as determined by the location of that reading. Figure 10 illustrates this procedure.

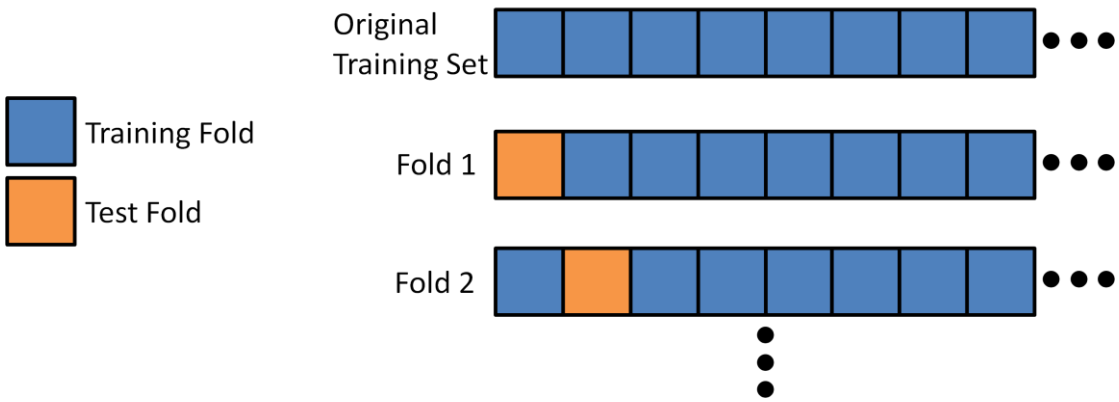


**Figure 10: Illustration of Multi-Fold Analysis**

The data was split into folds so that readings from a single data collection trip would stay together. This is done to avoid artificially inflating the accuracy of the algorithm under cross-validation. If while classifying a reading, additional readings taken from the same data collection trip were available in the training set, the algorithm would have a very high accuracy due to the similarity of the reading being classified and the training set readings. This similarity between such readings is a result of being observed under similar geographic, atmospheric, and temporal conditions. This would be an unfair advantage however, because under normal operation, the training set would never contain training data from the trip the vehicle was currently undertaking.

# Chapter 5: Algorithm Evaluation Results and Analysis

## 5.1 Introduction

The data used in this experiment was collected by four traffic control vehicles traveling in and around downtown Minneapolis. It was collected over the course of roughly 5 weeks comprised of 125 unique data collection trips, each corresponding to the data collected by one vehicle in one day. Over these trips, the hardware made 5.34 million readings and observed 531 unique CIDs. When stored, this data totaled a little over 3 GB.

The data was validated for the three test areas in downtown Minneapolis, as described in the previous chapter. In these analyses, only data in the test areas was considered. This means that for *both* training and test sets, the only readings considered were those originally observed in the test area. Again, the specific test areas used and their boundaries were arbitrary. The rationale for analyzing the data in smaller geographic test areas was that it would better capture the variability of the algorithm's accuracy over different portions of the travel zone's boundary.

For each test area, all readings taken in that area were subjected to the cross-validation process as described above. This process yielded an estimate of the system's accuracy as a function of distance from the boundary. These distances were discretized into 50 meter groupings.

The distance from the boundary for each reading was determined by finding the shortest distance, using simple geometry, between the position from which the reading was made

and the polygon or line segments representing the boundary. Calculating distance in this way yielded a measurement that did not consider the geometry of the local roads on which the readings were taken.

The analysis produced a 95% confidence interval on these accuracy estimates. This was accomplished with a simple binary output z-test. Because the accuracy estimates are reported in %, so are the confidence intervals, but it is important to note that these values refer to percentage points, not a percentage of the original accuracy.

The graphs that represent the results of this analysis show the percent accuracy as compared to the distance from the travel zone boundary. The confidence intervals as described above are represented in the figures below with error bars. Distances for which the accuracy and confidence interval is 0% (the lack of a bar) represent areas where there is no available data. They are included in the graphs in order to maintain x-axis consistency between graphs describing the same test areas. For reference, one city block in downtown Minneapolis is about 125 meters (410 feet).

### 5.2 Washington Ave

The results for the Washington Ave test area as it intersects the downtown travel zone boundary as shown in Figures 4 and 5 is summarized in Figure 11. Note that for the Washington Ave test area, positive distances correspond to positions outside of the downtown travel zone (east of the boundary) and conversely, negative distances

correspond to positions inside the downtown travel zone (west of the boundary). This is also the case for the alternate boundary positions.

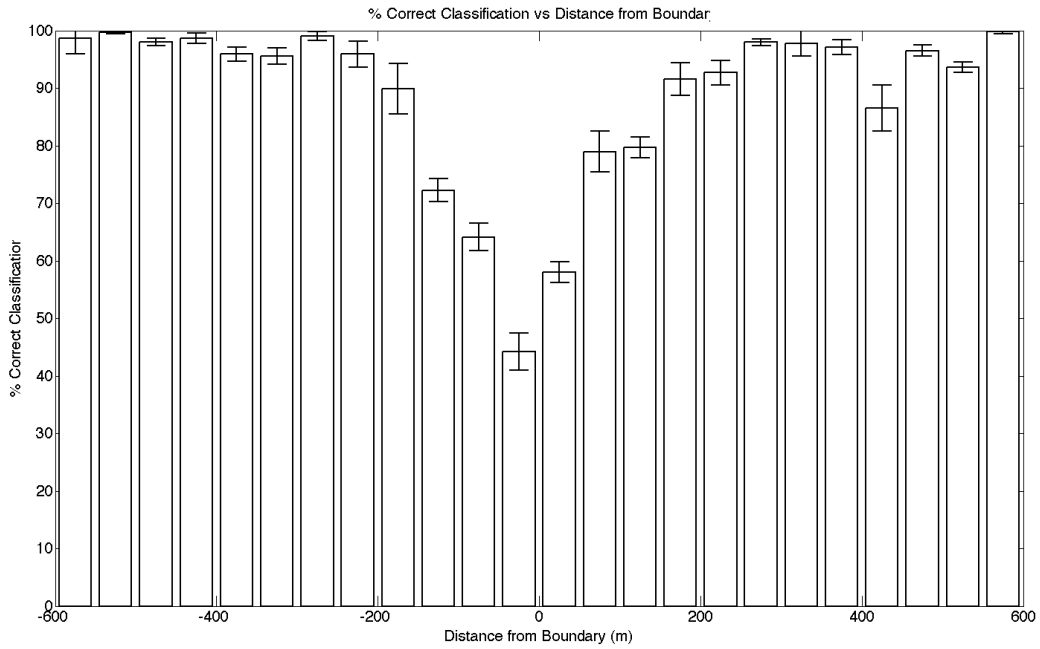


**Figure 11: Test Area 1 – Washington Ave Crossing Portland as Boundary**
East of boundary is positive.

The results from this test shows that to achieve a classification accuracy of roughly 95% in this test area and for this boundary, the vehicle must be at least 2 blocks, or 250m away from the boundary. It is also shown that the confidence intervals on these accuracy estimates are generally less than 5%.

It is also shown that occasionally, such as at the +400 to +450 meter distance away from the boundary, there are slight dips in the accuracy. This could be due to a number of factors. The most likely is that these dips are a result of random variation within the experiment and with additional data, the accuracies would be more smooth over a range

of distances. It is also possible that there is some physical feature in that area that negatively affects the modem's ability to receive accurate cell ID and strength information. These dips also manifest themselves in the other test areas, and it is assumed they do so for similar reasons.

The results for the Washington Ave test area as it intersects the alternate boundary at 4th Ave is shown in Figure 12.

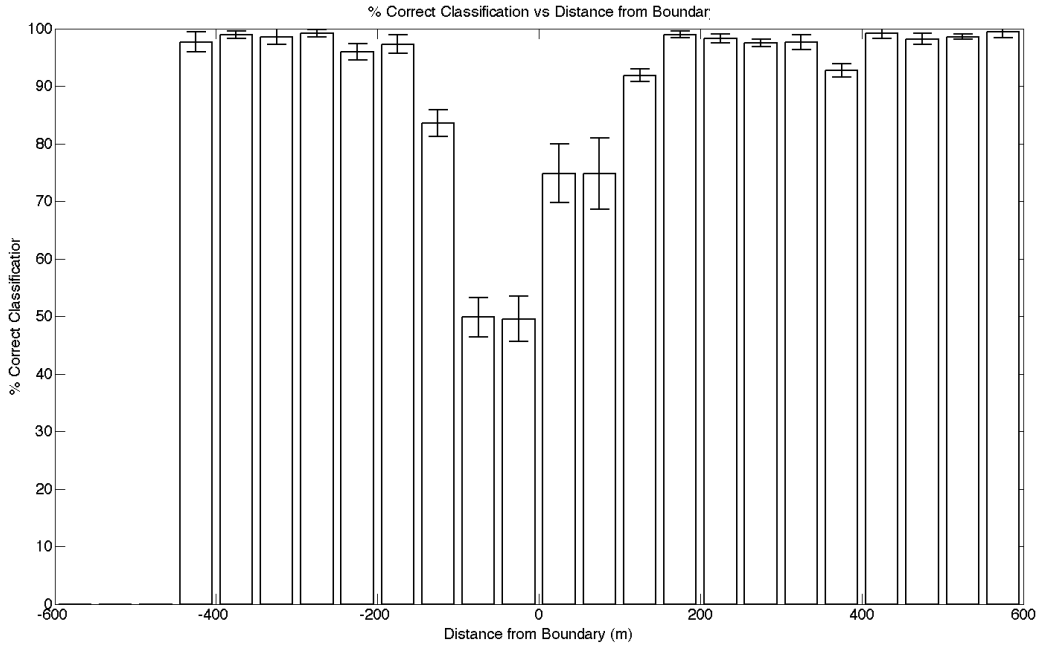


**Figure 12: Test Area 1 – Washington Ave Crossing Fourth as Boundary**
East of boundary is positive.

The results with this boundary are a little better than the original boundary at Portland Ave. Here, the same approximately 95% accuracy level is achieved at only about 150m, as opposed to 250m for the previous boundary.

The next plot, Figure 13, corresponds to the Chicago Ave boundary for the Washington

Ave test area.

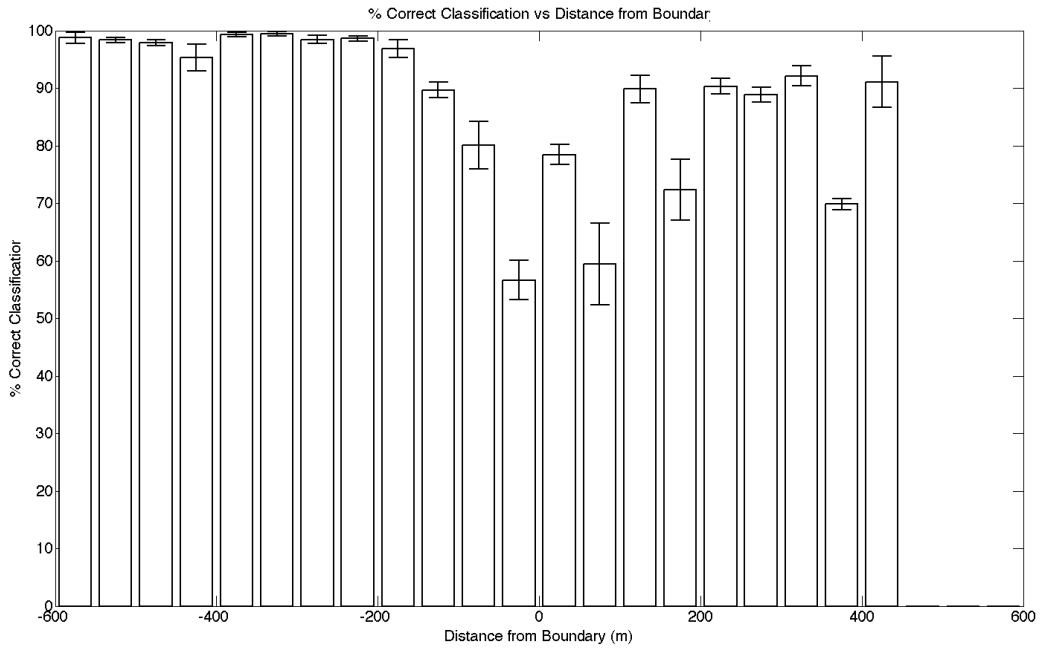


**Figure 13: Test Area 1 – Washington Ave Crossing Chicago as Boundary**
East of boundary is positive.

This boundary's results are noteworthy in a number of ways. The first is that it is not symmetric between the east (positive distances) and west (negative distances) sides of the boundary. The performance of the algorithm was generally worse when on the eastern side of the boundary. Again, this could be due to a number of possible factors. The most probable reason is the small amount of training data available for the east side of the boundary. As shown in Figure 6 above, for this boundary there are only 3.5 blocks between the boundary and the end of the test area from which training data can be used. If the test area was expanded, it's possible that this would improve the algorithm's performance.

### 5.3 First Ave

For the First Ave test area, positive distances correspond to positions south of the boundary and negative distances to positions north of the boundary. The results of the analysis performed in this area are shown below in Figure 14.

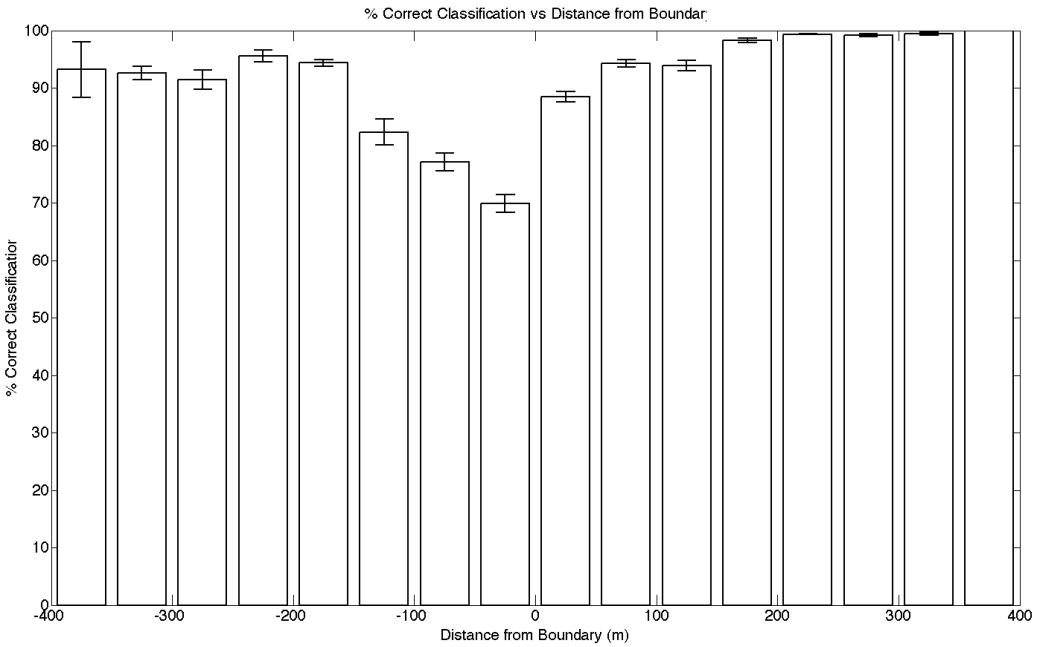


**Figure 14: Test Area 2 – First Ave Crossing Non-Street Boundary**
South of boundary is positive.

Similar to the Chicago Ave boundary for the Washington Ave test area, the results are not symmetric. Again, the relatively small test area may be the largest factor in explaining these results. However, looking at the positive (south) side of the boundary, the algorithm reaches 95% accuracy in about 50m from the boundary.

### 5.4 Eighth and Ninth Streets

For this test area, positive distances correspond to positions east of the boundary and negative distances to positions west of the boundary. The plots for these four intersections are shown in Figures 15 through 18 corresponding to intersections A through D as labeled in Figure 8.

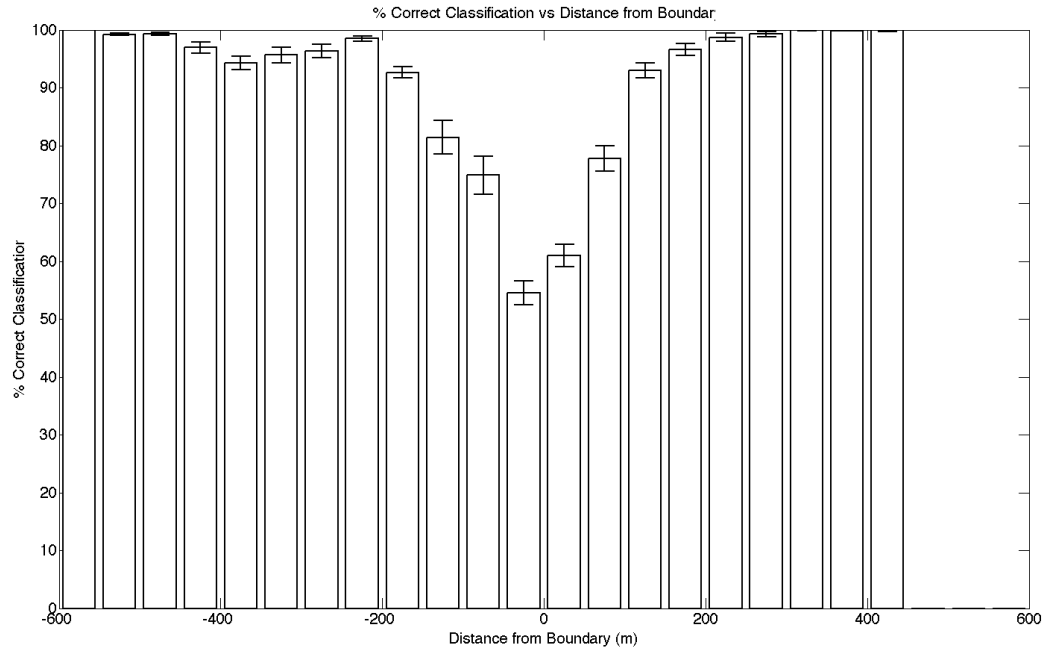


**Figure 15: Test Area 3 – 8th Crossing Marquette as Boundary (Intersection A)**
East of boundary is positive.

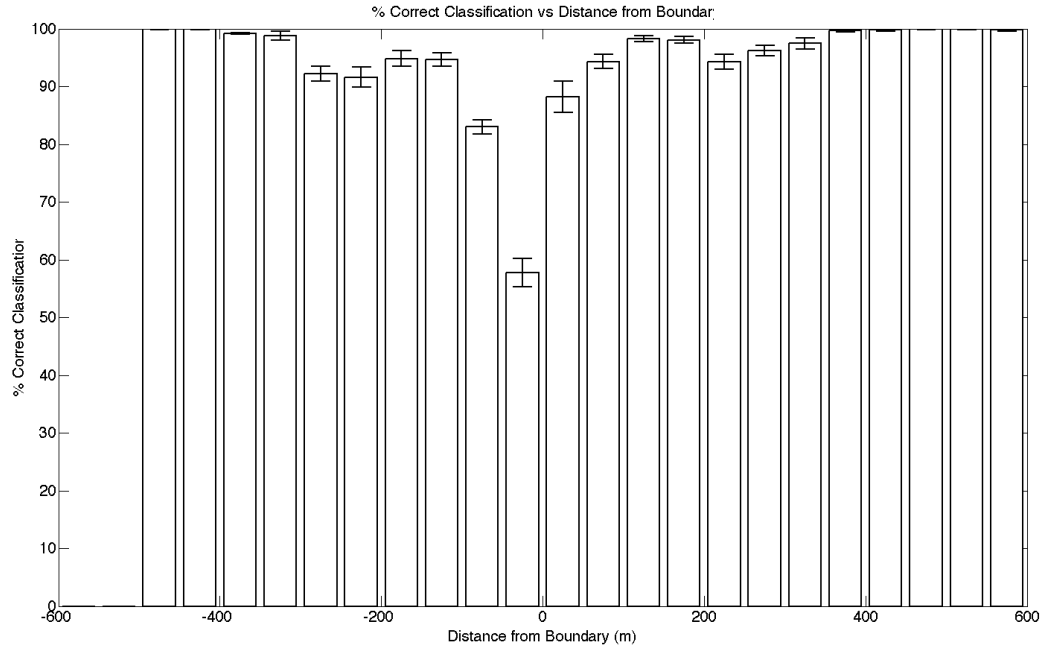


**Figure 16: Test Area 3 – 8th Crossing Nicollet as Boundary (Intersection B)**
East of boundary is positive.

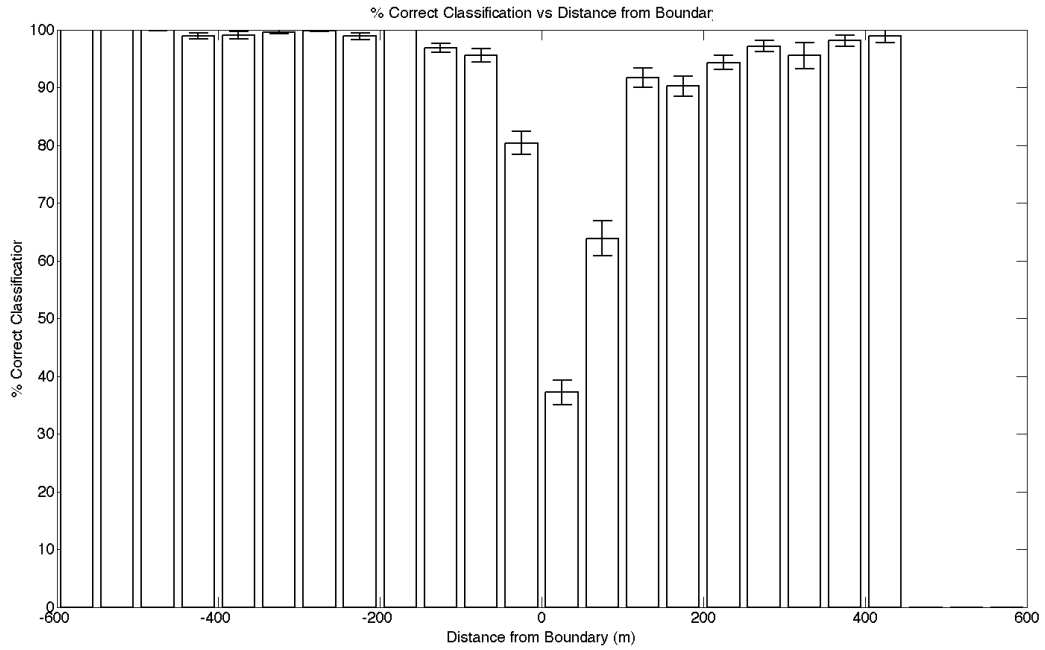


**Figure 17: Test Area 3 – 9th Crossing Marquette as Boundary (Intersection C)**
East of boundary is positive.

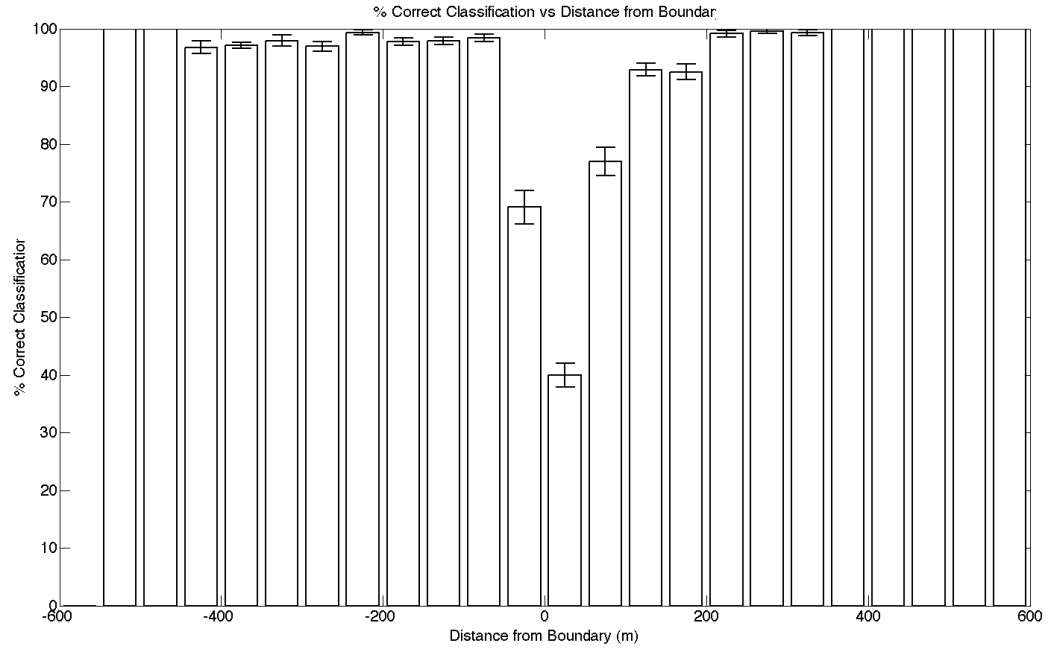


**Figure 18: Test Area 3 – 9th Crossing Nicollet as Boundary (Intersection D)**
East of boundary is positive.

The results show that in this test area, the accuracy reaches roughly 95% by the time the vehicle is 150 to 200m away from the boundary. Another important point is that over these four intersections, there is relative consistency between them. This indicates that although there is variation between nearby intersections, the results for a particular test area are relatively uniform. That said, some boundaries have better performance than others.

# Chapter 6: Conclusions

## 6.1 Discussion of Results

The results show that the accuracy with which the system determines the vehicle's current travel zone increases as the vehicle is further away from the travel zone boundary. This is an intuitive result because near the boundary, there are cell towers, or combinations thereof, that can be seen on both sides of the boundary. As the vehicle moves further away from the boundary, deeper within a travel zone, the cell towers encountered are generally never seen in any other zone. This uniqueness allows for near perfect classification.

It is important to note that as with any experiment, the results reported apply to the conditions of the experiment. In this case, the most significant condition was the size and quality of the training set. Generally, machine learning algorithms have better performance with relatively large training data sets. If better performance was required, it's possible that additional data collection could increase the algorithm's accuracy.

As initially predicted, the results show that the location of a zone boundary will affect the system accuracy. This result is particularly useful in the case where a travel zone boundary is flexible. That is to say that if the travel zone boundary or a portion thereof is not set, it can be placed to maximize the algorithm performance. This is clearly advantageous because then, travel zones can be determined in part by placing boundaries where exceptional performance is expected.

With any machine learning algorithm, it is expected that there will be some level of error in its classifications. Although an in-depth analysis is beyond the scope of this study, it is meaningful to consider methods for addressing and mitigating the risk associated with these errors. One method that we propose involves drawing a distinction between the legal or official travel zone boundary, identified by the policy makers and the computational boundary, from which the algorithm is trained. By decoupling these two boundaries, the computational boundary can then be placed inside or outside of the legal boundary. This would push the error prone, boundary ambiguity completely within or outside the travel zone's boundary. By doing so, the boundary positions can be selected so that the traveler pays the lower rate when there is ambiguity.

For example consider a boundary separating two travel zones with different associated prices shown below in Figure 19. Here, the computational boundary and the jurisdictional boundary coincide. The shaded box surrounding the boundary represents the geographic area in which the algorithm has difficulty accurately determining the vehicle's current travel zone.

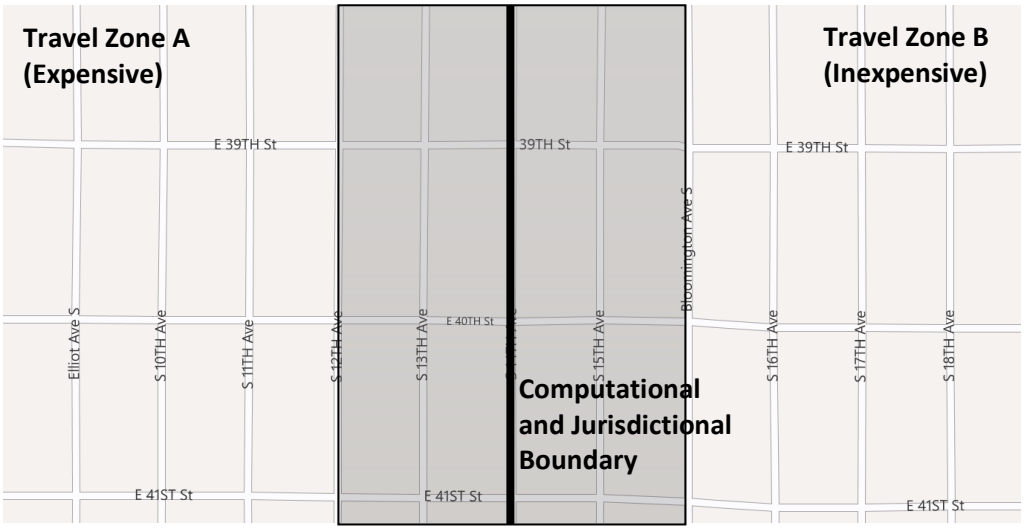


**Figure 19: Boundary Where Computational and Jurisdictional Boundaries Coincide**

Without moving the jurisdictional boundary, it is still possible to move the computational boundary which also moves the surrounding ambiguous area. Figure 20 shows this change.

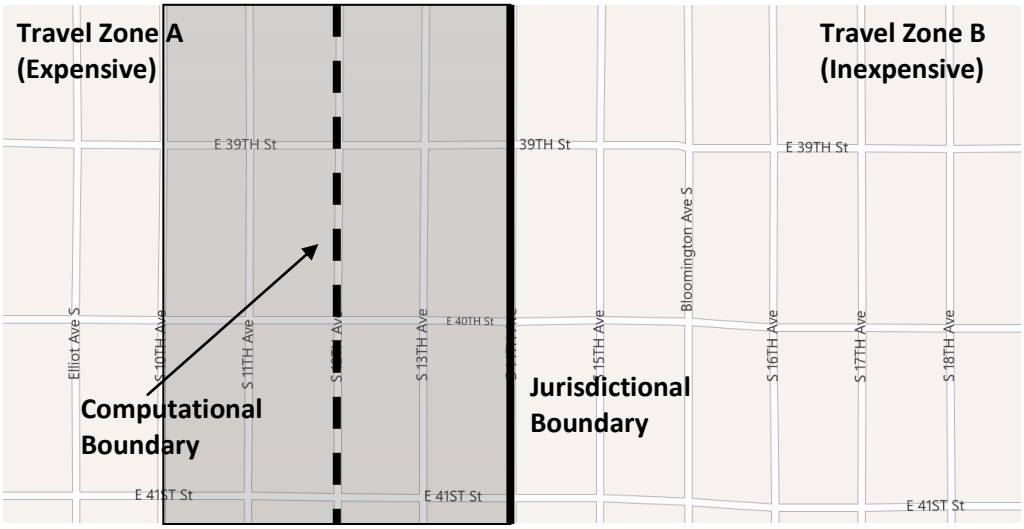


**Figure 20: Shifting Computational Boundary Away From Jurisdictional Boundary**

The ambiguity associated with determining the vehicle's current travel zone has been

placed within the expensive travel zone, as opposed to shared evenly between both travel zones. This means that it is very unlikely that the algorithm erroneously determines that a vehicle is in travel zone A, when it is really in travel zone B. When a vehicle is in travel zone A, but still in the ambiguous area near the boundary, it is possible for the algorithm to determine that the vehicle is in either travel zone. If the algorithm correctly determines the vehicle's current travel zone, that is clearly acceptable. If it does not classify the vehicle's current travel zone correctly, the vehicle will be charged the rate for travel zone B, which is less expensive and advantageous for the driver.

This method addresses ambiguity in a way that is advantageous for the driver, but may not be fair for the jurisdiction collecting the higher road user fee. If required, this method could be set up in reverse so that an ambiguity would result in the higher road user fee which would favor those collecting the revenue.

### 6.2 Future Work

The results of the experiment described here leave room for future work to both expand the results reported in this document as well as to complete tasks that were outside the scope of this project. First, the preliminary experiment performed on the accuracy of VMT obtained by integrating speed from the OBD-II port indicates that this method provides a suitable source of data from which VMT can be calculated. Future experiments could test a wider variety of vehicles under additional driving conditions and run more trials for each combination of factors. This would better characterize the

accuracy of the calculated VMT under different environments and with more measurements, could also obtain a better estimate of the method's accuracy.

The KNN algorithm used in the travel zone determination is conceptually simple, yet is computationally expensive for a training set as extensive as the one used in this project. The goal of this project was to evaluate the algorithm's performance in terms of accuracy, but not efficiency. This computation cost could be reduced or otherwise mitigated in a number of ways not addressed in this project. Currently, the training data is organized into a single text file that is effectively unorganized. Large improvements can be made to the efficiency with which the data is stored by utilizing sophisticated data organization techniques. One such method would be to utilize a k-d tree, which is a way to organize a training set of data to facilitate usage with a KNN algorithm. It is also likely that the implementation of the KNN algorithm can also be improved to increase computational efficiency.

Additional work could better characterize the algorithm's performance for varying amounts of training data. It is possible that there is an optimal amount of training data after which, there are diminishing returns on using additional data. Furthermore, there may also be an optimal data collection method. For a given amount of training data, there may be a particular data collection driving pattern that yields the best overall system accuracy. Finally, when considering methods through which training data needs are reduced, it is important to ensure that there is never a situation where a vehicle can pass through a boundary undetected by the system. This is of particular interest in areas with poor cellular coverage because it is possible that a portion of a boundary may pass

through a region with no cellular coverage. Future analyses should identify situations where there could be holes in the boundary, how this affects system accuracy, and determine methods to mitigate these issues.

The algorithm that determines the vehicle's travel zone was tested relatively thoroughly in the Minneapolis downtown area. Future studies could greatly expand the number and types of geographic regions in which the system is evaluated. For example, these could include rural areas with little or no cellular coverage, typical of less populated areas such as deserts or prairies, areas with unique terrain such as mountains or large bodies of water, or dense urban areas such as Manhattan or Los Angles whose large CBDs may require multiple travel zones.

In addition to improving on the findings of the experiments, future work in this area needs to consider additional items that were outside the scope of this project. These tasks, which are not necessarily specific to the methods proposed here, include the development of a back office system capable of aggregating MBUF information, assessing these fees to drivers, and then distributing this revenue to the correct jurisdictions. Additional considerations must be made to determine how much of the computational process occurs on the vehicle's hardware and how much occurs in the back office, as well as how the two communicate with each other. In implementing a real-world system, it is insufficient to address technical considerations alone. In order to make informed decisions about how an MBUF system functions, collaboration must occur between those creating the system, and the policy makers specifying the system's requirements.

### 6.3 Summary

The goal of this experiment was to evaluate the quality of a system capable of determining a vehicle's current travel zone through cellular assignment. This was accomplished by creating a training set of GPS-provided location and cellular modem readings throughout Downtown Minneapolis. Using this data, a multi-fold validation scheme was used to determine the accuracy of the machine learning algorithm used to classify readings into one travel zone or another. This validation was performed for three test areas in the downtown area. The metric used to examine accuracy of the algorithm was the percent of correctly classified readings versus distance from the zone boundary.

In a broad sense, this experiment confirms that is it possible to create a system capable of communicating with cellular towers to the degree that it can determine their unique CIDs as well as an indication of the tower's signal strength. Furthermore, this information can be saved and processed so it can serve as a training set for a machine learning algorithm capable of determining the vehicle's current travel zone. Additionally, the findings quantify the accuracy with which this can be done.

The results of the experiment show that the algorithm can correctly detect when a vehicle crosses a boundary with roughly 95% accuracy when the vehicle is at least 200m, or a little less than 2 city blocks, away from the boundary. The accuracies are even higher the further one is away from the boundary. This implies that a cellular based VMT system may be a feasible method to aggregate VMT by predetermined geographic zones,

especially if policymakers charge the lower rate in the ambiguous region between travel zones.

# References


1. Donath, M., A. Gorjestani, C. Shankwitz, R. Hoglund, E. Arpin V, P.M. Cheng, A. Menon, and B. Newstrom. **Technology Enabling Near-Term Nationwide Implementation of Distance Based Road User Fees.** Publication CTS 09-20, Center for Transportation Studies, University of Minnesota, June 2009. Available at http://www.its.umn.edu/Publications/ResearchReports/reportdetail.html?id=1790

2. Larson, R. and K Sasanuma. **Urban Vehicle Congestion Pricing: A Review.** Journal of Industrial and Systems Engineering. Vol. 3, No. 4, pp 227-242. Winter 2010.

3. MBUF Policy Task Force. **Report of Minnesota's Mileage-Based User Fee Policy Task Force.** MnDOT. December 2011. Available at http://www.dot.state.mn.us/mileagebaseduserfee.

4. Douma, F. and S. Aue. **ITS and Locational Privacy: Suggestions for Peaceful Coexistence.** *Journal of Transportation Law, Logistics and Policy*. Vol. 78, Num. 2, 2011, pp 89-108.

5. Chu, J. **Technologies That Enable Congestion Pricing: A Primer.** Publication FHWA-HOP-08-042, FHWA, U.S. Department of Transportation, October 2008.

6. Sorensen, P., L. Ecola, M. Wachs, M. Donath, L. Munnich, and B. Serian. **Implementable Strategies for Shifting to Direct Usage-Based Charges for Transportation Funding.** National Cooperative Highway Research Program Web-Only Document 143. June 2009. http://www.trb.org/Publications/Blurbs/162252.aspx.

7. Sorensen, P., M. Wachs, and L. Ecola. **System Trials to Demonstrate Mileage-Based Road Use Charges.** National Cooperative Highway Research Program Web-Only Document 161. October 2010. Available at http://www.trb.org/Publications/Blurbs/164521.aspx.

8. Whitty, J. **Oregon's Mileage Fee Concept and Road User Fee Pilot Program: Final Report.** Oregon Department of Transportation, November 2007. Available at: http://www.oregon.gov/ODOT/HWY/RUFPP/docs/RUFPP_finalreport.pdf

9. Broaddus, A. and C. Gertz. **Tolling Heavy Goods Vehicles: Overview of European Practice and Lessons from German Experience.** Transportation Research Record:

Journal of the Transportation Research Board. Vol. 2066, 2008, pp 106-113.

10. TransCore. **ROVR™ Products - Realtime Onboard Vehicle Reporting Unit – TransCore.** Accessed January 2012. http://www.transcore.com/products/ROVR-products.shtml

11. Bishop, C. M. **Pattern Recognition and Machine Learning.** Springer, New York, 2006.

# Appendix A: Analysis of Calculating VMT Through OBD-II

## A.1 Error definition and notes

For this experiment, error was defined based on the following equation:

$$error = \frac{odometer - calculated}{odometer}$$

The motivation for considering error in this form as opposed to a simple difference was to normalize the difference over the path lengths. This would allow for a more meaningful comparison between the city and highway paths.

It is noteworthy that as it is defined, the sign on the error is meaningful. Positive errors correspond to the situation where the calculated VMT is less than the odometer provided distance. Negative errors correspond to situation where the calculated VMT is greater than the odometer reading. Therefore, a non-zero mean of errors over the trips, as was the case in the experimental results, indicates a bias in the system.

## A.2 Expanded analysis methodology

To reiterate from the body of the document, the four factors that were considered were integration technique, vehicle type, driving conditions, and tire pressure. Treatments consisted of a combination of these four factors and were applied to a single trip, which serves as the experimental unit. Due to constraints on time and resources, each treatment

was applied to a single trip. Because it would be difficult to continually switch between vehicles and driving conditions, the treatments were randomized in a split-plot design. This is an experimental design for situations where the order in which the treatments are applied can't completely be randomized. This design addresses that issue by accounting for potential temporal effects in the analysis. More information can be found in an experimental design textbook, for example (Bishop 2008).

There were four whole plot units, each of which were a collection of three individual trips to which a vehicle and driving condition was applied. This means that on the whole plot level, the experiment consists of two factors (vehicle and driving conditions) that are crossed and completely randomized. Then within each collection, the three trips were randomly assigned a tire pressure level. Finally, to every experimental unit, both levels of the integration technique factor were applied.

In the process of the analysis, it was determined, by inspection, that the integration technique had no effect on the error. For this reason, only a single integration technique was considered, which reduced the experiment to the remaining three factors.

This "new" design is still analyzed as a split-plot design. The whole plot units are collections of trips and to each collection, a combination of vehicle and driving condition is applied. To reiterate, vehicle is considered to be a blocking factor. Then, to each of the three trips within a collection, each of the three tire pressures is applied.

The statistical package R is used to assist in the analysis. First, an lme model is fit considering vehicle type, as a blocking factor, tire pressure and driving conditions, as

well as the whole plot error term. An ANOVA table for this model is shown in Table A1.

**Table A1: ANOVA table for statistical model**

| | numDF | denDF | F-value | p-value |
|---|---|---|---|---|
| (Intercept) | 1 | 4 | 331.9728 | 0.0001 |
| Vehicle | 1 | 1 | 5.3359 | 0.2601 |
| condition | 1 | 1 | 40.2671 | 0.0995 |
| pressure | 2 | 4 | 1.2518 | 0.3783 |
| condition:pressure | 2 | 4 | 1.2454 | 0.3798 |

The ANOVA table indicates that none of the factors or factor interactions were significant.

Testing for normality and constant variance is important in order to confirm assumptions made in the analysis. In this case with such little data, these tests were inconclusive.

### A.3 Experimental Apparatus

Vehicles: Buick LeSabre, Chevrolet Impala

OBD-II reader: Elm 327

# Appendix B: Hardware and Software Used in This Project

## B.1 Introduction

This appendix describes the hardware platforms utilized throughout the course of this project. The software used on each of these platforms is also described, but only to the extent needed to better compare and contrast the platforms themselves. This does not include a discussion of the travel zone determination algorithm which is addressed in further detail in Appendix C. Additionally, because the bulk of the algorithm analysis was performed offline in MATLAB, this section will focus on the hardware platforms in the context of their role within the data collection phase of the project, as opposed to discussing the hardware that would be used in a deployable MBUF system.

## B.2 Android Development Phone 1

The first goals of the project were to confirm that cell towers' unique cell IDs were in fact accessible by a cellular modem. Additionally, it was important to determine what other tower information could be accessed. Later, this information would need to be logged and paired with location data as provided by GPS. To accomplish this, the Android Development Phone 1 was used. This device is shown below in Figure B1.


**Figure B1: Android Development Phone 1 (ADP1)**

The Android Development Phone 1 (ADP1) is a carrier unlocked, GSM cellular phone made by HTC (physically, it is identical to the HTC Dream/G1). The ADP1 was the first available development hardware for the Android operating system, which at the beginning of the project was relatively new. The reason this hardware was chosen was because it provided a portable and self-contained hardware platform that contained a cellular modem, a WIFI radio, and a GPS receiver.

The ADP1 was chosen over the iPhone by Apple, due mainly to pricing considerations. An iPhone would have required a costly 2 year contract for data and voice. The ADP1 however, did not require such a contract. Data and voice plans could be purchased separately and could be on a month-to-month basis with no minimum commitment. Additionally, pay-as-you-go plans could be used on the ADP1 which were also

unavailable on the iPhone. Lastly, the ADP1 would work with any GSM carrier (in the US this means AT&T and T-Mobile), and the iPhone could only work with AT&T. Note that these conditions were the case when the project was started in the early summer of 2009, and are not the case now as of Spring 2012.

The Android platform had a number of strong points. The biggest was the amount of available documentation, both official and unofficial, on the Internet. This drastically reduced the learning period when first starting with the ADP1. Additionally, the platform was developer friendly. That is to say, there were a number of very useful application programming interfaces (APIs). These provided easy methods by which to make calls and queries to the phone's various hardware components (e.g. GPS, WIFI, cellular modem, etc.). Lastly, because the platform was intended for consumer use, there were easy to use tools to create a graphical user interface.

The Android platform and the ADP1 hardware itself also had some limitations. The first was that although there were useful APIs to utilize in programming, all hardware calls had to be made through them. The platform would not allow low-level programmatic access to the hardware, which in some cases was problematic. The other main issue was with the physical robustness of the ADP1. The phone was made of plastic and screen was exposed. Under most conditions, this would be acceptable, but if a number of ADP1s were deployed in vehicles for a larger data collection effort, they might prove too fragile.

### B.3 Multitech rCell

As the project progressed, a new hardware platform was discovered that better fit the needs of the project. The rCell by Multitech is a cellular network-based router for industrial and commercial applications. The hardware features an ARM processor, a cellular modem, as well as both Ethernet and an RS-232 serial communication port. The device is housed in a relatively robust metal case. Unlike the ADP1, the rCell does not have a battery and requires external power from a cigarette lighter plug or standard wall power. Additionally, it did not include a GPS receiver, so in order to obtain location information, an external Garmin GPS receiver was attached through the serial port. The GPS receiver also required external power from a cigarette lighter plug. Because of this the rCell wasn't as self-contained as the ADP1 and when using it, it was necessary to have multiple cords running through the vehicle cabin.

The stock operating system on the rCell was replaced with a version of the Ångström Linux distribution. This allowed for programs to be written in the C programming language that made low level system calls to the various hardware components. Specifically, this allowed communication with the modem through the Hayes command set (also sometimes called AT commands). Because an unofficial operating system was being used, there was little to no support, and documentation was scarce compared to the Android platform. The other drawback to using the rCell this way was that there were stability issues associated with the non-stock operating system and the custom software being run on the device.

### B.4 Multitech Cellular Development Platform

The current hardware being used is the Multitech Cellular Development Platform (CDP). The CDP is essentially an upgraded version of the rCell with the operating system modification as described above. This allows the hardware to run the same software as the rCell. The difference is that with the CDP, this operating system comes stock, which eliminates or at least mitigates the stability, support, and documentation issues with the rCell. The other big difference is that this hardware includes a built-in GPS receiver, which shares a remote, wired antenna with the cellular modem. Additionally, there is an SD card slot that allows for easier data transfer during data collection.

# Appendix C: Considered Travel Zone Determination Algorithms

## C.1 Introduction

This appendix documents the four travel zone determination algorithms that were implemented and evaluated prior to the KNN algorithm currently in use. This includes details on the methods as well as a discussion of the motivation for choosing and then ultimately deciding against their use.

## C.2 Simple Cell Identification

The first algorithm considered was the original method described at length in (Donath et al. 2009), where a vehicle's current travel zone was determined by looking up the modem's current cell in a previously created table of known cell IDs. The basis for this method is the assumption that each cell tower's cell, or the geographic area over which a given cell tower can be received, is small enough that simply knowing what cell a vehicle is currently in is sufficient to determine the current travel zone. Then, data can be acquired that pairs these cell IDs with location information such that the result is a table that matches each cell ID with to single travel zone. Then in operation, when a modem determines its current cell, it uses the table to also determine its current travel zone.

The main advantage of this method is that it is very simple in that it requires very little computation and data storage. It also requires very little data collection, as each cell ID needs to be observed only long enough to be assigned to a travel zone. However, in

examining this method further, significant issues were identified. The primary issue was that this algorithm provided neither an acceptable precision nor accuracy in determining a vehicle's current travel zone. This was due to two main factors. The first was that the size of a single cell could be very large, covering area on both sides of a boundary. The other factor was due to the way in which a cellular modem determines its current cell. When a cell modem moves through space it generally chooses to associate with the cell tower with the highest reception strength. However, this isn't always the case depending on network factors such as usage and load. Additionally, the coverage boundary of a given cell might change in geographic area depending on external factors such as time, weather, and nearby electrical interference. This means for multiple trips across a boundary, the same cell tower's cell might not correspond to the same geographic area.

Although this method proved inadequate for travel zone determination when near a boundary, for cases where the vehicle is far away from a boundary, it is still a suitable method to ensure that the vehicle is still in the expected travel zone. Because of this, this method was incorporated into our travel zone determination algorithm as a pre-processing step. Cell IDs that are only seen in a single travel zone can be recorded and used to determine the current travel zone. However, if no such cell IDs were observed in a given modem reading, then a more complex algorithm would need to be used.

### C.3 Boundary Based Linear Discriminant

The next method considered attempted to address the major limitations with the simple

cell ID method described above. It did so by utilizing additional information other than just the modem's current cell ID. This algorithm would be based on all of the current observable cell IDs and their received signal strength indication (RSSI) values. These values would then be used with a linear discriminant that would determine the vehicle's most likely current travel zone.

The linear discriminant method is a parametric learning algorithm used to evaluate a sample (in this case a reading consisting of a collection of cell IDs and the associated RSSIs) and then determine the population (in this case the collection of readings associated with a particular travel zone) from which the sample most likely originated. This method is parametric in that it characterizes these populations with a number of statistical values. The values used in this application were the set of RSSI means for each population and a single covariance matrix for all populations. This allowed for each population to be described with a relatively small set of parameters (i.e. the means and covariances) instead of requiring all the training data. Then, to determine the most likely source of each reading, the normalized (by the covariance) distance between the reading and each population is computed. The smallest distance corresponds to the most likely source population.

This raises the non-trivial question of how to select the populations. In order for this algorithm to work, a number of assumptions must be true. The first is that each population must be uniform. This means that a travel zone would not make a good population in this context because depending on its geographical size, the characteristics of the travel zone could vary drastically at one end of the zone, compared to the other

end.

Instead of using travel zones as a source of the populations, smaller boundary zones were used instead. A boundary zone was defined to be a geographically small area that was on the boundary between travel zones. Each boundary zone is also a member of a travel zone. The important difference was that within these smaller boundary zones, there was better uniformity among readings taken within. This idea is illustrated in Figure C1. The left half of the figure shows two travel zones (red and blue) separated by an ambiguous boundary area around the nominal boundary between the zones. The right half shows a boundary defined by boundary zones, each of which can assumed to have uniform characteristics insofar as their use in a linear discriminant.

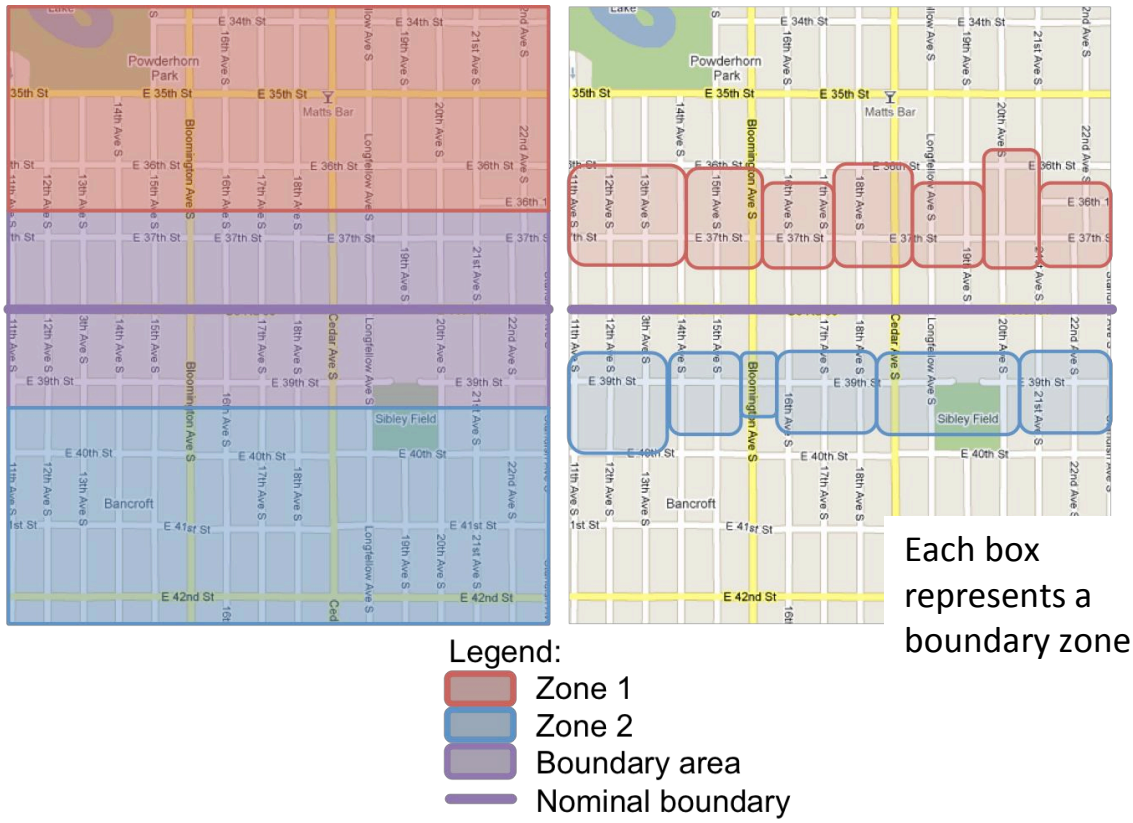


**Figure C1: Illustration of Boundary Based Zone Determination Algorithm**

Under normal operation, the algorithm continually evaluates the current readings' normalized distance from each of the boundary zones and monitors changes in the current boundary zone for changes in the travel zone.

This algorithm also proved to be unsuitable as the precision and accuracy was too low for the intended usage. The fundamental issue was that the use of this algorithm required an underlying normal distribution for the RSSI values, which was not the case. This means that the RSSI values for a given cell ID needed to be normally distributed about the mean for that cell ID in a given boundary zone. Due to the nature of the data, this was not the case.

### C.4 Regression Analysis Based Method

The regression analysis based method was a less sophisticated attempt to capture patterns in observable cell IDs and their corresponding RSSIs and identify differences between the two sides of a boundary without using a formal machine learning algorithm. Although it would likely be possible to automate the process, or a sufficiently similar process, later on once we began to use a more formal machine learning algorithm, this method was dismissed before it got to that stage.

This method is best explained with an example. Consider Chicago Ave in Minneapolis, MN as it crosses a travel zone boundary that coincides with I-94 just south of downtown. This stretch of road is illustrated in Figure C2 showing the zone boundary in purple and the path along Chicago in black.

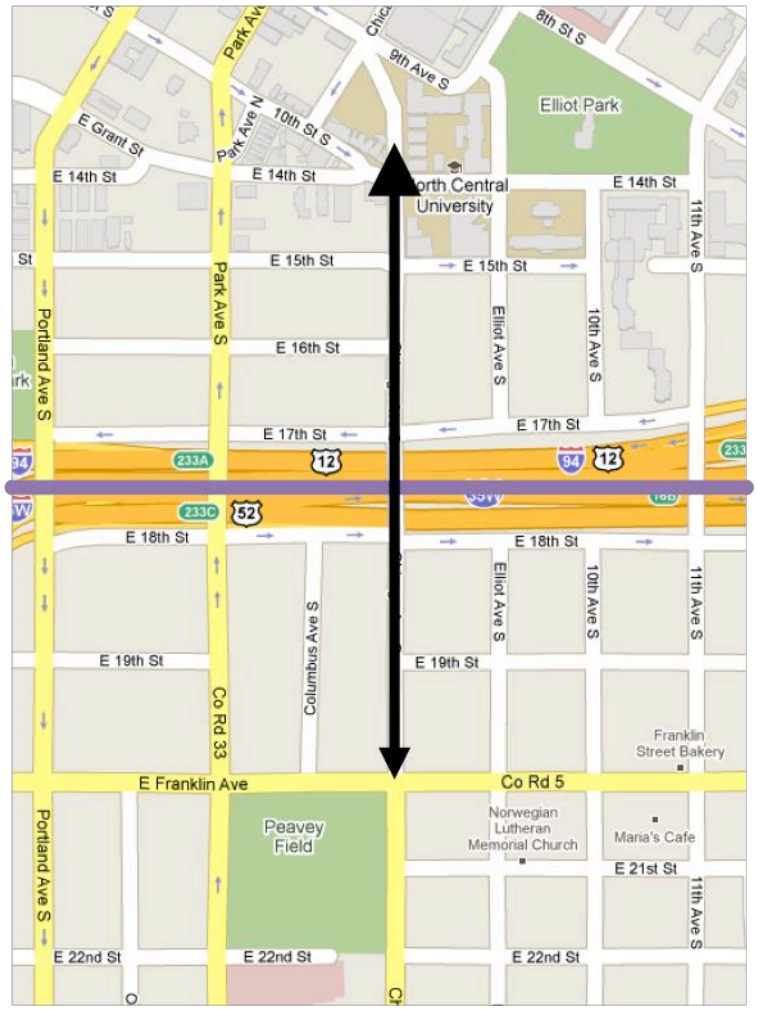


**Figure C2: Chicago Ave in Minneapolis Crossing a Boundary Coinciding with I-94**

Next, data is collected along this street matching the cell IDs visible as well as their corresponding RSSIs with the location in which they were observed. This data is represented over multiple trips in Figure C3. The x-axis shows 1-dimensional distance along Chicago Ave in meters, running north-south (or more specifically, in the MN State Plane South northing or y-direction). The boundary is marked with a purple vertical line. Grey vertical lines mark major cross streets along the stretch of Chicago Ave under consideration. The y-axis corresponds to the RSSI with which a given cell ID is received.

Each unique marker (color-shape combination) represents a single cell ID. Note that the same cell ID may be observed in many locations at many different RSSI values, and additionally, at a given location, a single cell ID may be observed at different RSSI values.

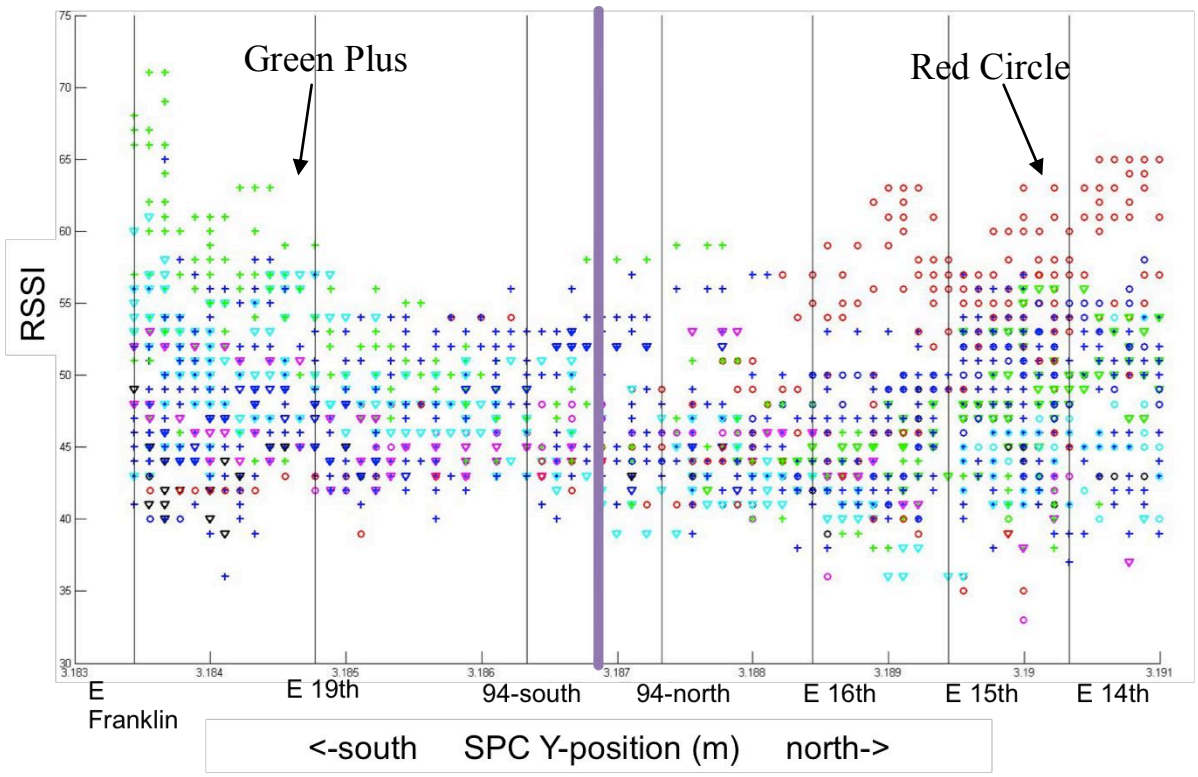


**Figure C3: RSSI Values Versus Position along Chicago Ave by Cell ID**

Returning to the problem at hand, this data must be used somehow to make a determination about which side of the boundary a vehicle is traveling based on patterns of which cell IDs are observed and at what RSSI levels. With this in mind, a simple pattern that could be used is that when the cell ID corresponding to a “red circle” is observed above an RSSI of 55, the vehicle is on the north side of the boundary. Another similar pattern could be that when a “green plus” is observed at an RSSI of 50 or higher, the

vehicle is on the south side of the boundary.

The creation of simple, human-observable patterns is the basis of this method. The advantage of this method was that it was computationally simple to execute this algorithm on the data collection hardware. Once created, rule sets could yield relatively high accuracy on a given boundary. However, it was time consuming to analyze the data and determine the set of rules required to create the algorithm. Additionally, once a rule set was created, it was only good for the single street it was created for. For a boundary that was crossed by multiple streets, there was no way to determine the street whose rule set should be applied to a given situation. As rule sets were expanded to include more streets crossing a boundary, suitable rules were less apparent. For example, consider the illustration shown Figure C4.

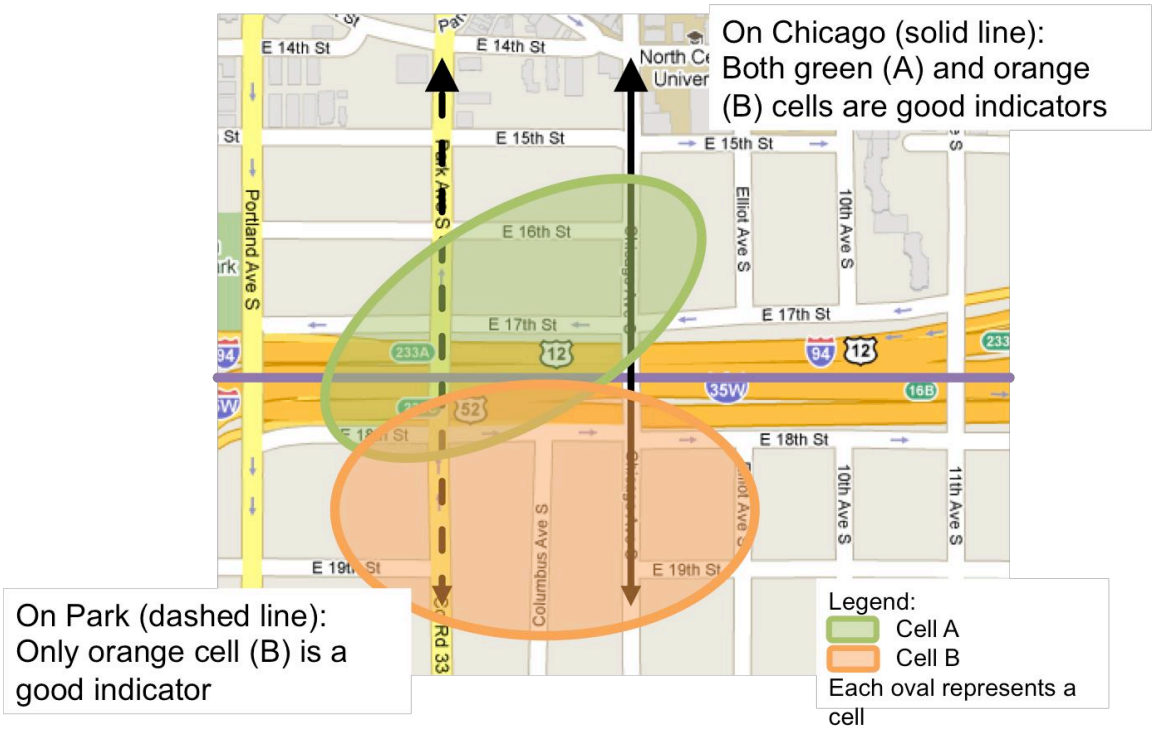


**Figure C4: Illustration of Pitfalls with the Regression Analysis Based Method**

Here, cells A and B are both good indicators on Chicago Ave. They don't overlap and seeing one cell tower or the other clearly indicates that the vehicle is on one side or the other of the boundary. However, along Park Ave, Cell A can be seen on both sides of the boundary, so it is not a good indicator. This means that without prior knowledge about which street the vehicle is using to cross the boundary, Cell A can't be used to determine the travel zone. Similarly, as more cross streets need to be considered, fewer and fewer cells are useful.

### C.5 Modified Naïve Bayesian Classifier

The next travel zone determination algorithm considered was a modified naïve Bayesian classifier. This method is based on Bayes' theorem, which states

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

This rule can be extended to the case at hand where one wants to determine the probability of a vehicle being a member of the $i^{\text{th}}$ travel zone (travel zone $i$) given the current modem reading of cell IDs and RSSI values ($\vec{x}$). This extension yields

$$P(zone_i|\vec{x}) = \frac{P(zone_i)\,P(\vec{x}|zone_i)}{P(\vec{x})}$$

This equation describes the probability of being in a particular zone given the current

reading as being equal to the probability of being in that given zone multiplied by the probability of seeing that same reading given the modem is known to be in that given zone, divided by the probability of seeing that reading anywhere.

This algorithm then diverges from the rigorous multivariate representation of Bayes' theorem and considers each currently observable cell ID separately. In this variation, each of the currently observable cell IDs contribute a vote, or a portion thereof, towards travel zones based on how frequently they are observed in those zones compared to other cell IDs observed in those zones. Then, these votes are summed and the travel zone with the highest vote count is determined to be the vehicle's current travel zone.

The advantages of this method are that it allows for non-uniform data collection between travel zones and it is a non-parametric algorithm that does not require any assumption about the underlying distribution. Unfortunately, this method did not perform well in tests and was discarded. Upon further examination, it was determined that the main factor contributing to this performance was that the algorithm still made an assumption about uniform data collection within travel zones, and this could not be ensured. Additionally, attempts to mitigate this and consider non-uniform data collection proved complex and ultimately did not increase the algorithm's performance.

### C.6 K-Nearest Neighbors

The current travel zone determination algorithm is an application of the k-nearest

neighbors algorithm. Initially, the algorithms considered were designed to discard all the training data in favor of retaining either a set of statistical parameters, or some other form of consolidation. This was based on the assumption that keeping and using the entire training set would require prohibitively too much data storage. However, on further examination, it was ultimately determined that this wasn't necessarily the case. Additionally, due to the nature of the data, methods that considered the entire training set would ultimately be necessary to yield results with adequate performance. The details of how the algorithm works are discussed in more detail in the body of this document.