Real-Time Traffic Prediction Improvement through Semantic Mining of Social Networks

Scott Grosenick

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computing and Software Systems

University of Washington

2012

Committee:

William Erdly

Mark Kochanski

Michael Stiber

Program Authorized to Offer Degree:

Computing and Software Systems

Abstract

Real-Time Traffic Prediction Improvement through Semantic Mining of Social Networks

Scott Grosenick

Chair of the Supervisory Committee:
Associate Professor William Erdly, Ph.D.
Computing and Software Systems

Many years of research have yielded computer modeling techniques that can predict the behavior of complex systems, such as traffic speeds in regional transportation systems, with high accuracy. However, the prediction accuracy suffers significantly when non-recurring events, such as traffic accidents, occur in these systems. Yet the impacts of such disruptions are *precisely* the events that vehicle operators need to be aware of when planning their trips. Techniques for autonomously detecting these events, such as automated incident detection from traffic flow data and computer vision, are active fields of research but currently offer significantly less accurate data than actual human observations. Therefore, introducing novel ways to identify and quantify disruptions using human input can improve modeling accuracy when speeds are disrupted, while raising new topics for research to address this large, unmet need. Blending human-relayed incident detection mined from social networks with existing traffic modeling techniques provides a promising new direction for improving accuracy in traffic speed prediction.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This research would not have been possible without the guidance and support of many people. While I am including the abbreviated list here, there are a whole cast of characters that contributed to this work in big and small ways. Thank you all!

I want to thank Charlotte, my wonderful wife, for bearing with me through the past few months. Somehow, despite only seeing the back of my head silhouetted against the glow of my computer screen, she still remembers me and enjoys my company.

Thanks to the rest of my family for their support and confidence. It helped me through the few low points and many long hauls. You were instrumental in keeping me on track.

I want to express my gratitude to all of the hard-working people of the CSS program. You are enabling people to realize their full potential every day that you come to work.

Thanks to my committee for helping to refine and shape the focus of this research over the weeks and months that we worked together. I never expected to see you smiling in my presence after plunking down a three hundred page document for you to critique.

Thanks to Sally Solaro for proofreading and providing feedback on my early drafts. It was great to hear that this work was approachable and interesting to non-computer science folks.

Thanks to Peter Dodd for reminding me that he and my sailing crew were the wind in my sails. (No thanks for phrasing it that way. ☺) The thought of burying the rails with everyone kept me driving forward through many, many late nights.

Finally, thanks to David Schomer – without his opus of caffeinated goodness served up at Espresso Vivace, I wouldn't be more than half way through my first draft.

Introduction

Many years of research have yielded computer modeling techniques that can predict the behavior of complex systems, such as traffic speeds in regional transportation systems, with high accuracy. The two most common prediction techniques are: travel time prediction along a route, which predicts the time taken to traverse a given sequence of traffic segments, and speed prediction on a single road segment, which is then stitched together with other predictions to provide an overview of traffic speeds in a region. Both methods are key components in a comprehensive traffic prediction system. However, this paper focuses on the prediction of traffic speeds on a single segment, as it applies to all vehicles that utilize the segment, without respect for their origin, destination, nor the vehicle's route between them. Some practices and methods used in route prediction are valid for both techniques and are leveraged in this experiment where appropriate.

The accuracy of both modeling techniques decreases significantly when non-recurring events, such as traffic accidents, occur in the systems they attempt to model. In order for a model to consistently predict conditions with high accuracy, it must include data about these events and incorporate their influence into its predictions. Current techniques for autonomously detecting traffic events, such as automated incident detection from traffic flow data and computer vision are vibrant fields of research but currently offer significantly less accurate data than actual human observations.

Therefore, introducing novel ways to identify and quantify disruptions using human input can improve modeling accuracy when flows are interrupted. The cost of hiring new people to monitor traffic and provide input data for prediction models is a high burden and therefore, is not an optimal solution. However, social networks provide a rich source of information broadcast and consumed by millions of vehicle operators in near real-time. Incorporating human-observed incident detection gleaned from semantic mining of social network data with existing traffic modeling techniques provides a rich source of real-time information for improving accuracy in traffic flow prediction.

Chapter I: Rationale

The rationale for pursuing this research follows several threads that the thesis itself intends to weave together. This chapter introduces these topics and narrows their scope to the specific approaches explored in this research. The organization is as follows:

Brief Introduction of Traffic Congestion: This section frames the problem, introduces factor that motivate the research, and sets the stage for the experimental solution.

Introduction to Artificial Neural Networks: This section describes the architecture of feed-forward neural networks and how they model information to generate predictions. It also describes how to train a neural network using back-propagation. It introduces the idea that training variations cause prediction variations and to provide context for using an ensemble of multiple ANN models as a mechanism for cross-validation.

Artificial Neural Networks in Traffic Prediction: This section describes the role of artificial neural networks in Data Driven Intelligent Transportation Systems, the current state of the art, and the lack of a viable automated data source for detecting incidents.

Using Twitter for Convergent Validity: This section describes how Twitter can be used as an early warning system for non-recurring events and how it can be harnessed to provide a human-curated data source for informing traffic prediction models.

Traffic Congestion

Traffic congestion is a topic that negatively affects nearly all people, regardless of whether they are contributing to it by driving a vehicle or trying to avoid it by using mass transit. The costs of traffic congestion are difficult to quantify but in the year 2000, it was estimated to account for 3.6 billion vehicle-hours of delay, 5.7 billion U.S. gallons of wasted fuel, and $67.5 billion in lost productivity (0.7% of GDP) or approximately $1000 per driver in large cities or $200 per driver in small ones. (Traffic Congestion) Congestion increases the time required to traverse road segments so, in addition to increased fuel consumption, it results in increased air pollution (Shawe-Taylor, De Bie, & Cristianini, 2006) and health problems such as heart attacks (Peters, et al., 2004).

Population growth continues to drive urban density. Despite the trend toward building more public transportation infrastructure in high density cities, traffic congestion can be expected to worsen. (Meliaa, Parkhurst, & Barton, 2011) However, we can assume that individual drivers are rational decision makers that will make an attempt to reduce their trip time if provided with sufficient information for doing so. (van Lint J. , 2006) This presents an opportunity to reduce traffic congestion by providing accurate traffic predictions to drivers so they can each optimize their individual travel plans. Individuals directly realize the benefit of this approach through shorter trip times. At the same time, the entire transportation system benefits from fewer vehicles unknowingly heading into areas of congestion, exacerbating backups in problem spots. While there are clear benefits to providing this information to drivers, it is not abundantly clear how to formulate such traffic predictions.

Many years of research have yielded computer modeling techniques that can predict the behavior of complex systems, such as traffic speeds in regional transportation systems, with high accuracy. These predictions suffer greatly when disruptions, such as traffic accidents, occur in these systems. Current techniques, such as automated incident detection from traffic flow data and computer vision are studied broadly but are significantly less accurate than actual human observations. Therefore, introducing novel ways to identify and quantify disruptions with human input can improve modeling accuracy while creating new topics for research to address this large, unmet need. Blending human-relayed incident detection mined from social networks with existing traffic modeling techniques provides a new method for improving accuracy in traffic speed prediction.

Artificial Neural Networks

Artificial Neural Networks (ANNs) are mathematical models that mimic the structure of a biological neural network. ANNs are comprised of a structure of interconnected nodes called neurons. (Artificial neural network). Each neuron is connected to other neurons to form a network, as in biological neural networks. Data is received by a neuron by connecting the output from one or more neurons to the subject neuron's input. Neurons in an ANN work together to model complex non-linear relationships between input and output data. Traffic prediction requires the capability to model complex relationships between present conditions and future conditions, which makes ANNs a reasonable choice for this application.

Structure

Neurons are grouped together into layers. Neurons within a layer depend only on output from neurons in the previous layer (neurons within a layer are not connected to one another.) The first layer of all ANNs is called the input layer, as these nodes receive the raw numeric input data, which is scaled or transformed by an activation function (typically a sigmoid function (Faghri & Aneja, 2007)), before it is emitted as output, which is consumed by the next layer. These neurons receive data by connecting the output from one or more other neurons. These neurons weight each input, combine it, and transform it via an activation function before emitting it as output. Data is transformed in this manner and flows from the input layer through one or more "hidden" layers until it eventually reaches the output layer, which performs its own weighted combination of the previous nodes' outputs and produces one or more final output values.

ANNs can be represented as a directed graph, as depicted in Figure 1 - An Artificial Neural Network. There are many types of ANNs that implement various schemes for structuring their nodes and the connections between them. This document uses ANN to refer to a Feed Forward ANN, where the graph does not contain cycles and data always flows from the input layer toward the output layer without re-entering a node or layer that has been previously visited.

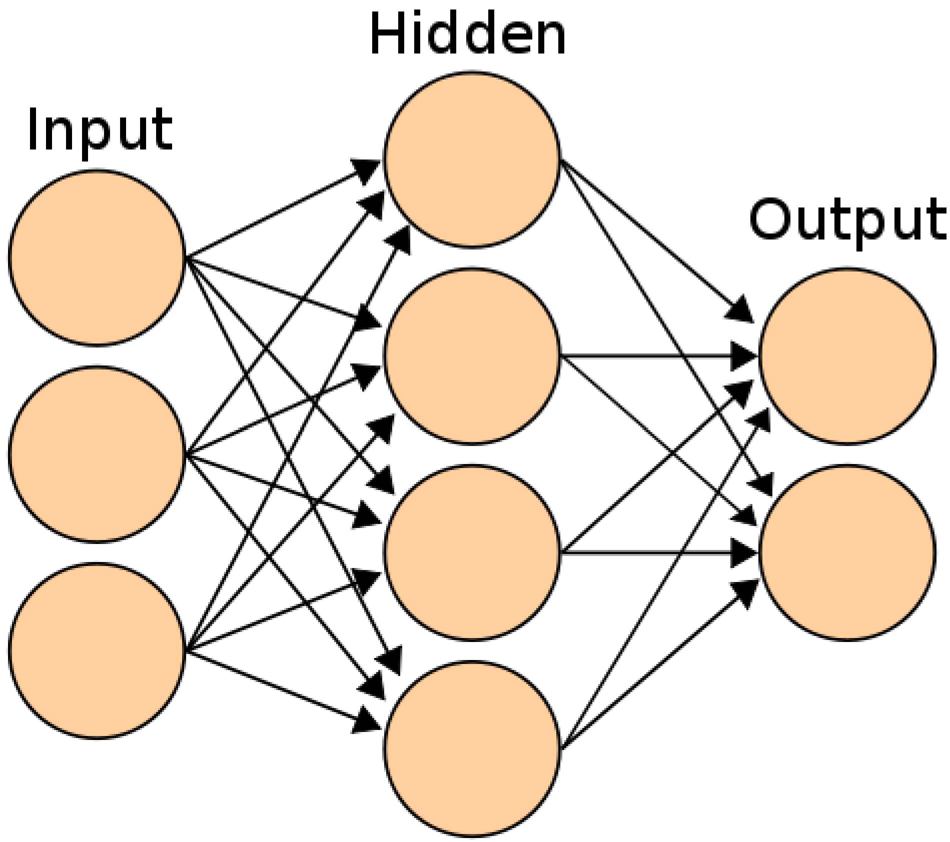


Figure 1 - An Artificial Neural Network (Artificial neural network)

The number of input nodes is fixed, as each piece of input data is fed to a distinct input node. The number of output nodes is also fixed, as each output node predicts a distinct piece of data. The number of hidden layers and nodes within each hidden layer is not prescribed. There are no heuristics for determining how many of each to use (Faghri & Aneja, 2007), so a trial and error approach is typically used to determine which configuration provides the best predictions. The complexity of an ANN is related to the number of nodes and layers that it contains. An increase in structural complexity affords an increase in “degrees of freedom” for an ANN to model more complex problems but requires an increase in training data to properly “learn” the problem space. It follows that it is best to use the simplest structure that yields acceptable results.

Each neuron performs its computation by receiving one or more inputs, performing a weighted combination of the inputs, scaling it according to the activation function, and emitting the result as its output. The weights that each neuron uses for combining its input data are independent from all other neurons. Neurons’ weights are assigned by an offline trial and error process called training, where example data is fed through the network and its weights are adjusted to minimize the error in predicting the expected output.

Training

ANNs can be trained using one of two methods: unsupervised learning and supervised learning. Unsupervised learning is a method of training an ANN without providing specific expected outcomes for it to attempt to predict. This model of learning is typically used in data mining or clustering applications where the intended outcome is not known a-priori. (Unsupervised Learning, 2012) Supervised learning refers to the method of training where an expected outcome is known and therefore allows direct computation of the error between the expected output and the predicted one. (Supervised Learning, 2012) This study is done using supervised learning, as the modeling experiments require the error to be quantified and actual traffic speeds are available for training & validation. Success will be measured by the relative error in predictions produced by each experiment.

Back-Propagation is the most common method for training an ANN. (Faghri & Aneja, 2007) It is performed by providing example data to an ANN and computing the error of its outputs. The error is then presented to the output nodes, which adjust their weights to minimize the error, which more accurately predicts the expected outcome. The error is then propagated backward to each node, where they perform their own adjustments until the input nodes are reached. (Backpropagation, 2012) After each node has adjusted its weight, the weight adjustment process is repeated through hundreds or thousands of iterations. These iterations are called epochs. Training an ANN is an iterative process that is done until the error of the predictions reaches an acceptable threshold or until the error reaches a plateau, where further training epochs do not yield an improvement.

The complexity of an ANN affects how much data is required to learn weights that will produce an accurate prediction. In the extreme case, training with a single example can allow the ANN to model the relationship between the input and output for that example. That ANN will be unable to predict anything other than its single example, so its utility is quite limited. Ideally, the examples in the training dataset would include all possible interactions between input variables that are necessary to produce all possible outcomes. In practice, this is not possible.

Evaluation

Given that ANNs can model complex, non-linear relationships between input and output data (Faghri & Aneja, 2007), many training epochs must be executed to arrive at internal weights that yield accurate predictions. If too many epochs are run, the ANN risks being over-trained. If an ANN's structure is sufficiently complex enough to model the relationship between its input and output and is trained over an excessive number of training epochs, the nodes' weights effectively "memorize" the training data. This behavior is called over-fitting, as the ANN no longer approximates the dataset from which the training data was sampled. Instead, it only models the training dataset itself. (Artificial neural network)

Ensembles and Confidence Levels

There are many parameters that can be set for how an ANN adjusts its weights during each epoch while training. For example, the concept of momentum is applied to weights to ensure that they aren't snapped to drastically different values in each training epoch. Given the connected-ness of the ANN structure, a change in one neuron's weight will affect all downstream computations. Momentum is a smoothing function to reduce abrupt changes from one epoch to the next, which makes the learning process less erratic, allowing weights to be optimized in fewer epochs. (Faghri & Aneja, 2007) In addition, a ceiling for the amount of change applied to a weight in each epoch (known as the learning rate) can be adjusted to aid the training process. There is no heuristic for determining how to set these values, so experiments are typically repeated using different values. Changing the training parameters will cause each neuron's final weights to be different, which results in a different prediction. (Faghri & Aneja, 2007) This means that two models that are trained on the same data can produce different predictions, depending on the parameters used to train them. Which one is correct?

A common approach to arbitrate between ANNs trained with different parameters or on a different subset of training data is to train multiple ANNs, generate predictions from each of them, and combine the predictions. The theory behind this approach is that the random error across a whole suite of ANNs will be orthogonal and will cancel out, while their "agreement" will result in a more accurate prediction. This is known as an Ensemble (Ensemble Learning, 2012) and is an entire field of study in its own right. Several techniques regularly used in Ensemble Learning have been applied in traffic prediction to get a rough measure of the confidence of a prediction.

An important point to remember is that ANNs are just a chain of mathematical functions. They will always produce an output for a set of inputs, without any indication of how accurate it is. This means we won't know how much credibility to give a model's prediction without more data. However, a good proxy for determining prediction confidence can be obtained by generating predictions from an Ensemble of ANNs and quantifying the dispersion of their predictions. In cases where there is relatively low dispersion (high agreement), the predictions can be assumed to have high confidence. Conversely, low agreement among the ANNs indicates that the prediction has low confidence. (van Lint J. , 2006)

Artificial Neural Networks in Traffic Prediction

There are many approaches to monitoring and predicting traffic today. Zhang, et al. propose a Data Driven Intelligent Transportation System (D^2ITS) which employs computer vision, automated incident detection, and sensor-based models, that learn complex traffic interactions; to stitch together massive amounts of data. The information produced by D^2ITS is then used to inform infrastructure planning, vehicle management systems, and traveler information systems. Ultimately, it was determined that highly accurate computer vision systems are currently too expensive for most transportation institutions; and today's state-of-the-art computer vision technology is not mature enough to provide information with enough consistency to be relied on. In addition, they concluded that computer vision systems had problems with difficult shadows and was confused by variations in vehicle types and sizes, which occur very regularly in transportation infrastructures. (Zhang, et al., 2011) ANN-based automated incident detection systems also suffer significant decreases in accuracy when environments change due to rain, snow, or even glare. (Shehata, et al., 2008)

However, data driven models fed by in-road sensors have proven very successful in modeling the complex interactions between many factors that can influence future traffic conditions; (van Lint J. , 2006) and have been considered an essential component of any intelligent transportation system for almost two decades. (Cheslow, Hatcher, & Patel, 1992) Data driven models are also very cost-effective, as they use the same data that is already collected in most urban centers to publish current speed and flow information. The two most common approaches use data modeling to predict travel time along a route (van Lint J. , 2004) and to predict instantaneous speeds and/or flow on a road at some future time. (Faghri & Aneja, 2007) (Park, Messer, & Urbanik II, 2007) (Zheng, Lee, & Shi, 2006) However, Zhang, et al. claim that making predictions based on a single data source, such as road sensor data, does not yield reliable accuracy in D^2ITS. As an example, traffic accidents generate different patterns than those seen in recurrent congestion, which is why they offer the multi-part strategy of using computer vision and automated incident detection in addition to sensor-based modeling. This "fusion strategy" provides convergent validity for predictions, as it can cross-validate a prediction based on inputs from multiple sources. (Zhang, et al., 2011) The key takeaway is that road sensor data needs to be augmented by somehow "seeing" when something out of the ordinary is occurring that changes how traffic behaves.

Challenges in Predicting Traffic Conditions using ANNs

A number of attempts have been made to synthesize a multi-source model for prediction, due to the onerous requirements for obtaining real distinct sources, cited by Zhang, et al. These approaches are attempts to work around the error intrinsic in single-source prediction models without the burden of true multi-source data. For example, random subsampling works by teaching only part of the dataset to a model. This yields models trained on slightly different datasets, which causes variations in their predictions.

Different modeling techniques respond to data corruption and omission instances very differently. (Kotsiantis, 2007) One way to leverage the heterogeneity of errors is to utilize multiple models of different types in Ensemble Learning (Ensemble Learning, 2012) to effectively "bolster the signal" and "cancel out" the noise in predictions. (van Lint J. , 2006) Ensemble methods yield predictions with a lower error rate than a single model. (Park, Messer, & Urbanik II, 2007)

The proper selection of input data is important to consider when using ANNs to model traffic data. Likewise, the capability for ANNs to produce accurate predictions decreases as predictions are made further in the future. Chen & Chen experimented with many different configurations of input and output data to determine which produced the most accurate predictions. They considered the temporal granularity of samples, how far they needed to "look back", and how far they could "look forward." Their conclusion was that data older than 32-48 minutes is not useful in predicting future traffic conditions. At the same time, they found that prediction accuracy degraded quickly as "look forward" periods increased in all cases. When using samples with a granularity of 4 minutes, accuracy dropped quickly when predicting between 8 and 16 minutes in the future. (Chen & Chen, 2007)

These approaches attempt to minimize errors and omissions in training data as well as prediction inaccuracy caused by the dynamics inherent in one model vs. another, all while ensuring that the input and output data windows yield the best predictions possible. However, all of the aforementioned techniques are still single-source prediction mechanisms. When performing online predictions in the field, their data still comes from a single source: in-road sensors. This does not meet the multi-source requirement set forth by Zhang, et al. for achieving reliable accuracy. In addition to these techniques, a separate source of data is needed to cross-validate the sensor data and indicate when the traffic flow does not reflect a normal pattern.

Human Validation

Data Driven Intelligent Transportation Systems (D^2ITS) are aimed at generating information autonomously. They are intended to drive regional transportation infrastructure that can tune itself dynamically via on-ramp metering and other mechanisms to consistently optimize traffic flow. A part of ITS involves presenting users with accurate information about its current state without requiring human curators. However, the current state-of-the-art technology is not capable of replacing humans. (Zheng, Lee, & Shi, 2006)

Therefore, we must conclude that traffic data generated by human curators is required to provide data driven models with the information required to model non-recurring traffic events, such as a traffic accident or spontaneous lane closure.

Twitter as a Proxy for Human Validation

Social Networking has become nearly as ubiquitous as email. As of August 2011, the social network Twitter had over 300 million users, generating over 300 million 140-character messages, known as "Tweets," each day. (Twitter Blog, 2011) Twitter is unique among social networks, in that the follower / followee model is much more loosely coupled than a traditional friend-based network model, such as Facebook. Actual friend-to-friend collaboration on Twitter is done by a sparse, hidden network of connections that underlies the declared follower / followee relationships and only represents about a quarter of Twitter activity. In fact, most inter-personal connections on Twitter are meaningless from an interaction perspective (Huberman, Romero, & Wu, 2008) Even conversations with complete strangers can easily be joined based purely on interest. (de Moor, 2010) Twitter requires no reciprocity, so it resembles an information dissemination network more than another social network. (Kwak, Lee, Park, & Moon, 2010)

Twitter does, however, lend itself well to a different collaboration model. It applies a principle of least collaborative effort (de Moor, 2010) by limiting messages to 140 characters and allowing everyone to respond to anything on the network. Viewing Twitter as a broadcast + amplify model of collaboration more closely resembles how its users interact. In this model, a message is broadcast from a user to whoever is following them and is also accessible by anyone who searches for it via the Twitter API or website. Followers who find value in the information can then re-broadcast (re-tweet) it, amplifying the message and exposing it directly to their followers – a potentially distinct set of new recipients. Sakaki, et al. showed how Twitter tweets could be used to detect and broadcast the location and trajectory of earthquakes faster than the Japan Meterological Agency. By tracing the initial tweet(s) back, they were able to locate the epicenter of the earthquake and subsequent tweets were found to radiate in the direction of the shock wave's travel. (Sakaki, Okazaki, & Matsuo, 2010)

Challenges in Semantic Mining of Twitter Messages

Twitter's loose model for inter-personal connections and lack of structure for tweets present challenges in mining information from its data. The connection graph is not a good proxy for a Page Rank-style authority model so discovering the structure of a hub and spoke topology among users requires an expensive probabilistic approach. (Lawrence, 2011) (Huberman, Romero, & Wu, 2008) Therefore, it is better to start with a known authoritative source when examining how data propagates through Twitter.

The abbreviated message style of Twitter has fostered the use of hash tags to express the relationship of a tweet to a topic or concept. However, even mining sentiment (which is much simpler than a full semantic classification) from hash tags requires a statistical approach. (Davidov, Tsur, & Rappoport, 2010) Twitter's simplicity creates a low barrier to posting, which allows more noise into network from both automated spam tweets and ad-hoc hash tag creation (which may simply be misspellings or rephrasing of existing hash tags.) Most importantly, most hash tags tend to have a short life, limiting their long-term utility. Of the hash tags with a long lifespan, such as #obama, most end up applied inconsistently over time. The few long-lived hash tags that managed to maintain semantic consistency over time tend to be application or source names included in tweets generated automatically by applications. (Laniado & Mika, 2010) Therefore, attempts to mine Twitter data cannot rely on hash tags to accurately map a tweet to a concept, which rules them out as a proxy for semantic analysis.

Twitter's mechanism for re-broadcasting tweets while preserving attribution (re-tweeting) is problematic, precisely due to the structure-free format of Tweets. The process of re-tweeting is analogous to forwarding email but the individual user is not forced to specify that it is a re-tweet and include attribution, because those constraints undermine the principle of least collaborative effort. Therefore, re-tweets are subject to stylistic variations in both their structure and the motivation for their re-broadcast, making them difficult to aggregate. The information that is re-tweeted tends to be time-sensitive, such as news or traffic (Boyd, Golder, & Lotan, 2010). This implies two things: 1) traffic information itself likely to be amplified and 2) detection and aggregation of traffic information without an authority is difficult.

Semantic Analysis of Traffic Broadcasts from Authoritative Sources

Official traffic broadcasts do not suffer from the problems inherent in generalized mining of Twitter data. Major metropolitan transit authorities, such as the Washington Department of Transportation, Oregon Department of Transportation, and California Department of Transportation broadcast their own messages on the Twitter accounts. (Their tweets can be viewed by the Twitter usernames @wsdot_traffic, @ODOTPDXMtHoodFA, and @Caltrans8, respectively.) These user accounts are recognized authorities for traffic data that do not require cross-validation through a social graph. The trained staff of a regional transportation authority is intentionally monitoring and reporting on traffic conditions affecting the majority of their transportation system. It can be assumed that these primary sources cover the majority of interesting roadways and use consistent conventions for naming events, locations, and severity.

Restrictions in Twitter message length, coupled with the fact that transit employees are intended to represent government agencies, actually benefit semantic processing of traffic broadcasts. The messages must be concise in specifying the location, type, and severity of an event, so the variations expected in their contents should be more constrained than free-form text. Finally, abbreviations for common elements should be more consistent among broadcasts from trained staff than from the public-at-large.

Detection of re-tweets remains a difficult problem to solve. However, its importance in modeling traffic data is secondary to the proper aggregation of primary data sources, such as initial tweets from transit authorities. While the accurate capture and aggregation of traffic broadcast re-tweets could provide incremental value in traffic predictions, it is theorized that this is not critical in prediction accuracy.

Traffic Accident Ontologies

For consistency and reusability, the plain text of tweets should be mapped into an ontology. However, few ontologies have been published for classifying traffic impediments according to the concrete impact of their severity. TADO is a risk-based ontology that focuses on the circumstances surrounding the occurrence of an accident with the intent of predicting and avoiding areas with a high occurrence of accidents. (Wang & Wang, 2011) It includes detailed geospatial information about type of structures near an incident (the type of roadway and nearby buildings) in addition to environmental conditions such as light and weather during an incident. The latter provides an interesting avenue for additional signals to add in future iterations of this experiment. However, the ontology does not include any information about the severity of an incident and its impact on surrounding roadways.

Traffic-related ontologies have also been developed with the intent of applying spatial clustering algorithms to incidents (Hwang, 2003) but these typically omit details about the magnitude of the incident that could aid in prediction of traffic around it. Dieng's experiment in graph comparison used several ontologies produced by traffic experts, which come closer to the mark. They call out categories of accidents, such as Conflict-in-intersection and Accident-of-2-

vehicles-in-current-section (Dieng, 1996) but lack attributes that are relevant to predicting traffic speeds, such as whether extraordinary visual distractions such as fire or an overturned vehicle are present.

An ontology to classify tweets about traffic incidents should leverage the relevant concepts from prior efforts, where applicable. Given that the literature does not contain a ready-made ontology for semantic classification of traffic incident tweets with the intent of describing their impact on traffic speeds, the concepts that are published must be paired with attributes gleaned from uncovering patterns present in tweets from traffic authorities to form a traffic incident ontology.

Chapter II - Methods

This chapter describes the concrete aspects of the experiment. The experiment identifies in-road speed sensors in popular road segments and trains ANN models to predict future speeds. The organization is as follows:

Experimental Design: This section describes the structure of the experiment and how it addresses threats to validity.

Data Selection and Preparation: This section describes how datasets were chosen for experimentation, their characteristics, and how they are modeled internally prior to generating the training and validation datasets.

Traffic Incident Ontology: This section discusses different attributes of Tweets and the ontology built from analyzing them for their impact on traffic disruptions.

Experimental Parameters: This section discusses how the specifics of setting up this forecasting experiment, including dataset subsampling and how far in the future it attempts to forecast speeds.

Evaluation Criteria: This section describes the methods for evaluating the experiments and how they are calculated.

ANN Architecture and Training Configuration: This section describes the pre-work done to determine the optimal settings for training the ANNs and the result configuration that was used for training the ANNs in the experiment.

Terms and Definitions: This section provides a list of terms and succinct definitions for concepts that have been introduced.

Experimental design

This experiment generates predictions of road segment speeds in the future. The predictions are made by modeling either sensor data combined with social data (the treatment group) or the sensor data without social data (the control group.) The experimental hypothesis claims that the error of the predictions generated by sensor and social data combined will be less than the error of the predictions generated by sensor data alone. See Figure 2 - Experiment Architecture.

An experiment is conducted for a randomly selected sensor using a Posttest-Only Control Group Design:

$$\begin{array}{ccc} R & X & O \\ R & & O \end{array}$$

The "treatment" applied to the treatment group is the incorporation of social data in its model. The control group is the same subject sensor without the "treatment" social data in its model. This is acceptable because the experiment subjects are simply datasets that are not altered by history or maturation effects. The validity of an experiment conducted on a single road segment raises questions of whether a selection bias threatens its internal validity. To address this threat, this study uses a random sample containing ten percent of the sensors available on the subject road segment to conduct the experiment in multiple locations, to measure its outcome.

ANNs present a unique challenge in experimental design, as the best practices for constructing them can produce varying results when executed multiple times on the same data. This is a side-effect of the standard process of randomizing an ANN's weights prior to training it. To address this problem, techniques called K-Fold Cross Validation (also known as random subsampling) and Ensemble Forecasting are used. K-Fold Cross Validation randomly selects data points for a subset of the training data and constructs an ANN model using that subset. The process is repeated K times to produce K different models. Ensemble Forecasting combines the results of multiple models to produce a prediction that represents the consensus of the ensemble. Combining model predictions into an ensemble increases their accuracy and helps to quantify prediction confidence. In this study, the treatment group contains an ensemble of models trained with social data and the control group contains an ensemble of models trained without social data.

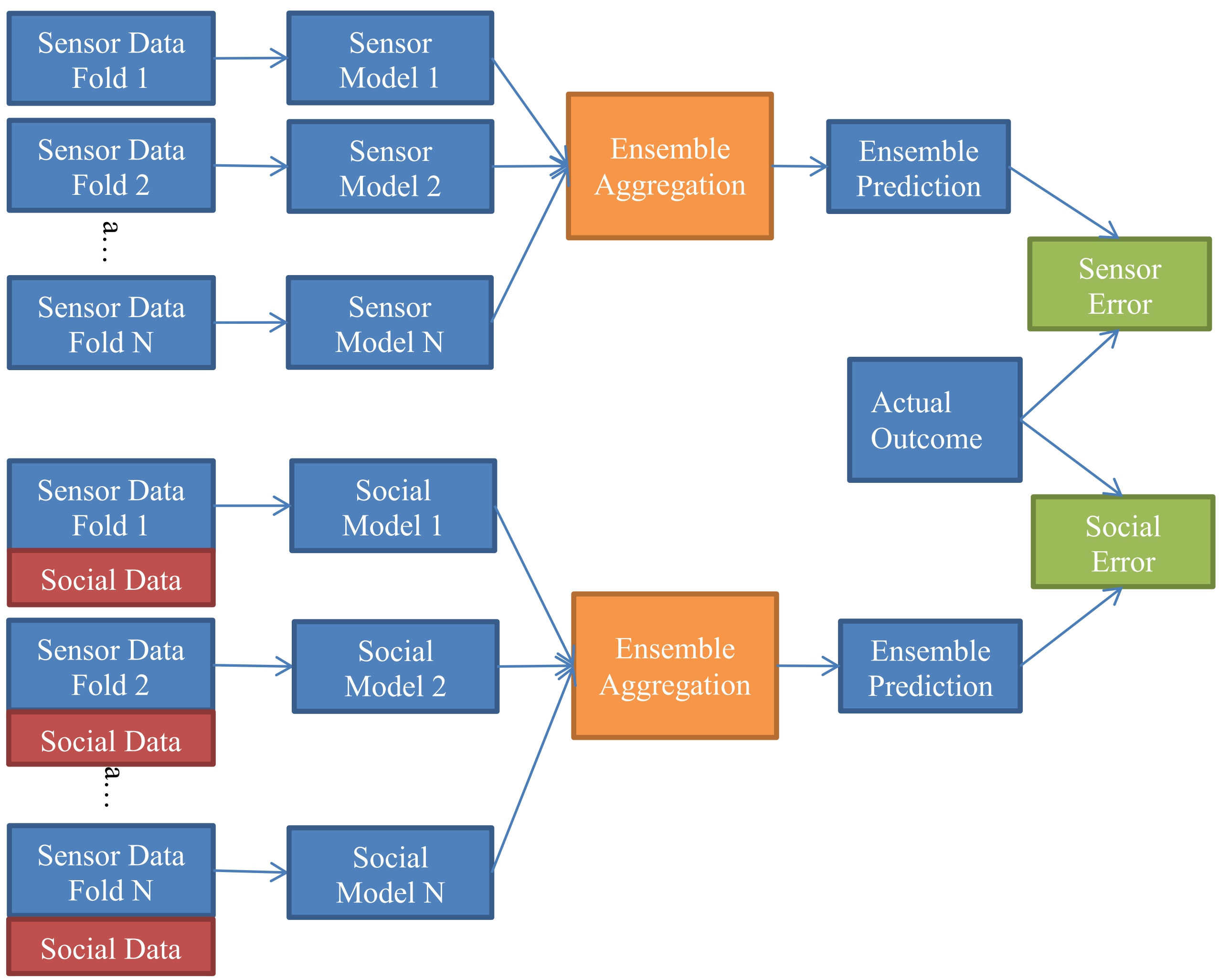


Figure 2 - Experiment Architecture

Data Selection and Preparation

This section describes how data is collected and processed prior to modeling for this study.

Road Segment Selection

Choosing which road segments to model is challenging. Selecting segments from the complete pool of available segments at random avoids selection bias but has low utility, as most tweets from @WSDOT_Traffic broadcast events on high occupancy segments. In fact, tweets about events occurring on segments of the I-5 corridor outnumber those on the next most common roadways by a ratio of 3:1 (See Figure 3 – Tweets broadcast by @WSDOT_Traffic user between 2/12/2012 and 3/13/2012. Note that the "N/A" bucket comprises conversational tweets that are not intended to notify consumers of specific events.) Therefore, this experiment only considers segments along I-5 within the Seattle city limits. (@WSDOT_Traffic is primarily focused on Seattle. Other regions' traffic data is broadcast using different Twitter handles, such as @WSDOT_Tacoma.) The city limits of I-5 are defined as being bounded by milepost 156 at the south end and 174 at the north end.

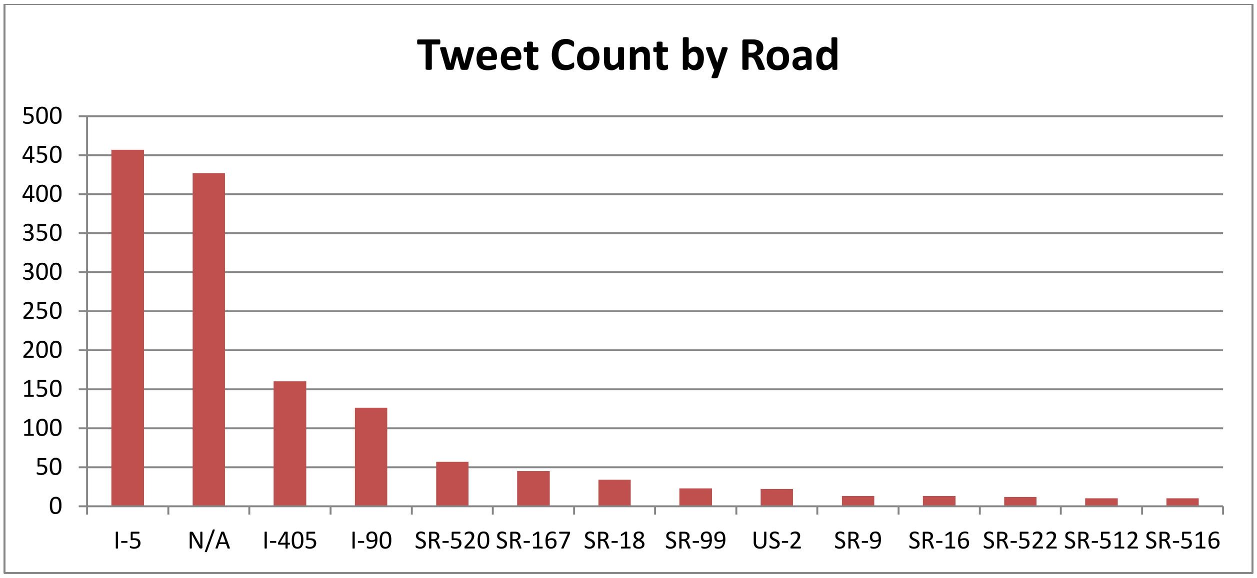


Figure 3 – Tweets broadcast by @WSDOT_Traffic user between 2/12/2012 and 3/13/2012.

Despite constraining the problem to this roadway within these bounds, there are 223 active sensors that can be modeled. It is impractical to attempt to model and analyze all of them, especially when each lane a separate sensor which produces very similar readings to those around it. It is also important to choose an unbiased, yet representative subset when sampling which sensors to model. Modeling ten percent of the sensors provides a representative sample. To select the sensors without bias, a random number is assigned to each sensor. Sensors with numbers greater than or equal to .9 are modeled in this experiment. This approach yielded 31 sensors to model. Table 1 - Sensors to Mode details the specific sensors modeled in this experiment. The ID assigned below is a serial number given to each sensor when it is imported into the experiment database. For convenience, these IDs are used to refer to specific sensors instead of the WSDOT Sensor ID for the remainder of this document.

ID	WSDOT Sensor ID	Milepost	Direction	Lane	Location Description
38	005es15821:_MNH_T5	158.21	N	5 (HOV)	S. Victor St
39	005es15845:_MN__T2	158.45	N	2	S. Norfolk St
49	005es15892:_MN__T3	158.92	N	3	S. Benefit St
72	005es15996:_MS__T1	159.96	S	1	S. Holden St

77	005es16040:_MN__T2	160.40	N	2	S. Myrtle St, NB
80	005es16040:_MNH_T5	160.40	N	5 (HOV)	S. Myrtle St, NB
87	005es16064:_MS__T4	160.64	S	4	S. Holly St
91	005es16097:_MN__T3	160.97	N	3	S. Graham St.
94	005es16120:_MN__T1	161.20	N	1	Swift Ave-NB
98	005es16120:_MNH_T5	161.20	N	5 (HOV)	Swift Ave-NB
105	005es16186:_MN__T2	161.86	N	2	S. Pearl St
108	005es16186:_MS__T2	161.86	S	2	S. Pearl St
109	005es16186:_MS__T3	161.86	S	3	S. Pearl St
111	005es16186:_MSH_T5	161.86	S	5 (HOV)	S. Pearl St
118	005es16237:_MS__T3	162.37	S	3	S. Oregon St
119	005es16237:_MSH_T5	162.37	S	5 (HOV)	S. Oregon St
132	005es16377:_MN__T1	163.77	N	1	S. Walker St, NB
141	005es16395:_MNH_T5	163.95	N	5 (HOV)	S. Holgate St, NB
149	005es16426:_MN__T3	164.26	N	3	S. Atlantic St
151	005es16466:_MN__T1	164.66	N	1	4^{th}/Dearborn-NB
155	005es16466:_MS__T1	164.66	S	1	4^{th}/Dearborn-SB
161	005es16512:_MN__T4	165.12	N	4	Yesler Way, NB
168	005es16583:_MN__T4	165.83	N	4	University St-NB
169	005es16732:_MN__T1	167.32	N	1	E. Galer St
176	005es16802:_MS__T4	168.02	S	4	E. Roanoke St
179	005es16831:_MS__T3	168.31	S	3	E. Hamlin St
200	005es17075:_MS__T2	170.75	S	2	Lake City Way
206	005es17162:_MS__T2	171.62	S	2	NE 88^{th} St
233	005es17328:_MS__T3	173.28	S	3	NE 120^{th} St
239	005es17375:_MS__T2	173.75	S	2	NE 130^{th} St-SB
242	005es17375:_MSH_T5	173.75	S	5 (HOV)	NE 130^{th} St-SB

Table 1 - Sensors to Model

The distribution of sensors is depicted in Figure 4 - Sensor distribution.

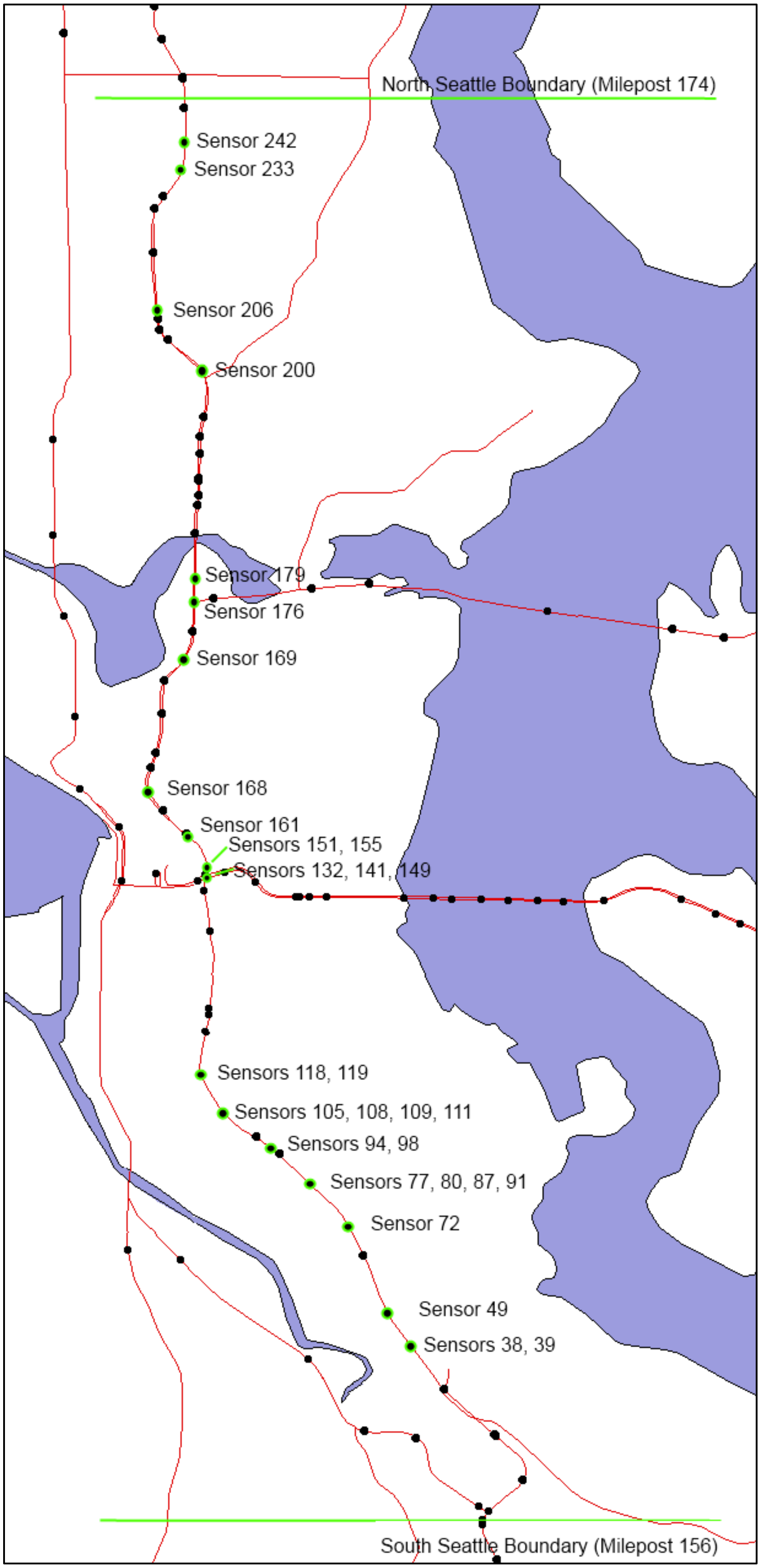

North Seattle Boundary (Milepost 174)
Sensor 242
Sensor 233
Sensor 206
Sensor 200
Sensor 179
Sensor 176
Sensor 169
Sensor 168
Sensor 161
Sensors 151, 155
Sensors 132, 141, 149
Sensors 118, 119
Sensors 105, 108, 109, 111
Sensors 94, 98
Sensors 77, 80, 87, 91
Sensor 72
Sensor 49
Sensors 38, 39
South Seattle Boundary (Milepost 156)

Figure 4 - Sensor distribution

Collecting Sensor Data

The traffic data used in this experiment is sourced from the Washington Department of Transportation (WSDOT) and is available for downloading at http://data.wsdot.wa.gov/Traffic/NW/FreewayData/5minute/. The data is provided in five minute intervals for each sensor. The data is downloaded, extracted, and pre-processed for import into a database by a series of steps depicted in Figure 5 - Ingestion Pipeline.

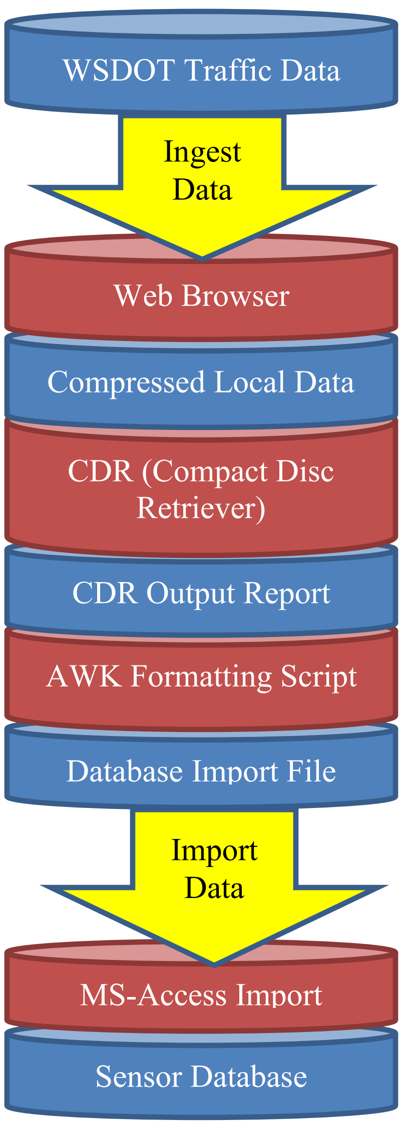


Figure 5 - Ingestion Pipeline

The actual data published by the WSDOT is a packed format that is designed to reduce its size. The packed format is more efficient than plain text for storing and transferring the data over the Internet. The WSDOT provides tools to extract the raw data from their packed format, so the packed format's details are uninteresting. The data format information is described for completeness and ease of reproducing the experiment data.

The WSDOT's tool called CDR (Compact Disc Retriever) is used to extract the raw data from the packed files. The procedure used to extract the data is described in Appendix 1 – WSDOT Data Extraction Method.

Collecting and Labeling Twitter Data

Twitter exposes a public REST API for searching its tweets, as documented at https://dev.twitter.com/docs/api/1/get/search. It offers multiple options for searching their data store but this experiment only uses tweets from an authoritative source; the Washington Department of Transportation (WSDOT). WSDOT has several Twitter accounts that offer information filtered by locality, such as @WSDOT_Tacoma and @WSDOT_Passes. This experiment uses the broadest account offering Seattle traffic information, @WSDOT_Traffic.

The Twitter search API does not consistently return tweets older than one week, so the WSDOT_Traffic tweets were captured each week over a four week period starting February 13, 2012 and ending March 12, 2012. The timestamp and the contents of the tweet, including any hashtags, were recorded.

The data was hand-labeled to ensure that the quality is higher than that of an automated process while helping to inform the ontology. It was difficult to determine which attributes occurred frequently enough to label a-priori. However, only a few attributes occurred regularly enough to include in the ontology. See the section Traffic Incident Ontology for details. In addition, very few instances of individual people re-tweeting WSDOT messages occurred during the data collection period. Of those, roughly half were re-tweets by local news organizations, such as King 5 News, komonews.com, and Northwest Cable News. Further, the messages were not usually re-tweeted verbatim, so correlating tweets with re-tweets quickly became a sparse data problem whose application in this research would be beyond the scope of this work.

Mapping Tweets to Sensor Data

The cardinality of tweets to sensor data records is many-to-many relationship: a tweet may describe an incident affecting a large area, so speeds recorded by many sensors could be reduced by it. Likewise, multiple tweets can affect speeds reported by a single sensor, as in the case when multiple incidents occur. Each set of sensor training data includes all tweets, regardless of whether they are expected to influence the speeds observed by the sensor. The magnitude of a tweet's impact is determined by its proximity to the sensor, in addition to the attributes of the actual event, such as the capacity impact. To address the case of multiple tweets causing an impact at the same time, the training data for a time segment is repeated once for each tweet that is active. This is done to prevent models from learning interaction effects between concurrent tweets. Combining multiple tweets into a single input is an interesting avenue for exploration but is out of scope for this experiment.

Twitter assigns timestamps to tweets as they are broadcast. The range of values for these timestamps is nearly continuous, which is very different from the timestamps assigned to the sensor data. The WSDOT reports speed information in five minute increments, so the two schemes must somehow be mapped together. This experiment aligns tweet timestamps to sensor timestamps by determining which sensor timestamp occurred after the tweet and replacing the tweet's timestamp with the sensor's. This implies that a tweet broadcast at 1:01AM would be treated as occurring in the 1:05AM time slot. The approach is designed to be slightly conservative and never allow out-of-order errors in training data, where the tweets would be seen as describing events happening in the future. (Tweets used in this experiment are always reactive, so they always correspond to an event that exists, rather than conditions that could cause an event.)

Data Model

Data is normalized into a star schema: when importing new records, sensor names and dates are scanned for values that don't already exist in the Sensor and DateTimes dimension tables. New values are inserted into these tables with unique numeric identifiers. The actual speed information is stored in a fact table named SpeedData. The fact table only contains numeric data, as the human-readable fields have been replaced with numeric ones during normalization. The schemas and cardinality of these tables are depicted in Figure 6 - Normalized Data Model.

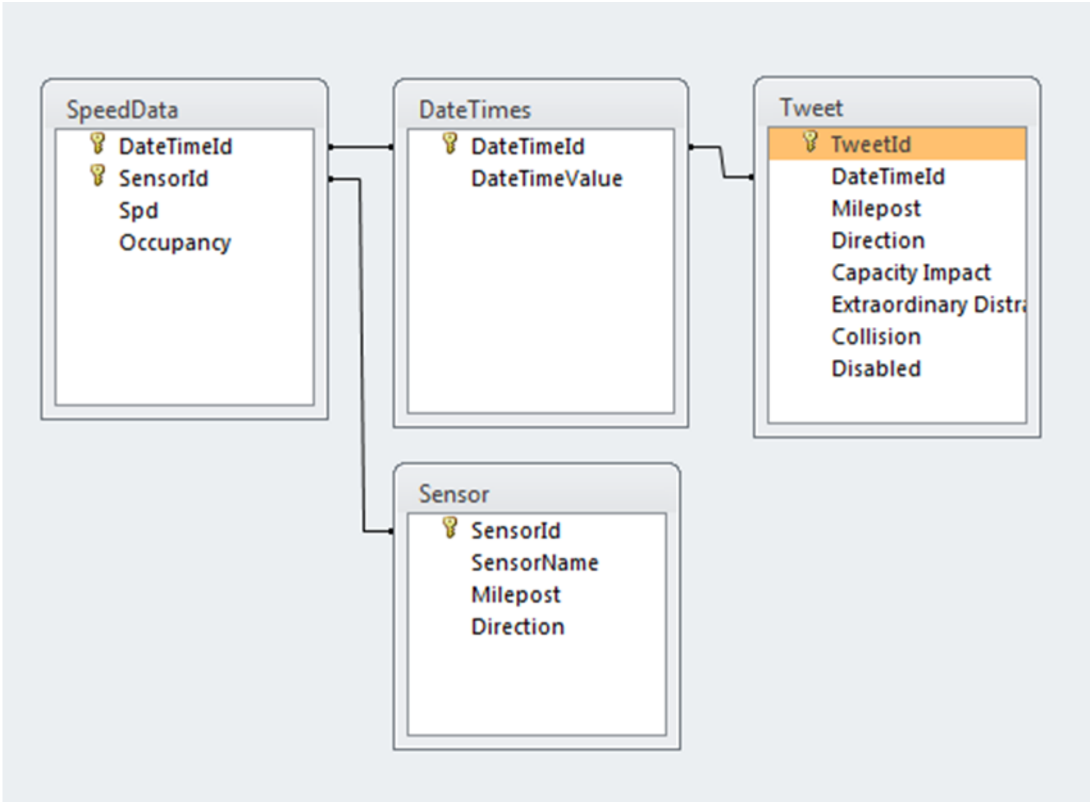


Figure 6 - Normalized Data Model

Tweets are incorporated into the data model as a separate table, with DateTimeId as a foreign key that associates them with a set of timestamps. When training or evaluation datasets are generated, the tweets' proximity to a sensor and whether their direction aligns with the sensor's direction are computed.

Traffic Incident Ontology

Initial attributes were derived from the literature and general knowledge of types of incidents that have occurred. All incidents are assumed to have a flat hierarchy, existing as uniform objects containing the same attributes. Research spanning a longer time period may allow for a more full-featured ontology. However, aside from opaque disabled vehicles and collisions, ontological information was quite sparse in the WSDOT tweets.

Message Type

A tweet can announce a new incident, the continuation of an existing incident event, or the clearing of an incident. Incidents have a well-defined life cycle: an incident occurs, it has some impact (which could last for a few minutes or several hours), and it clears. Therefore, tweets broadcast to inform users of an incident should map to one of these three events. Some tweets are not intended to cite a specific incident but describe present conditions on a road segment as their event. The cardinality of tweets to events is many-to-one: zero or more tweets may be broadcast for each event type. These indicate whether an event should have a negative effect on future predictions (initial / continuation) or whether they indicate that conditions should improve. For this study, announcements of current conditions are considered to Continuation messages, as they indicate that something is occurring but don't indicate whether traffic is taking a turn for the better or worse. See Table 2 - Message Type Examples.

Message Type	Example
Initial	On I-5 northbound just south of SR 18 there is a disabled vehicle blocking the right lane.
Continuation	Tow truck arrived on scene: On I-5 northbound just south of SR 18 there is a disabled vehicle blocking the right lane.
Clearing	Cleared: On I-5 northbound just south of SR 18 there is a disabled vehicle blocking the right lane.

Table 2 - Message Type Examples

Location

Impediments must correspond to a physical location to be useful for measuring their impact on traffic speeds. Impediments will always be considered to be on a roadway of interest, so the name of the roadway is one part of the location. Incidents can occur anywhere along a roadway, so the location must be specified, in addition to the roadway name. In order to calculate the distance from an incident to a speed sensor, this ontology uses the milepost to specify the location of incidents. Finally, major roadways typically have two directions of traffic flow, so this will be indicated as well.

Examples of tweets with ontological location mapping are shown in Table 3 - Location Attributes.

Road Name	Direction	Milepost	Tweet Text
I-5	S	163	On I-5 southbound at S Spokane St there is a collision partially blocking the right lane.
I-5	N	165	On the I-5 northbound collector-distributor at Yesler Way there is a disabled vehicle partially blocking the right lane.
I-405	N	2	On I-405 northbound just north of S R167 there is a collision blocking the right center lane and the right lane.
I-5	Multiple	164	UPDATE: List of ramp closures...N/B I-5 & S/B I-5 to Dearborn Street & N/B I-5 to E/B I-90 all due to police activity.

Table 3 - Location Attributes

Extraordinary Distraction

This binary attribute attempts to include concept of whether an extraordinary distraction is present that can magnify the severity of an incident. The value of this attribute is zero, unless the description of the incident contains information indicating that there is a significant reason that the incident would draw more attention than a stalled vehicle or collision. See Table 4 - Extraordinary Distraction Examples.

Distraction Examples
Car fire
Police activity
Overturned vehicle
Ambulance on scene

Table 4 - Extraordinary Distraction Examples

Capacity Impact

Blocking impediments typically affect a single lane, reducing the capacity by the amount of traffic carried by that lane. The impact is significantly increased when multiple lanes are blocked. Conversely, when an incident is moved to the side of the road, it represents a visual distraction but no physical impediment to flow, so its capacity impact is zero. See Table 5 - Capacity Impact Examples.

Capacity Impact	Description	Example
0	No physical reduction in capacity.	Just cleared to the right shoulder. RT @JenniferKimKOMO: disabled car nb5 approaching 272nd... in the left lane
1	Mild reduction in capacity. Single lane	On I-5 southbound at NE 50th St there is a disabled vehicle blocking the right center lane.
2	Severe reduction in capacity: multiple/all lanes	All lanes still blocked on S/B I-5 at Ravenna, along with the ramp from Ravenna. Here's a look: http://t.co/mDBqJmVk

Table 5 - Capacity Impact Examples

Incident Type

The most common types of incidents are stalled or disabled vehicles and collisions. They are categorized accordingly. See Table 6 - Incident Type Examples.

Incident Type	Example
Stall	On I-5 northbound just south of SR 18 there is a disabled vehicle blocking the right lane.
Collision	On I-5 southbound at Dearborn St there is a collision blocking the HOV lane.

Table 6 - Incident Type Examples

Experimental Parameters

Appropriate Look-ahead and Look-back Periods

The design of a forecasting experiment includes the selection of look-ahead and look-behind periods. Choosing the appropriate periods for look-ahead and look-back is one of the most difficult challenges in setting up a prediction experiment of this nature. The look-ahead period determines how far into the future the model is attempting to look ahead. Long look-ahead periods tend to be inaccurate because the data directly affecting the prediction is not included as part of the input. In the extreme case, traffic speeds on a road one year from today would be most closely related to the conditions near that time and likely only coincidentally related to conditions today.

The look-back period is the amount of time in the past we consider conditions relevant for predicting current conditions. This period also needs to be chosen carefully, as the influence of previous readings decrease as their ages increase. Providing extraneous data increases model complexity, as the model must learn to ignore data that doesn't influence the outcome. Therefore, a look-back that is so large that it contains old data that does not affect the outcome will actually decrease the model's accuracy

Selection

Chen and Chen experimented with the effect of look-back and look-ahead periods on prediction accuracy. (Chen & Chen, 2007) Their experiments using data collected at four minute intervals is most closely related to the five minute data collection interval used in this experiment, so the conclusions they made about that dataset are applied here.

They concluded that using five collection intervals (20 minutes) as the look-back period produced the most accurate predictions. They did not include results for look-back periods more than five collection intervals in duration. They are not explicit that six or more intervals would be unhelpful, but they do cite the most accurate look-back intervals as four and three, when the collection interval is eight minutes (32 and 24 minutes, respectively.) It can be inferred that looking back more than 30 minutes does not yield more accurate predictions. This experiment uses five collection intervals as the look-back period (25 minutes) as an approximation of Chen and Chen's findings.

Look-ahead period selection is a balance between utility (it is not useful to predict conditions 1 second in the future) and accuracy (predictions 60 minutes in the future are not useful if they are only accurate 1% of the time.) The accuracy of look-ahead periods cited by Chen and Chen showed a steady *decrease* in accuracy when moving from one to two to three collection intervals, using a four minute collection interval and a 20 minute look-back period. The observed degradation in accuracy between look-ahead periods between one and two collection intervals is half of that observed between look-ahead periods between two and three collection intervals. Therefore, this experiment will use two collection intervals (10 minutes) as its look-ahead period for predictions.

K-Folding and Datasets

An ensemble of models can produce predictions with a lower error than individual models. (Ensemble Learning, 2012) In addition, the disagreement between models' predictions can also be used to indicate how well the ensemble models the conditions it attempts to predict. (van Lint J. , 2006) An ensemble used in this experiment will be comprised of one model trained on each of the k-folds of the dataset. This experiment will use a k-value of 5, producing ensembles of 5 models for all 31 road segments to model, yielding 155 models. The experiment is performed with one ensemble trained on sensor data alone for a road segment and repeated using sensor data augmented with social data, which doubles the model count to 310.

K-fold cross-validation requires a training dataset (comprised of data from February 13, 2012 through March 13, 2012), which is partitioned into training segments 5 different times (one for each fold.) A disjoint dataset called the validation dataset is used to evaluate the ensemble after all its ANNs have been trained. The validation dataset is comprised of data from March 14, 2012 through March 21, 2012, which ensures that none of the ANNs in the ensemble have been trained or evaluated on the data.

Training Dataset

To partition the training dataset, each time interval is assigned five random numbers ranging from 0-1.0. The random numbers are sourced from www.random.org, to ensure that they do not suffer from computational pseudorandom bias and are uniformly distributed. (Haahr, 2012) The first random number for a time interval is its seed for fold1, the second random number for that interval is its seed for fold2, and so forth. Time intervals with seeds greater than 0.05 are selected to comprise the training segment in a fold. This attempts to use as much data as possible for training and holding back a representative amount for creating diversity in the training data and the ANNs trained on it.

Validation Dataset

The validation dataset is never used in training processes. It is a final scoring of the ANN ensemble that is used to determine its accuracy. This allows the experiment to simulate the performance of this technique as if it were deployed to be used by real vehicle operators while quantifying the error of its performance. This dataset is not partitioned – ensembles are evaluated against the entire validation dataset.

Operational Definitions of Independent Variables

This section describes how variables are translated into training and validation datasets for modeling. These variables represent input to the ANN that it will use to predict traffic speeds.

Time

Provided as multiple parts, to facilitate learning of repeating patterns, as depicted in Table 7 - Time Decomposition

Day of Week	Ranges from 0-6, with 0 indicating Sunday.
Hour of Day	Ranges from 0-23
Minutes Past the Hour	Ranges from 0-59 (chunked into 5-minute increments, due to the WSDOT data format. E.g.: 0, 5, 10, 15…50, 55.)

Table 7 - Time Decomposition

Previous Segment Speeds

The speeds measured from the previous 25 minutes (in 5 minute intervals.) They are each provided as a separate input, so this is represented as $S_{T-1}, S_{T-2}, \ldots\ S_{T-5}$, where T is a 5 minute interval.

Age of Last Traffic Impediment Broadcast

The number of minutes elapsed since the last tweet from the WSDOT about traffic in the subject road segment. This will range from 0 to 60 minutes – after 60 minutes have elapsed, it will be assumed that the tweet is no longer relevant.

Extraordinary Distraction

A value of zero or one, indicating that a tweet mentions an extraordinary distraction (1) or not (0). Note that none of the social data from the validation dataset contained this attribute, so there is no drilldown into its effect on prediction win probability.

Capacity Impact

An ordinal field indicating what portion of capacity is affected. This is a coarse measure, indicating whether zero, one, or multiple lanes are blocked. The range of values representing the aforementioned conditions is {0, 1, 2}, respectively.

Incident Type

A pair of boolean values, indicating whether the incident is a stall, a collision, or neither. The most common incident types broadcast by the WSDOT are stalls and collisions.

Impediment Direction

A value ranging from 0 to 1, indicating whether the impediment is in the same (1) or opposite (0) flow direction as the sensor. Note that this is specific to each sensor, so an example that has a value of zero for the direction in a dataset for a northbound sensor would have a value of one in the dataset for a southbound sensor.

Occupancy

This is a value ranging from 0 to 1, representing the number of vehicles present in the segment divided by the segment's vehicle capacity.

Operation Definition of Dependent Variable

The models attempt to predict a single outcome: the traffic speed ten minutes in the future. This is the sole dependent variable

Future Segment Speed

The predicted speed on the subject segment two collection intervals (10 minutes) in the future is represented as (S_{T+2})

Evaluation Criteria

The experiment deemed to have improved predictions if the error between speeds predicted by the ensemble and observed speeds decreases when models are trained with social data.

Ensemble Prediction

The ensemble's prediction is calculated as the mean of all K-folds:

$$Ensemble\ Prediction = \frac{\sum_{i=1}^{k} Prediction_i}{k}$$

Ensemble Disagreement

The standard deviation of the ANNs' predictions will be considered the disagreement of the ensemble for each prediction. Ensemble disagreement is not used to determine success or failure of a prediction but is used to suggest relative confidence across predictions. This is calculated as the standard deviation of prediction error across the K-folds in an ensemble. Ensemble aggregation is calculated on each example in the validation set. However, the mean and standard deviation of values observed across the validation dataset must be calculated in order to indicate the relative confidence of an individual prediction.

$$Ensemble\ Disagreement = \sqrt{\frac{\sum_{i=1}^{k}(Prediction_i)^2}{k} - \left(\frac{\sum_{i=1}^{k}(Prediction_i)}{k}\right)^2}$$

Error Calculation

The experiment is determined to have improved predictions if the root mean square error between speeds predicted by the ensemble and observed speeds decreases when models are trained with social data. RMS error is the standard measure for experiments that test ANN accuracy. This is calculated across the entire validation dataset, containing N examples:

$$RMS\ Error = \sqrt{\frac{\sum_{i=1}^{n}(Ensemble\ Prediction_i - Observation_i)^2}{n}}$$

Traffic prediction experiments have also used the Mean Absolute Percentage Error for quantifying results. Whereas RMS error optimizes for the best aggregate fit of the data, MAPE optimizes for higher accuracy at low speeds. The ANNs are trained to optimize their RMS error but MAPE is calculated to depict results in the same language used by other studies. MAPE also anomalous when observed speeds approach zero, as dividing the prediction error by a tiny number yields a disproportionately high absolute percentage error. To address that issue, this study's calculations use the greater of Observation $_i$ or 1.0.

$$MAPE = \frac{\sum_{i=1}^{n}\left(\left|\frac{Ensemble\ Prediction_i - Observation_i}{Observation_i}\right|\right)}{n}$$

Theoretical Route Travel Time

This experiment considers each road segment as a point along a route heading either north or south on a roadway. In order to evaluate the combined effect of sensor predictions vs. observed results in a way that is meaningful to vehicle operators, a theoretical route travel time is calculated. This route assumes that at a given point in time, the theoretical travel time along the route can be calculated by assuming a vehicle can travel at the speed observed or predicted at a sensor until it reaches the next sensor in the route, where its speed is defined by the second sensor's observation or prediction. This continues until the user reaches the milepost that defines the boundary of the roadway that is being evaluated.

$$TravelTime_{Theoretical} = \sum_{n=1}^{Last\ Sensor} \frac{Distance\ to\ Sensor_{n+1}}{Speed\ of\ Sensor_n}$$

This travel time is purely theoretical, as it does not account for the time that elapses while in transit between sensors, where the speeds may have changed. Traffic maps depicting current conditions provide exactly this kind of data but quantifying differences between them is difficult. Instead, comparing theoretical travel time is a good alternative.

ANN Architecture and Training Configuration

ANNs are all trained using the Multiple Back-Propagation tool version 2.2.4, from Noel de Jesus Mendonca Lopes (http://dit.ipg.pt/MBP/).

The literature indicated that there is no heuristic for determining the optimal configuration for training ANNs a-priori. It is recommended that many configurations be attempted and those yielding the best results on a particular dataset be used. The first fold of Sensor Id 38's training data was used to test various combinations of learning rate, momentum, and training epoch counts to determine which yielded the best results. This was done once for the training set with social data and once for the training set without social data.

To test the parameters, an ANN is trained with combination of learning rate and momentum settings for 100 epochs. The error on the training data and the evaluation data is noted and then the ANN is trained for an additional 150 epochs, for a total of 250. Error is noted for this configuration, and the process is repeated, testing epochs 500, 1,000, 2,500, 5,000, up to 10,000, 20,000, or 50000, if the results continue improving. This test is performed for all combinations of learning rate and momentum, where values are in the set {0.3,0 .5, 0.7 }. The results are included in Table 30 - ANN Training without Social Signals and Table 31 - ANN Training with Social Signals, located in Appendix 2 – Testing Training Parameter Effects on Error.

Training an ANN over more epochs requires more time to train it. Given that this experiment requires the training of 310 ANNs, consideration must be made regarding how much burden training a greater number of epochs incurs against the significance of the potential increase in accuracy. To aid in this decision, tables of training parameters and errors are summarized by the minimum error produced at each number of training epochs. Based on the table below, there is no significant difference in the error observed when training for over 1000 epochs. This is likely caused by the ability of the ANNs to quickly achieve a nominal fit of the training data due to the aggressive initial learning rates. The experimental epoch count was decided by picking the middle values (5,000 and 10,000) and breaking the tie by selecting the higher number of epochs (10,000), as it appeared to be one of the better options for the models trained with social data, as shown below. This corresponded to a learning rate of 0.3 and a momentum of 0.5.

Training Epochs	Min Error
1000	0.0647
2500	0.0648
5000	0.0648
10000	0.0648
20000	0.0647
50000	0.06462

Table 8- Minimum Error Values by Epoch for Non-Social Data. Note the minute differences in the error across the table. A .0001 error corresponds to roughly 0.01MPH, which is not important.

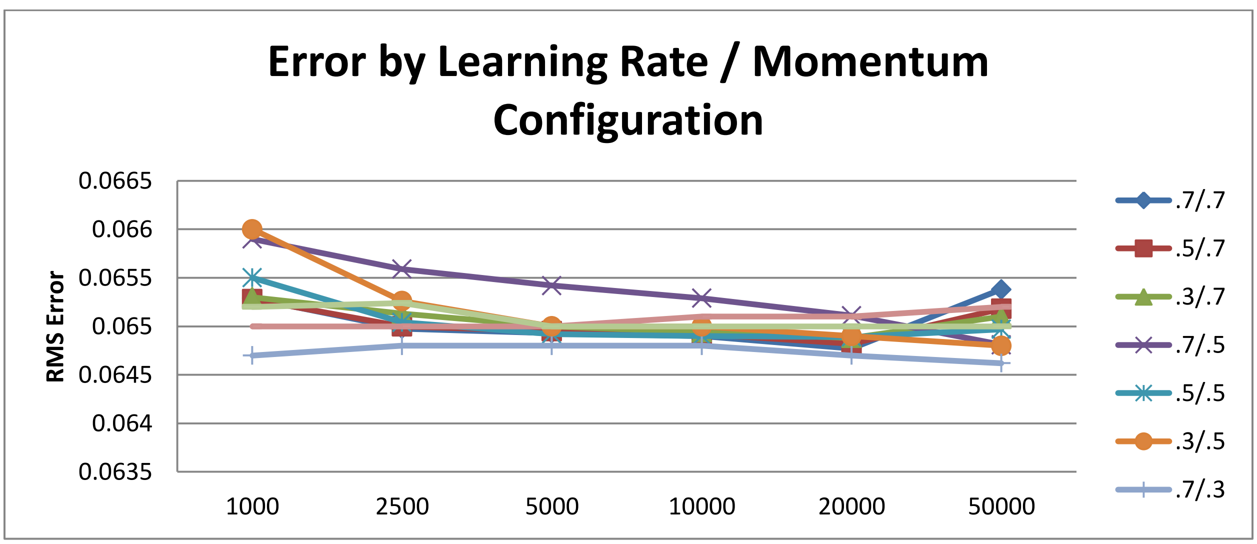


Table 9 - Minimum Error Values by Epoch for Non-Social Data

The smallest error in the social training dataset was observed when trained for 20,000 epochs. However, this is not significantly better than the error observed when training for 10,000 epochs. In order to reduce the burden of training the social ANNs, the experiment trained these ANNs for 10,000 epochs. This error was produced with a learning rate and momentum of .3 and .5, respectively. See Table 10 - Minimum Error Values by Epoch for Social Data

Training Epochs	Min Error
1000	0.0628
2500	0.0625
5000	0.0621
10000	0.0619
20000	0.0617
50000	0.0617

Table 10 - Minimum Error Values by Epoch for Social Data

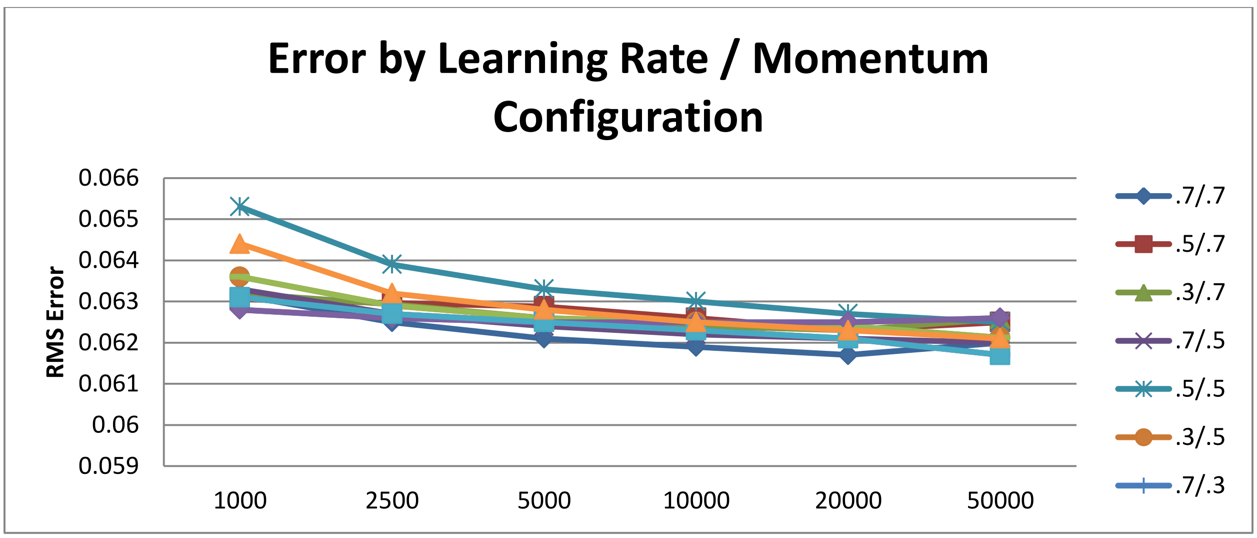


Table 11 - Minimum Error Values by Epoch for Social Data

Terms and Definitions

Collection Interval – the amount of time (in minutes) that each data sample represents.

Training Dataset – the dataset used to train the ANNs to model future traffic speeds.

Validation Dataset – the dataset used to evaluate the ANNs' predictions. This data is never used to train ANNs and is disjoint from the Training Dataset.

Look-ahead Period – the amount of time between the collection of a sample and the expected realization of a prediction. This is described in minutes as well as a multiple of collection intervals. This describes how far in the future the model is trying to predict conditions.

Look-back Period – the amount of time between the collection of a sample and the previous samples, where it can be assumed that older samples have insignificant contributions to predicting future conditions. This describes how far in the past the model is considering samples to predict future conditions.

Tweet – a message broadcast on the Twitter social network

Chapter III – Experimental Results

This section describes the details of the experiment, teasing out statistics about the input data, presenting the aggregate results, and drilling into the details.

Training and Validation Data Characteristics

Speed data observed across the system is subject to significant variation. However, the validation and training datasets exhibit similar characteristics, with their mean and standard deviation differing only slightly. Note that both sets have significant numbers of invalid examples that needed to be excluded from experiments. Examples are flagged as suspect or invalid in the WSDOT dataset when their sensor is unresponsive sensors or sending invalid data.

Statistic	Training Dataset	Validation Dataset
Mean	53.95 MPH	53.25 MPH
St. Dev	20.09 MPH	19.77 MPH
Valid Example Count	147028	35500
Invalid Example Count	72078	19566
Min Value	0	0
Max Value	100	100

Table 12 - Training and Validation Dataset – Sensor Statistics

Each sensor's training and validation sets can display different characteristics, depending on how often the sensor returned invalid data. Most sensors' distribution was similar to Sensor 38, shown in Table 13 – Dataset Breakdown for a Typical Sensor (Sensor 38). The speeds are skewed toward the maximum values, with the mean ending up near the first quartile (Q1) value and a mode near the third quartile value (Q3.)

Sensor 38
Training Dataset Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	66.32	66.34	66.34	66.33	66.32

StDev	3.46	3.40	3.35	3.44	3.38
Mode	67	67	67	67	67
Example Count	5053	5054	5059	5072	5057
Min	15	15	15	15	15
Max	72	72	72	72	72
Q1	66	66	66	66	66
Median	67	67	67	67	67
Q3	68	68	68	68	68

Training Dataset With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	66.32	66.34	66.34	66.33	66.32
StDev	3.26	3.40	3.34	3.43	3.38
Mode	67	67	67	67	67
Example Count	6026	5072	5077	5090	5075
Min	15	15	15	15	15
Max	72	72	72	72	72
Q1	66	66	66	66	66
Median	67	67	67	67	67
Q3	68	68	68	68	68

Validation	Without Social Data	With Social Data
Mean	65.98	65.53
StDev	4.33	5.51
Mode	67	67
Example Count	1298	1573
Min	22	22
Max	73	73
Q1	65	65
Median	67	67
Q3	68	68

Table 13 – Dataset Breakdown for a Typical Sensor (Sensor 38)

Sensors 161 and 242 are notable exceptions to this pattern. They both report a much lower mean speed and a mode of zero. See Table 14 - Anomalous Sensor Data (Sensor 161). This characteristic is present in both the training and validation sets, so it is unlikely to be an error. The number of examples in these sets is comparable to the others, so it is also unlikely that the statistics are anomalous due to an excessively small sample size.

Sensor 161

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	24.85	24.95	25.00	24.91	24.97
StDev	22.15	22.16	22.12	22.13	22.13
Mode	0	0	0	0	0
Example Count	7530	7514	7541	7520	7521
Min	0	0	0	0	0
Max	87	87	87	87	87
Q1	0	0	0	0	0
Median	27	27	28	27	28
Q3	47	48	48	48	48

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	25.13	24.96	25.00	24.91	24.98
StDev	21.54	22.14	22.11	22.12	22.12
Mode	0	0	0	0	0
Example Count	8399	7532	7559	7538	7539
Min	0	0	0	0	0
Max	87	87	87	87	87
Q1	0	0	0	0	0
Median	26	27	27	27	27
Q3	47	48	48	48	47

Validation	Without Social Data	With Social Data
Mean	22.91	23.10
StDev	21.33	20.76
Mode	0	0
Example Count	1900	2134
Min	0	0
Max	61	61
Q1	0	0
Median	22	23
Q3	46	46

Table 14 - Anomalous Sensor Data (Sensor 161)

Tweet Characteristics

The ontological characteristics of traffic incident tweets from the @WSDOT_Traffic account are present with the characteristics detailed in Table 15 - Training and Validation Datasets – Social Statistics. Comparing the training and validation datasets, the breakdown of the attribute occurrence is very similar for Capacity Impact. The balance of disabled vehicles vs. collisions shifted but they were both in the high-30% to 50% range. The incidents mostly occurred in the southern half of the roadway examined.

Statistic	Training Dataset	Validation Dataset
Example Count	248	74
Capacity Impact Instances	190 (77%)	56 (76%)
Capacity Impact = 1	167 (67%)	49 (66%)
Capacity Impact = 2	23 (9%)	7 (9%)
Disabled Vehicle Instances	91 (37%)	38 (51%)
Collision Instances	99 (40%)	28 (38%)
Extraordinary Distraction Instances	9 (4%)	0 (0%)
Mean Incident Milepost	166.99	165.03
StDev Incident Milepost	3.79	4.11

Table 15 - Training and Validation Datasets – Social Statistics

Multiple tweets were reported for most incidents, as shown in Table 16 - Tweets Broadcast per Incident.

	Training	Validation
Min	1	1
Max	7	4
Mean	1.35	1.28
St Dev	0.93	0.64

Table 16 - Tweets Broadcast per Incident

Tweets are not uniformly distributed by location and have a different distribution in the validation dataset than in the training dataset. See Table 17 - Tweet Distribution in Training and Validation Datasets and Figure 7 - Tweet Distribution in Training and Validation Datasets.

Milepost	Training Tweet Count	Evaluation Tweet Count
157	6	5
158	4	5
161	4	6
163	30	8
164	36	5
165	10	5
166	21	8
167	22	17
168	22	0
169	14	9
170	30	1
171	18	1
172	19	0
173	5	3
174	7	1

Table 17 - Tweet Distribution in Training and Validation Datasets

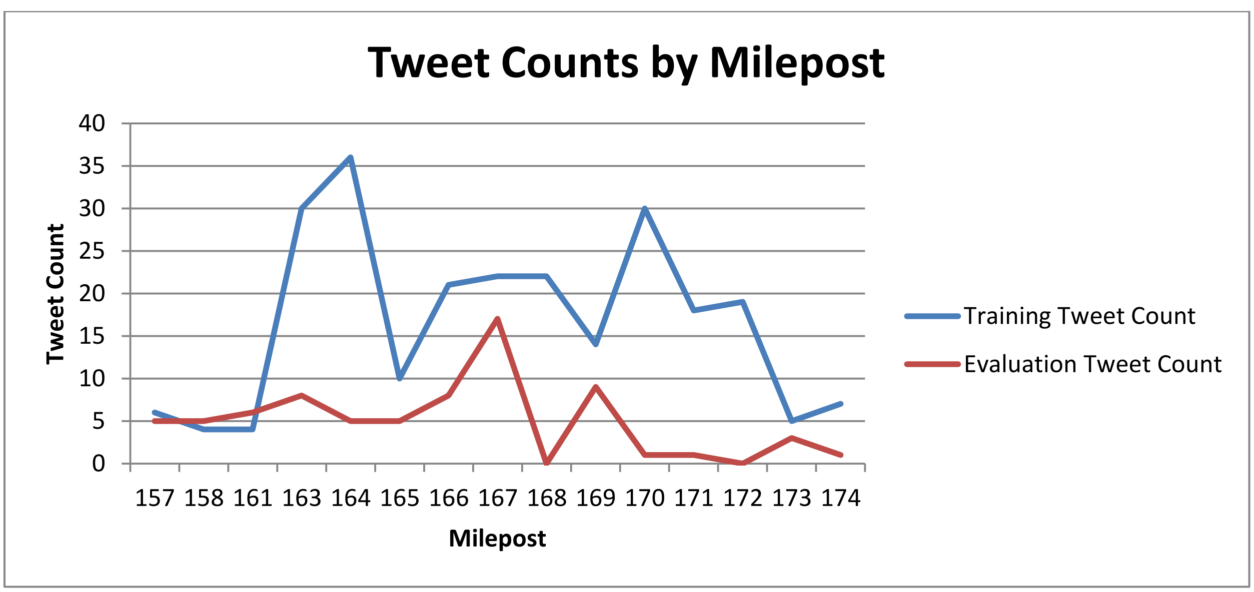


Figure 7 - Tweet Distribution in Training and Validation Datasets

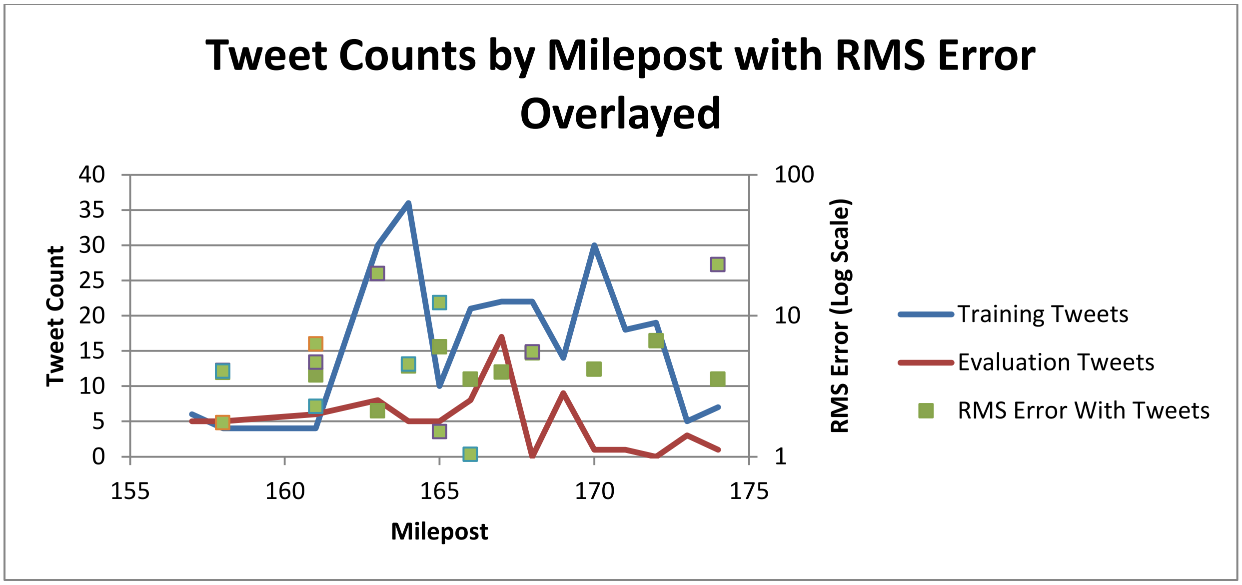


Figure 8 - Tweet Counts by Milepost with Sensor Error Overlayed. There doesn't appear to be a correlation between tweet counts and error magnitude.

Tweets are quite pervasive when matched with the sensor data, as roughly a third of data points in both training and validation contain a tweet. Examples with tweets show speeds that are slightly less skewed toward the max observed values than those found in the general training and validation datasets.

Examples	1210
Examples containing a Tweet	219
Probability an example containing of Tweet	0.18
Min Speed w/ Tweet	12
First Quartile Speed w/ Tweet	18
Median	25
Third Quartile	63
Max Speed w/ Tweet	72
Mean	35.56621005
Standard Deviation	20.78195015

Table 18 - Sensor 206, Fold 1 Training Set Tweet Breakdown.

Examples	377
Examples containing a Tweet	125
Probability an example containing of Tweet	0.331565
Min Speed w/ Tweet	14
First Quartile Speed w/ Tweet	23
Median	30
Third Quartile	35
Max Speed w/ Tweet	69
Mean	50.9955
Standard Deviation	18.42876

Table 19 - Sensor 206, Fold 1 Validation Set Tweet Breakdown. Roughly a third of examples have tweets associated with them.

Aggregate ANN Prediction Performance

The performance for each ANN ensemble is listed below. Two of the sensors were not reporting data (Sensor 141 and 233), so they do not have predictions. A decrease in the RMS prediction error observed in ANNs using social signals is considered a win. There are 8 instances of social wins (noted with an asterisk * in

Sensor ID	Sensor Data			Sensor Data w/ Social Signals		
	RMS Error (MPH)	Disagreement Mean	Disagreement St Dev	RMS Error (MPH)	Disagreement Mean	Disagreement St Dev
38	2.41	0.27	0.28	3.95	0.72	0.86
39	2.89	0.25	0.21	4.08	0.47	0.75
49	3.35	0.31	0.35	4.04	0.74	1.14
* 72	1.78	0.22	0.13	1.74	0.62	0.59
77	3.52	0.35	0.38	3.80	1.00	1.25
80	4.45	0.32	0.36	4.67	1.11	1.21
* 87	2.37	0.20	0.24	2.28	0.67	0.75
91	6.06	0.37	0.46	6.30	1.07	1.37
94	6.46	0.44	0.36	6.74	0.95	1.23
* 98	19.01	0.98	0.55	17.99	2.15	1.77
105	4.92	0.34	0.21	5.70	1.17	1.28
108	2.92	0.31	0.20	2.94	0.85	1.57

109	2.85	0.29	0.23	2.90	0.60	0.97
* 111	14.21	0.59	0.56	13.84	1.17	1.01
118	2.11	0.11	0.05	2.11	0.21	0.44
* 119	19.96	0.90	0.60	19.92	1.69	1.75
* 132	4.16	0.40	0.22	4.39	0.83	0.72
141	No Data			No Data		
149	4.15	0.35	0.26	4.53	1.01	1.04
151	5.53	0.44	0.34	6.02	1.31	1.38
155	1.47	0.23	0.13	1.50	0.59	1.25
* 161	12.38	0.75	0.50	12.36	1.51	1.38
168	3.42	0.38	0.36	3.54	0.70	0.98
169	3.78	0.32	0.27	3.97	0.77	0.70
176	4.88	0.42	0.32	5.44	0.99	1.15
179	4.78	0.37	0.31	5.52	0.69	0.90
200	3.63	0.25	0.32	4.16	0.56	0.81
206	6.08	0.47	0.36	6.63	1.24	1.42
233	No Data			No Data		
239	3.49	0.35	0.26	3.55	1.06	1.58
* 242	23.87	1.38	0.83	23.00	2.10	1.50

Table 20 - RMS Errors Observed in Predictions) where the aggregate RMS error of predictions products by ANNs with social signals is lower than the ANN trained without social signals, as shown in the data below.

Sensor ID	Sensor Data			Sensor Data w/ Social Signals		
	RMS Error (MPH)	Disagreement Mean	Disagreement St Dev	RMS Error (MPH)	Disagreement Mean	Disagreement St Dev
38	2.41	0.27	0.28	3.95	0.72	0.86
39	2.89	0.25	0.21	4.08	0.47	0.75
49	3.35	0.31	0.35	4.04	0.74	1.14
* 72	1.78	0.22	0.13	1.74	0.62	0.59
77	3.52	0.35	0.38	3.80	1.00	1.25
80	4.45	0.32	0.36	4.67	1.11	1.21
* 87	2.37	0.20	0.24	2.28	0.67	0.75
91	6.06	0.37	0.46	6.30	1.07	1.37
94	6.46	0.44	0.36	6.74	0.95	1.23

* 98	19.01	0.98	0.55	17.99	2.15	1.77
105	4.92	0.34	0.21	5.70	1.17	1.28
108	2.92	0.31	0.20	2.94	0.85	1.57
109	2.85	0.29	0.23	2.90	0.60	0.97
* 111	14.21	0.59	0.56	13.84	1.17	1.01
118	2.11	0.11	0.05	2.11	0.21	0.44
* 119	19.96	0.90	0.60	19.92	1.69	1.75
* 132	4.16	0.40	0.22	4.39	0.83	0.72
141	No Data			No Data		
149	4.15	0.35	0.26	4.53	1.01	1.04
151	5.53	0.44	0.34	6.02	1.31	1.38
155	1.47	0.23	0.13	1.50	0.59	1.25
* 161	12.38	0.75	0.50	12.36	1.51	1.38
168	3.42	0.38	0.36	3.54	0.70	0.98
169	3.78	0.32	0.27	3.97	0.77	0.70
176	4.88	0.42	0.32	5.44	0.99	1.15
179	4.78	0.37	0.31	5.52	0.69	0.90
200	3.63	0.25	0.32	4.16	0.56	0.81
206	6.08	0.47	0.36	6.63	1.24	1.42
233	No Data			No Data		
239	3.49	0.35	0.26	3.55	1.06	1.58
* 242	23.87	1.38	0.83	23.00	2.10	1.50

Table 20 - RMS Errors Observed in Predictions

Win/Loss Comparison

To assess the aggregate performance of each social ANN, the probability of a loss is calculated. This is the probability that the social ANN will generate a prediction with a smaller error than the ANN trained with sensor data alone. Probabilities near .5 indicate no noticeable effect, as that’s equivalent to a coin toss for determining which is more accurate. Higher probabilities mean a greater chance of a social ANN out-performing the sensor-based ANN and conversely, lower probabilities indicate that the social ANN has a greater chance of being less accurate than the sensor-based ANN.

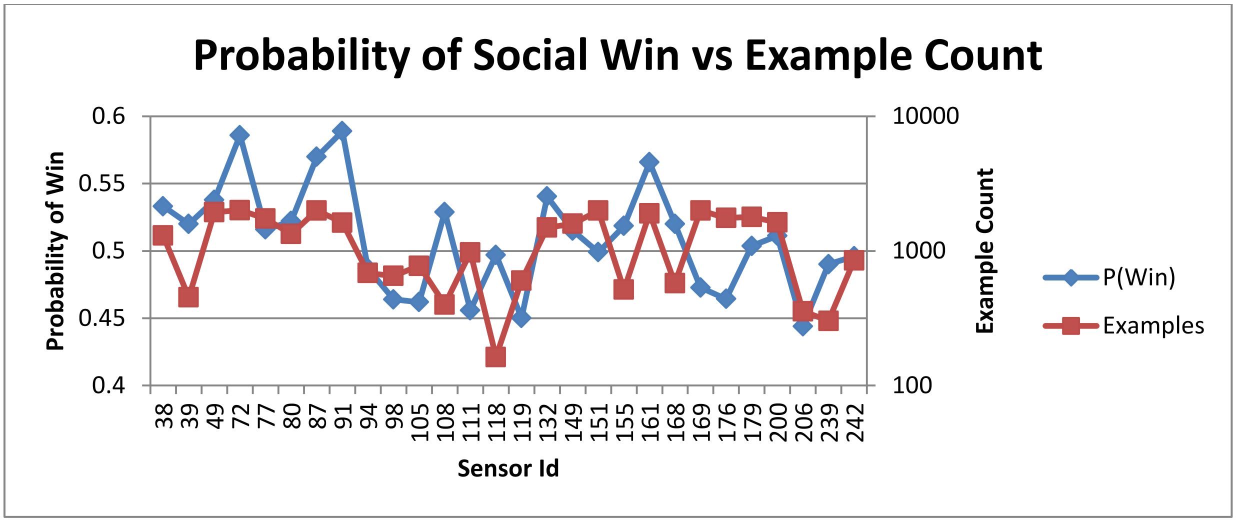


Figure 9 - Social win probability by Sensor. Example count is also shown

Some of the sensors that had lower RMS error with Social signals, such as 72, 87, 91, and 161 also had a higher probability of a win. Some sensors that did not show an improvement in RMS error had a higher probability of winning with social signals, as observed with sensor 108. The latter indicates a predictor that can be relied on in a greater number of instances but when wrong, has significantly greater error than the model without social signals. To determine which factors were important in producing a win or loss for each experiment with social data, sensor performance is examined against social attributes. To keep visuals clean, only social wins are graphed in this section. The complete analysis is located in Appendix 4 – Detailed Result Breakdown.

Capacity Impact

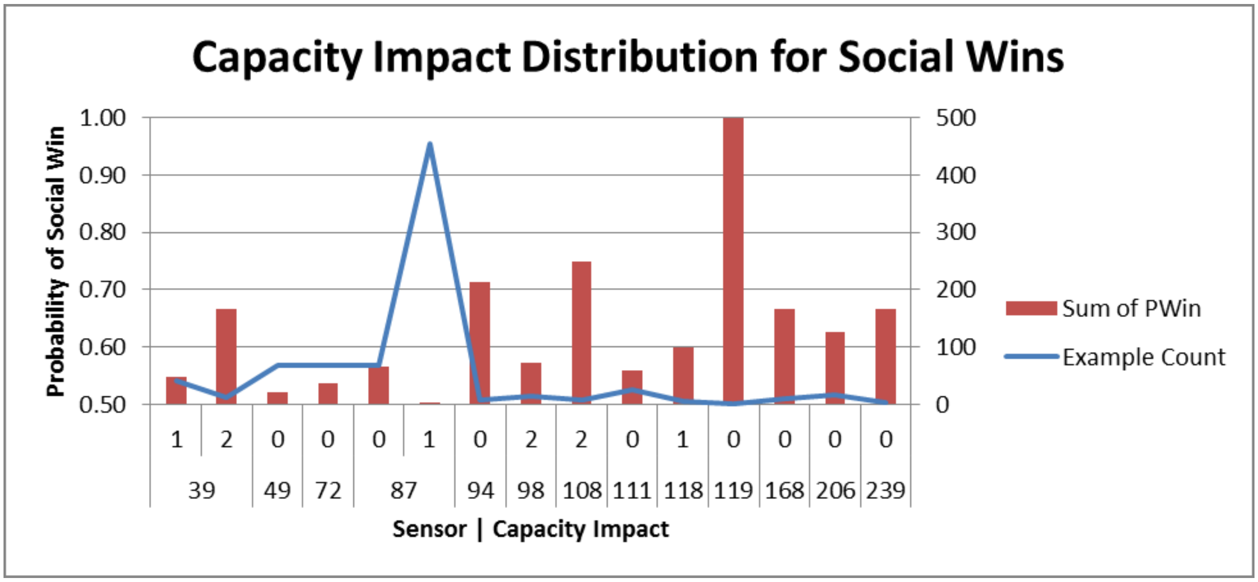


Figure 10 - Capacity Impact Distribution for Social Wins

The chart of wins does not indicate any single impact level where social signals clearly show an improvement. Most of the wins occur for scenarios where there are very few examples. This raises the question of whether wins are connected with low example counts. Overall, this is not the case, as depicted in Figure 11 - Social Win Probability vs. Capacity Impact for Low Example Counts. There are thirty one losses (PWin < .5) and only thirteen wins (PWin > .5) as shown in Table 21 - Low Example Count Win Loss Counts. These drilldowns examine cases where the example counts are less than fifty.

Capacity Impact	Losses	Wins
0	6	6
1	3	3
2	22	4
Total	31	13

Table 21 - Low Example Count Win Loss Counts

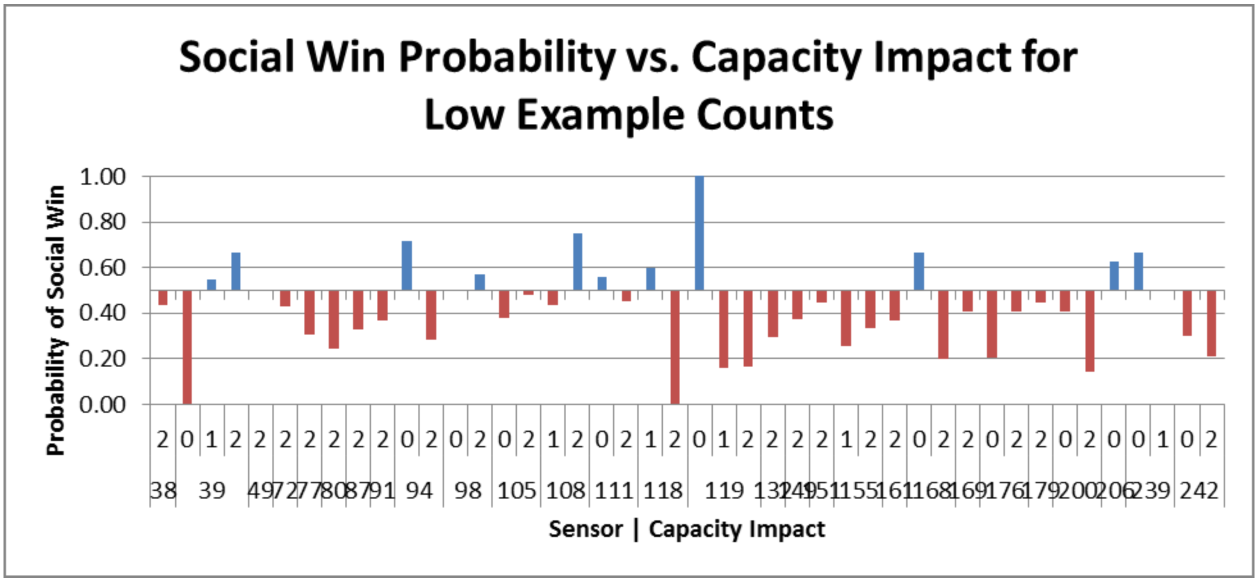
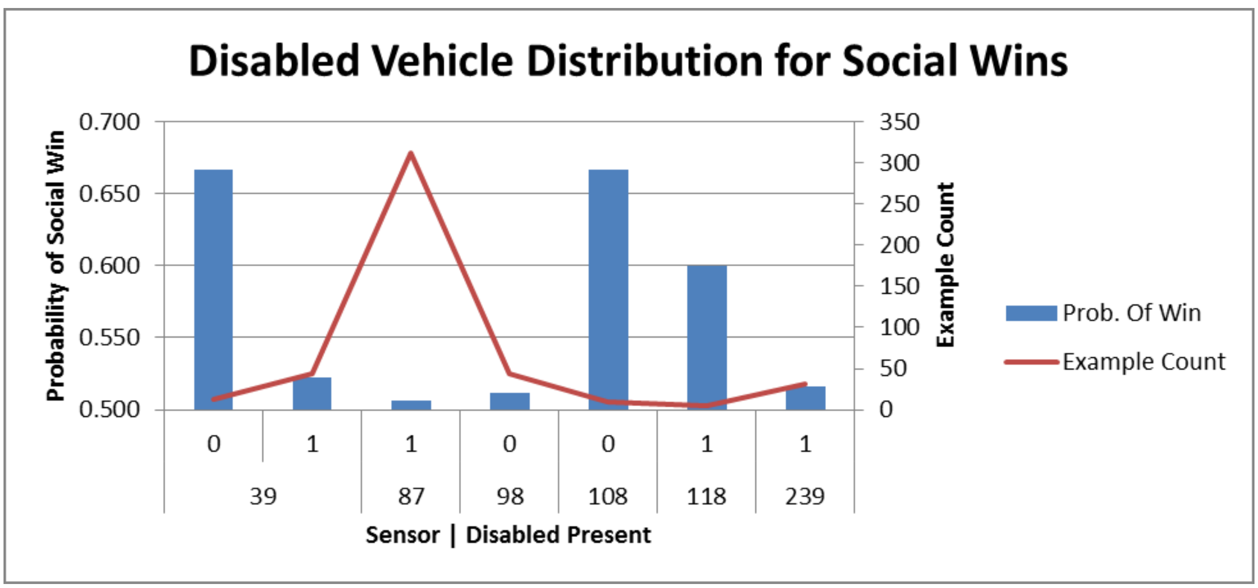


Figure 11 - Social Win Probability vs. Capacity Impact for Low Example Counts

Disabled Vehicles and Collisions



The most significant win probability, with respect to the presence of disabled vehicles, came in instances where a tweet did not contain a report of a disabled vehicle (indicated by a zero value in the chart.) It is worth reiterating that tweets can contain information about a collision, disabled vehicle, or neither. The latter case corresponds with tweets mentioning slow traffic without specifying a particular cause. This case also reports significant wins when data is scarce, so it is worth looking at a win/loss breakdown when the number of examples is less than fifty.

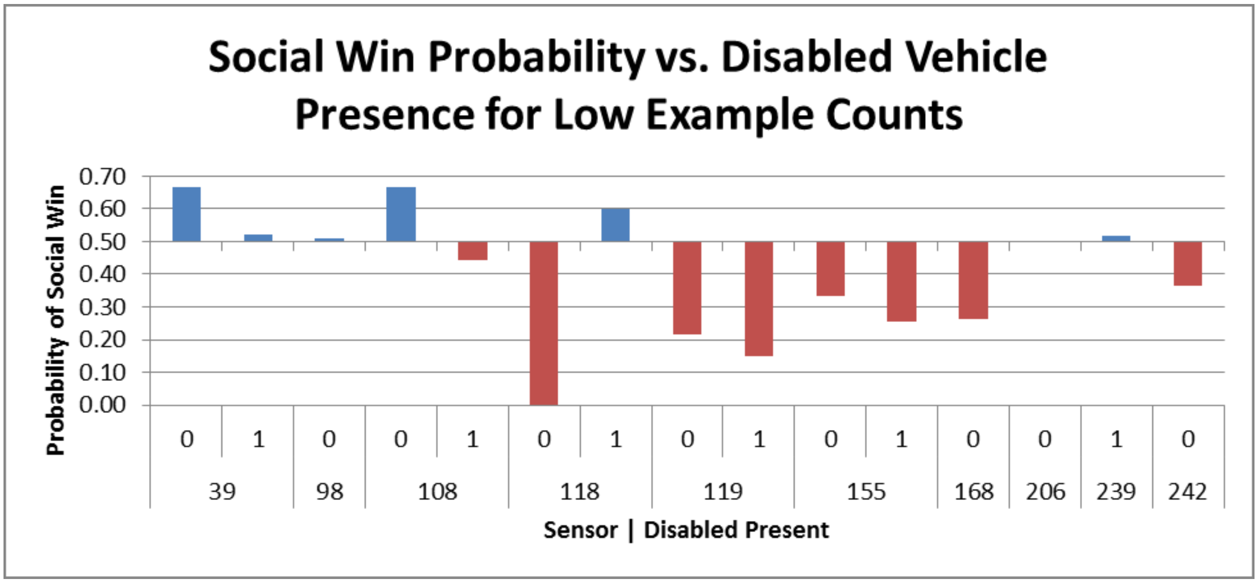


Figure 12 - Social Win Probability vs. Disabled Vehicle Presence for Low Example Counts

In this case, there were six social wins vs. eight social losses, so low example counts don't necessarily precede a social win.

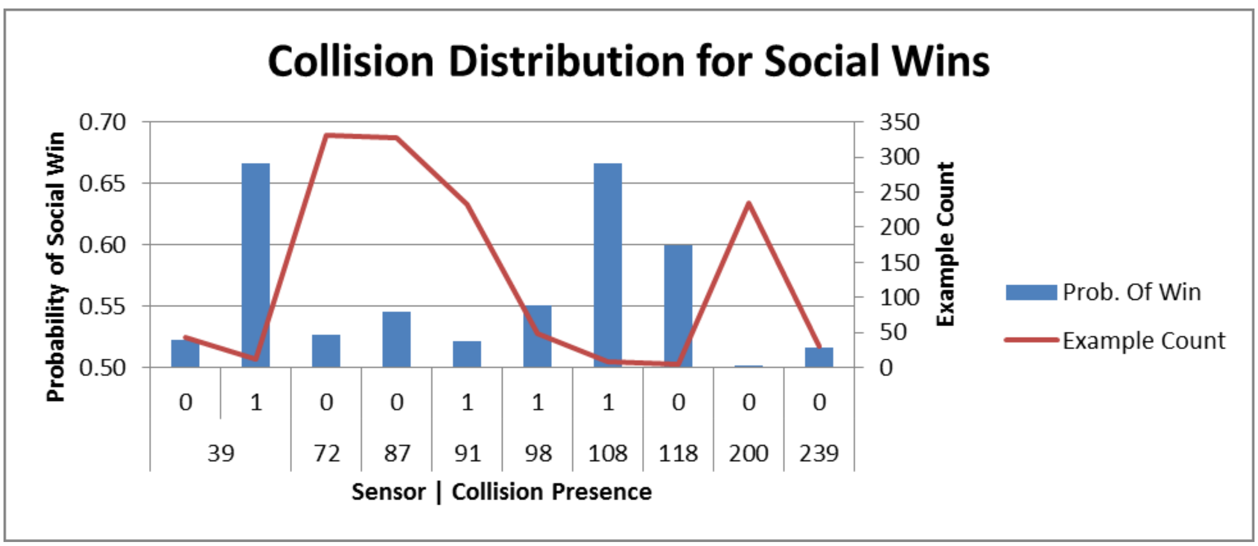


Figure 13 - Collision Distribution for Social Wins

The sensors showing the largest gains had low example counts again. As in the previous analysis, it raises the question of whether scarce data correlates with wins in this case. When examining this data slice, there were six social wins and ten losses for instances of fewer than fifty examples, as shown in Figure 14 - Social Win Probability vs. Collision Presence for Low Example Counts.

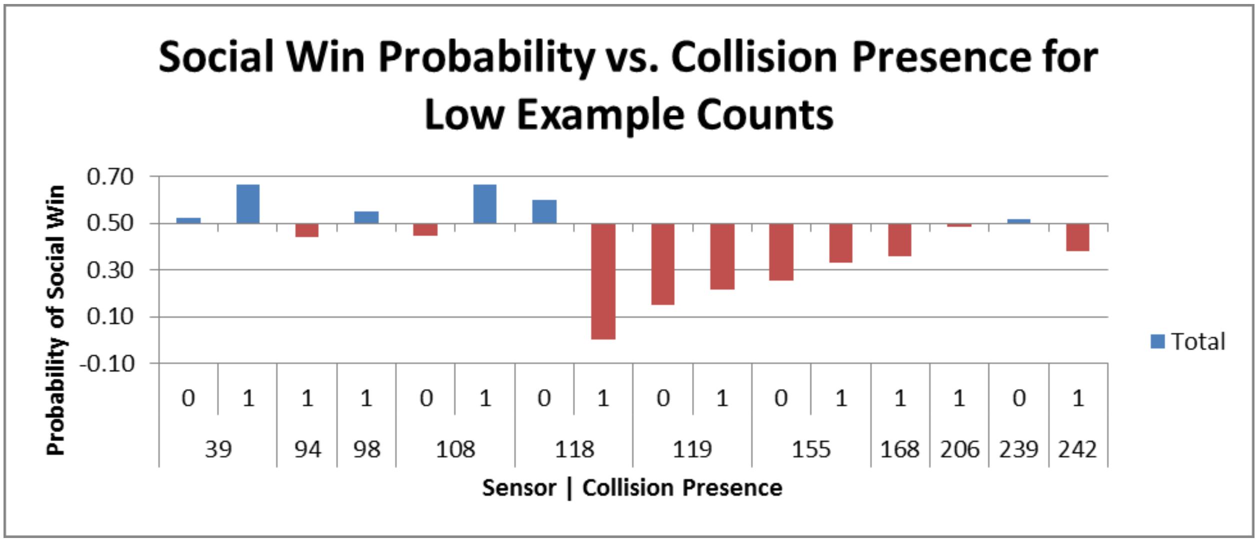


Figure 14 - Social Win Probability vs. Collision Presence for Low Example Counts

Direction

Direction is a little more challenging to dissect, due to the way it was modeled in this experiment. Recall that multiple tweets during a single time sample are provided as separate examples during training and validation. However, when aggregating the data to determine a win or loss for a sample, they must be combined. When multiple tweets are present for a sample, the presence of a collision or disabled vehicle is described as the maximum value among all tweets. However, tweets may describe incidents affecting the opposite lane, the same lane, or a combination of both. To aggregate the direction of active incidents, the mean of direction is used. Therefore, a value of zero indicates that all tweeted incidents are in the opposite direction. A value of one indicates that all tweeted incidents are in the same direction. Values between one and zero indicate some combination of the two. The three categories are depicted in Figure 15 - Social Wins When All Incidents are in the Opposite Direction, Figure 16 - Social Wins When All Incidents are in the Same Direction, and Figure 17 - Social Wins When All Incidents are in a Combination of Directions below.

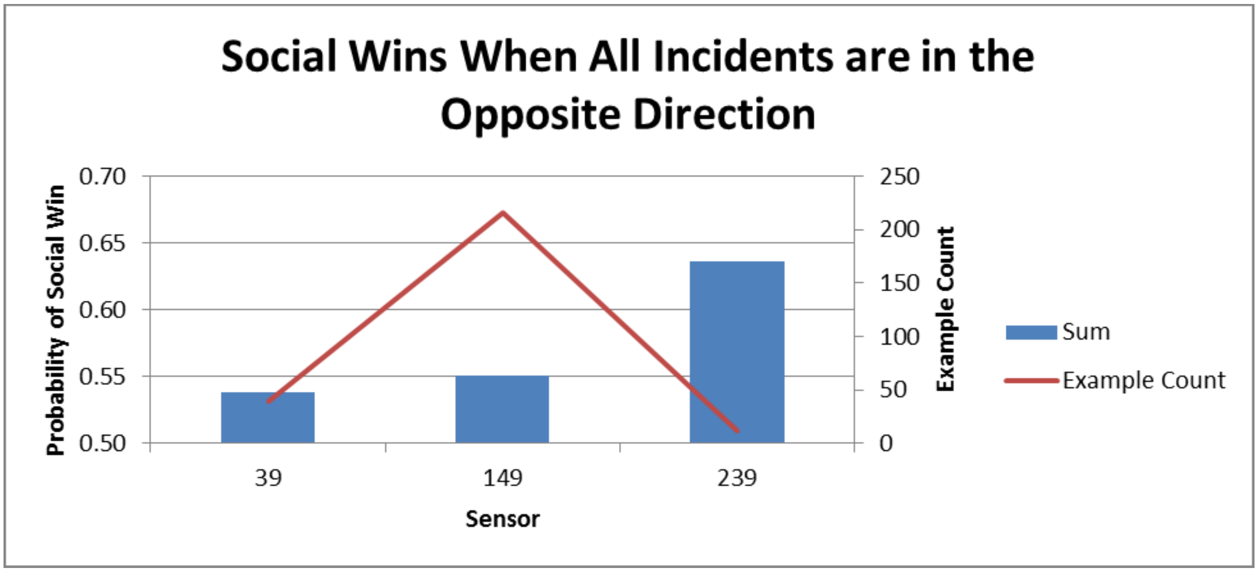


Figure 15 - Social Wins When All Incidents are in the Opposite Direction

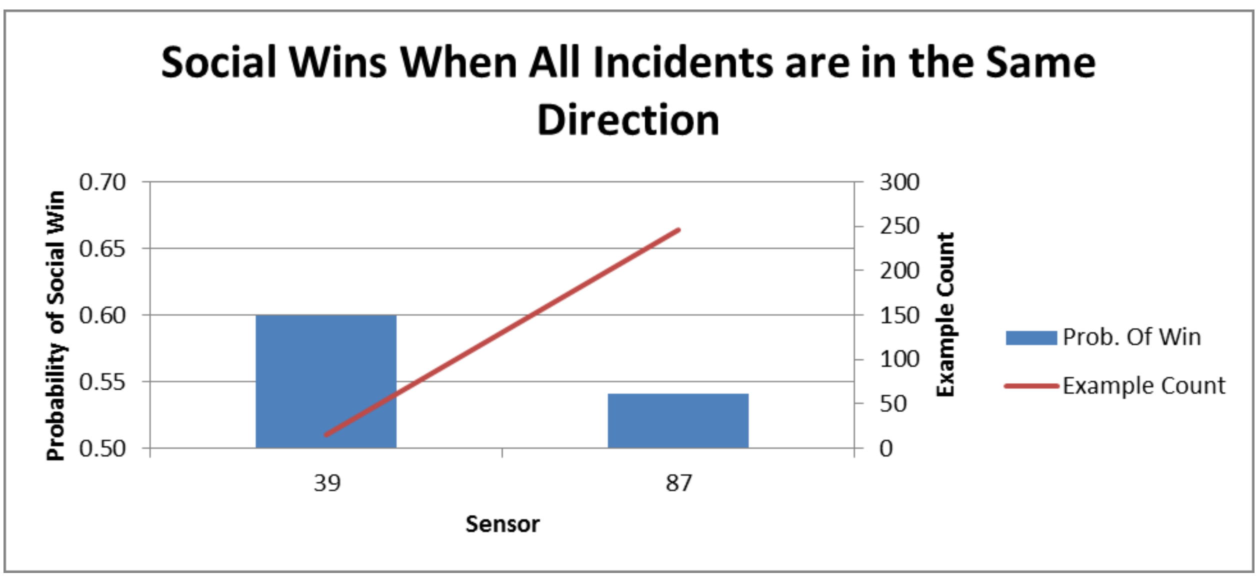


Figure 16 - Social Wins When All Incidents are in the Same Direction

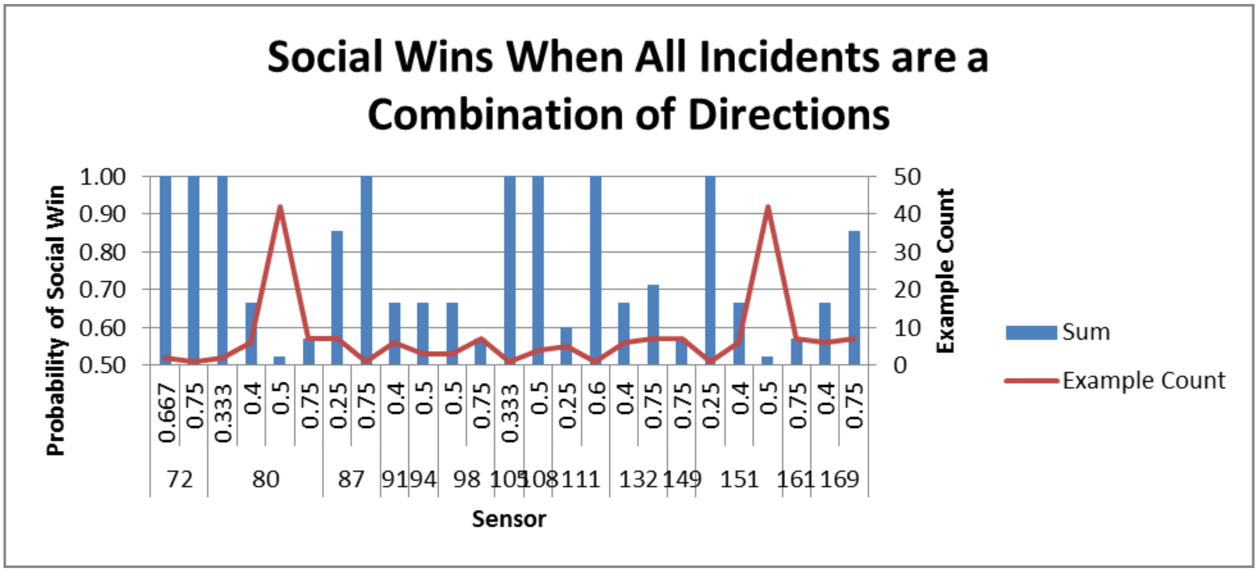


Figure 17 - Social Wins When All Incidents are in a Combination of Directions

There are several interesting facets to point out about this drill down: the majority of wins occur when there are multiple tweets, without any particular pattern (this is expected, as direction is a weaker signal when aggregated like this.) The sensors that show a correlation with a particular direction (such as sensor 39, where it only recorded wins when incidents were either in the same or opposite directions but not a combination of both) can indicate that the way multiple incidents are modeled makes their impact difficult to learn. As in previous analyses, low example counts are a threat to external validity. In this case, there are sufficient examples for sensors 39 and 149 to consider the effect a real observation, using the statistical rule-of-thumb that considers thirty samples the point at which there is critical mass.

Sensor	Prob. Of Win	Example Count
39.00	0.54	39
149.00	0.55	216
239.00	0.64	11

Table 22 - Social Wins When All Incidents are in the Opposite Direction

On the other end of the spectrum, only sensor 87 showed a meaningful correlation with incident direction.

Sensor	Prob. Of Win	Example Count
39.00	0.60	15
87.00	0.54	246

Table 23 - Social Wins for Sensors 39 and 87 When Incidents are in the Same Direction

Distance to Incident

The nearest distance to an incident is the most relevant way to aggregate multiple distances when comparing performance. This results in a much simpler analysis but it still requires stratification, as the distance from each sensor to a single incident will be different, which makes distinct distance buckets uninteresting. The observed wins have the distribution depicted in Table 24 - Distribution of Social Wins by Distance, so stratifying distances into buckets where the distance was less than five miles, five to ten miles, and greater than ten miles should be reasonable. There were many instances of sensor/bucket combinations having very small example counts, so the chart below only includes those with thirty or more examples (Figure 18 - Social Win Probability vs. Stratified Distance to Incident.)

Metric	Value
Mean	5.92
StDev	4.01
Min	0.03
Max	15.79

Table 24 - Distribution of Social Wins by Distance

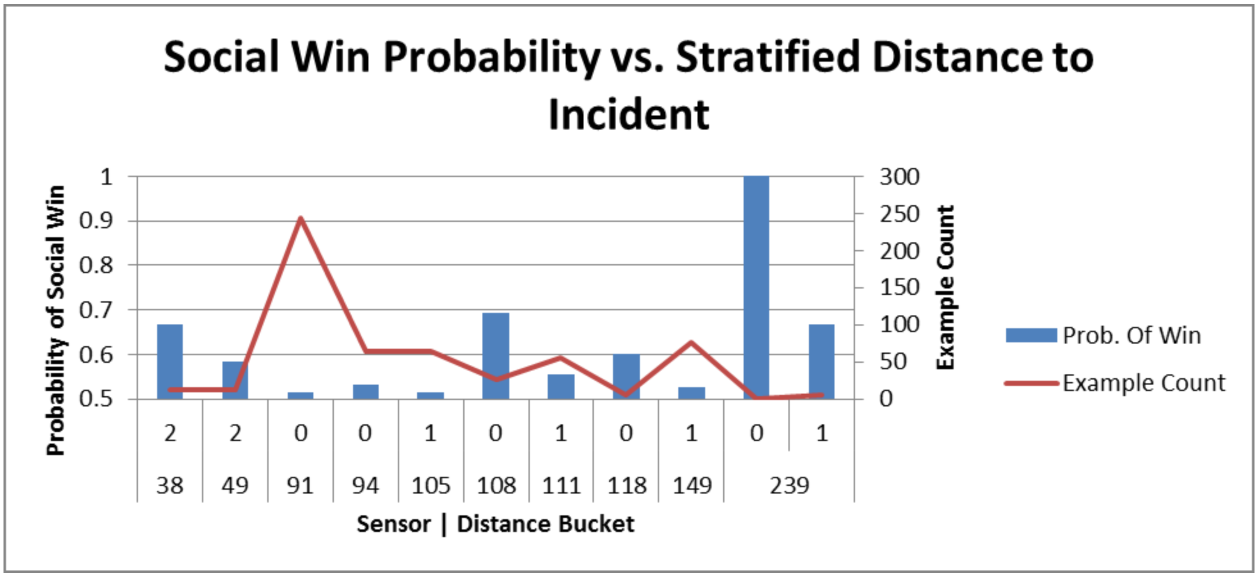


Figure 18 - Social Win Probability vs. Stratified Distance to Incident

Some of the more unexpected results, such as sensors 38 and 49 resulting in wins when incidents are reported more than ten miles away, occur when there are very few examples, as shown in Table 25 - Social Win Probability vs. Distance Buckets. The more interesting results are those shown by sensor 94and sensor 111, as they have both a significant number of examples and show an interesting win probability when incidents fall into buckets zero and one, respectively.

Sensor / Bucket	Prob. Of Win	Example Count
38		
2	0.667	12
49		
2	0.583	12
91		
0	0.514	243
94		
0	0.531	64
105		
1	0.516	64
108		
0	0.692	26

111		
1	0.554	56
118		
0	0.6	5
149		
1	0.526	76
239		
0	1	1
1	0.667	6

Table 25 - Social Win Probability vs. Distance Buckets

Given that sensor 111 is sensitive to incident distance, it is curious that the sensor does not show wins for incidents nearer to it (in bucket zero.) Conversely, sensor 94 shows a decrease in wins when incidents are farther away. One would expect that the impact of an incident on road speeds would decrease as the distance to the incident increases. Figure 19 - Social Win Probability vs. Stratified Distance to Incident for Sensors 94 and 111 indicate that sensor 94 follows the expected performance degradation, whereas sensor 111 does not.

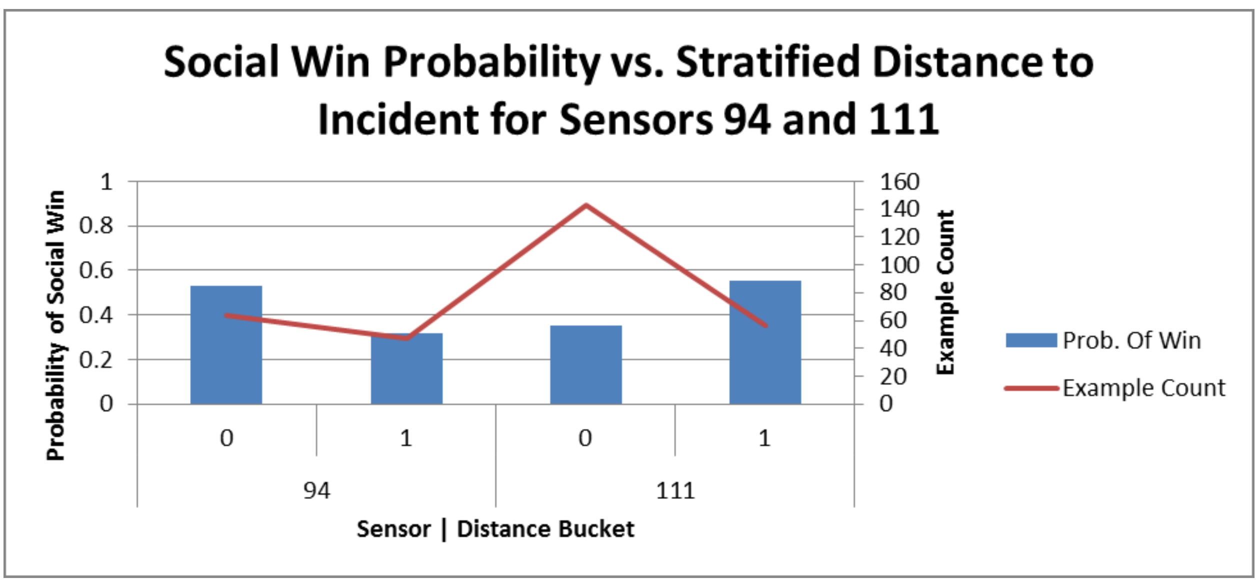


Figure 19 - Social Win Probability vs. Stratified Distance to Incident for Sensors 94 and 111

Figure 20 - Sensor 111 Probability of Social Win vs. Minimum Distance to Incident (also included in the data breakdown appendix) supports the claim that this sensor actually did perform markedly better when incidents were five to ten miles away than when incidents were nearer. When incidents were farther than ten miles away, the accuracy decreased as expected.

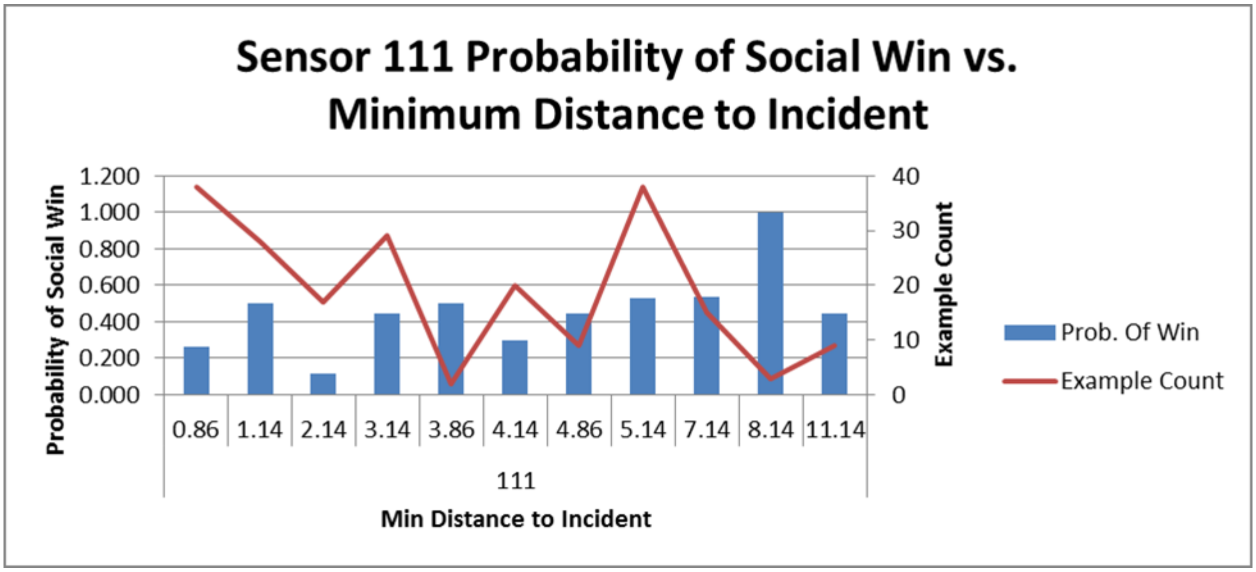


Figure 20 - Sensor 111 Probability of Social Win vs. Minimum Distance to Incident

This sensor is located in the HOV lane, so it may be interesting to examine whether other HOV lanes behave in a similar way. Excluding sensors with low example counts, it does appear that some HOV sensors show a slight sensitivity to incident distance, as evidenced by sensors 91, 94, 105, 111, and 149.

Sensor - Bucket	Prob. Of Win	Example Count
38		
2	0.667	12
49		
2	0.583	12
87		
1	0.504	262
91		
0	0.514	243
94		

0	0.531	64
105		
1	0.516	64
108		
0	0.692	26
111		
1	0.554	56
118		
0	0.6	5
149		
1	0.526	76
239		
0	1	1
1	0.667	6

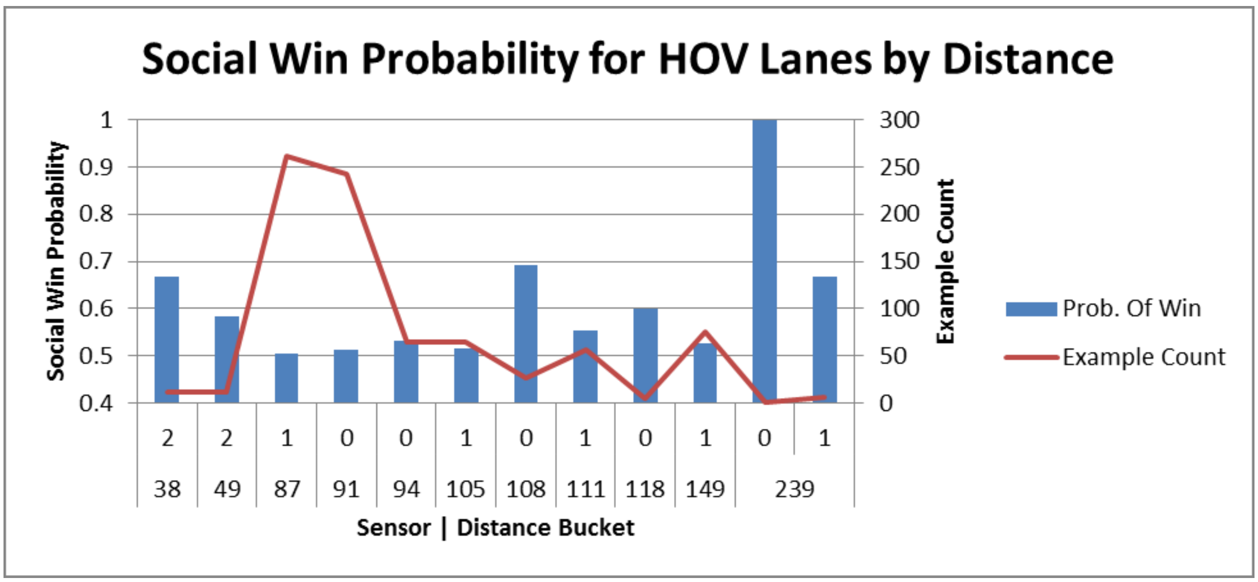


Figure 21 - Social Win Probability for HOV Lanes by Distance

Age of Tweets

There are many, many data points for sensors with win probability greater than .5, depending on the age of the last tweet. However, a significant number of these points had very few examples, which undermines the generalizability of any analysis that includes them. The following analysis excludes data points with fewer than thirty samples. Figure 22 - Social Win Probability vs. Age of Last Tweet where Example Count is Greater than 30. raises interesting questions about the data. The impact from a tweet may not be observed immediately on a segment, especially if it is in the opposite direction or a significant distance away. In a number of instances, win probability increases significantly as the age of the last tweet increases, which seems counter-intuitive. One possible explanation could be that the social ANN does a better job at modeling the recovery after the incident took place. Sensor 87 is a good example for drilling down into this behavior.

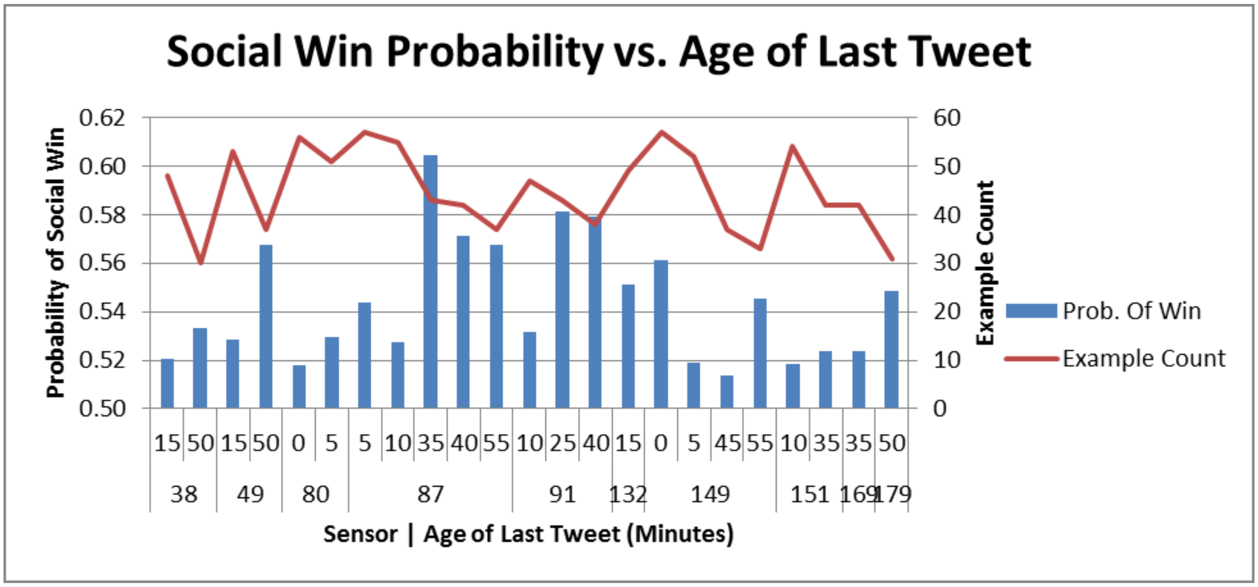


Figure 22 - Social Win Probability vs. Age of Last Tweet where Example Count is Greater than 30.

This sensor showed several instances winning with social data, when traffic patterns significantly deviate from the norm. Figure 23 - Sensor 87 Social Win for Incident Occurring 3/17/2012 After 21:00 shows the breakdown of predictions made by social and sensor-based ANNs and the actual speed that they attempt to predict. In this case, a tweet arrived at 22:35 (which corresponds to the dip of the Actual Speed near the middle of the graph) indicating that 2 lanes were blocked. The sensor-based ANN did not register a noticeable difference, whereas the social-enhanced ANN lowered its predictions until the tweet expired approximately 30 minutes later.

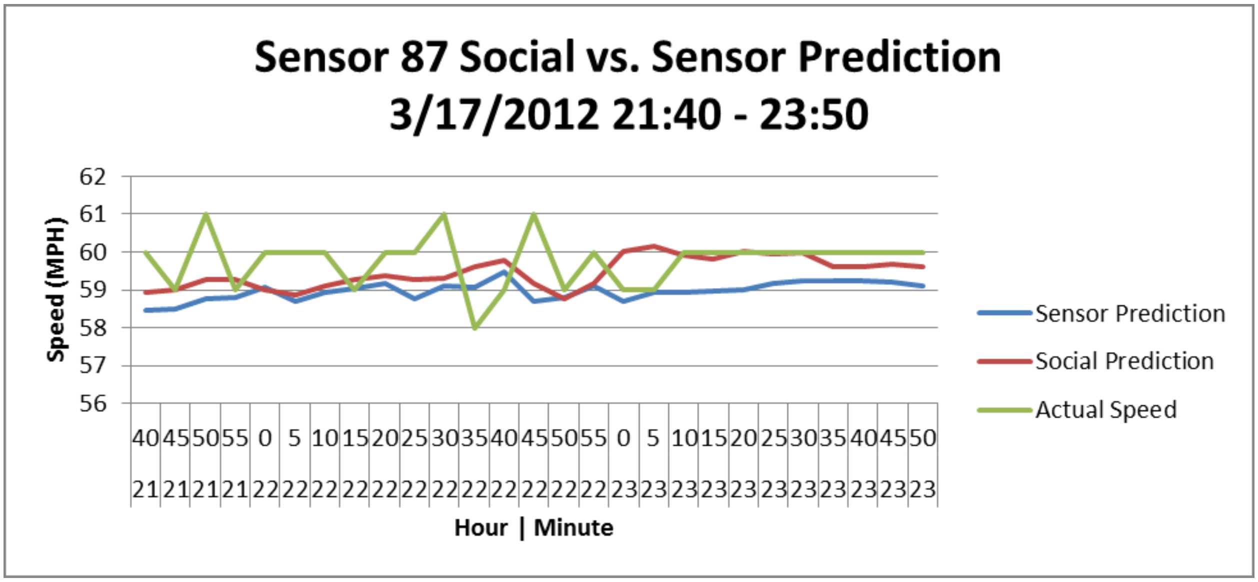


Figure 23 - Sensor 87 Social Win for Incident Occurring 3/17/2012 After 21:00

Theoretical Travel Times

To provide an overall sense for what a consumer could expect from models trained with social signals, theoretical travel times are examined. The travel times the ten best and ten worst instances were selected, based on the number of wins vs. losses observed for social models along the entire northbound or southbound corridor. For example, some worst cases scenarios have zero wins for the social model. The predictions were used to generate a theoretical travel time for the corridor, as described in the methods section. The social predictions, sensor predictions, and actual transit times are graphed against one another to illustrate the difference between the sensor and social models, as well as their differences from the observed speeds.

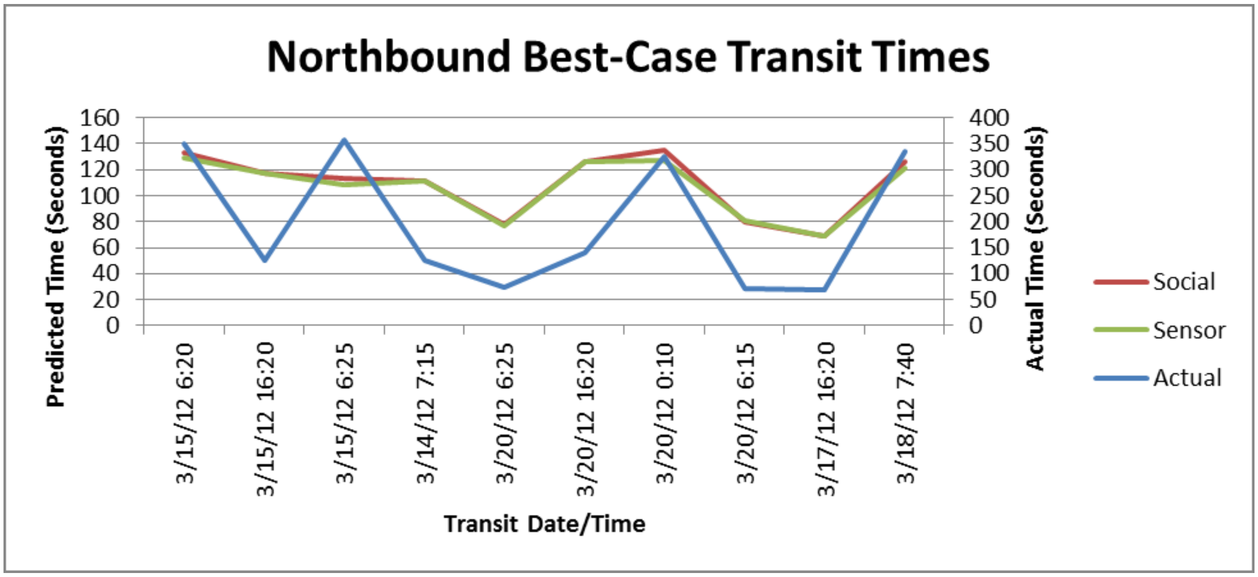


Figure 24 - Northbound Best-Case Travel Times

Date/Time		Actual	Social	Sensor
3/15/12	06:20	350	133	129
3/15/12	16:20	125	118	117
3/15/12	06:25	357	113	108
3/14/12	07:15	126	112	111
3/20/12	06:25	73	78	77
3/20/12	16:20	139	127	126
3/20/12	00:10	325	135	127
3/20/12	06:15	72	80	81
3/17/12	16:20	68	69	69
3/18/12	07:40	334	126	122

Table 26 - Northbound Best-Case Travel Times (in seconds)

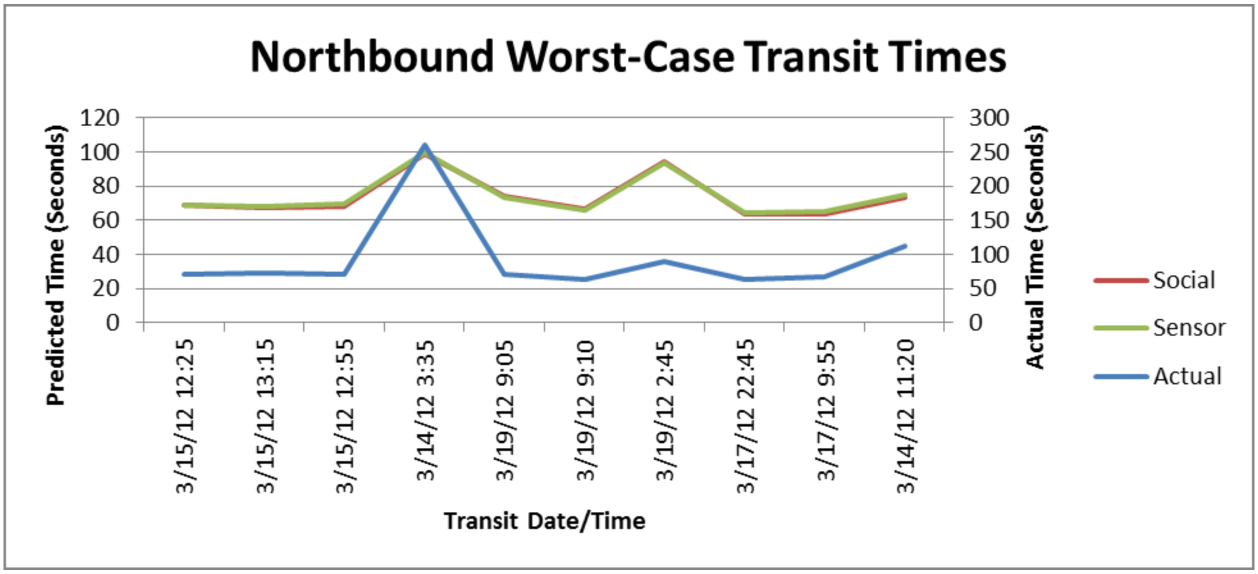


Figure 25 - Northbound Worst-Case Travel Times

Date/Time		Actual	Social	Sensor
3/15/12	12:25	71	69	69
3/15/12	13:15	73	67	68
3/15/12	12:55	70	68	69
3/14/12	3:35	259	99	99
3/19/12	9:05	71	74	73
3/19/12	9:10	64	67	66
3/19/12	2:45	89	94	94
3/17/12	22:45	63	64	64
3/17/12	9:55	66	64	65
3/14/12	11:20	111	73	75

Table 27 - Northbound Worst-Case Travel Times (in seconds)

It is immediately apparent, that even in the worst case, relying on social model predictions does not cause a significant negative impact on travel times, even in the worst case.

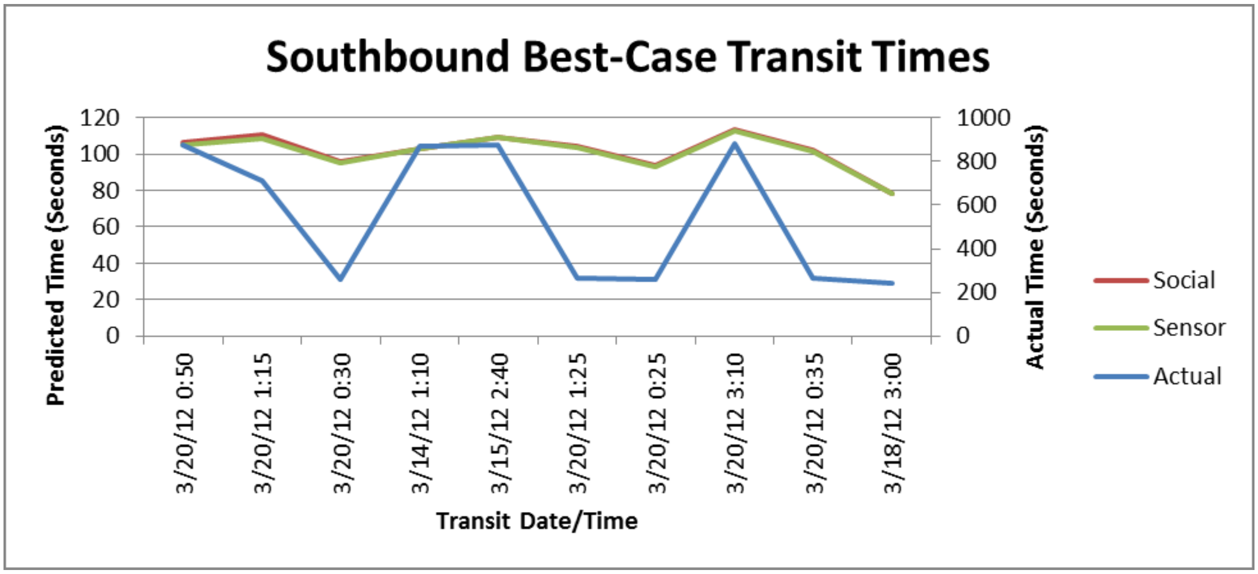


Figure 26 - Southbound Best-Case Transit Times

Date/Time	Actual	Social	Sensor
3/20/12 0:50	878	106	105
3/20/12 1:15	710	111	109
3/20/12 0:30	258	96	95
3/14/12 1:10	871	103	103
3/15/12 2:40	877	110	109
3/20/12 1:25	263	104	103
3/20/12 0:25	260	94	93
3/20/12 3:10	879	114	113
3/20/12 0:35	265	103	102
3/18/12 3:00	245	79	78

Table 28 - Southbound Best-Case Transit Times (in seconds)

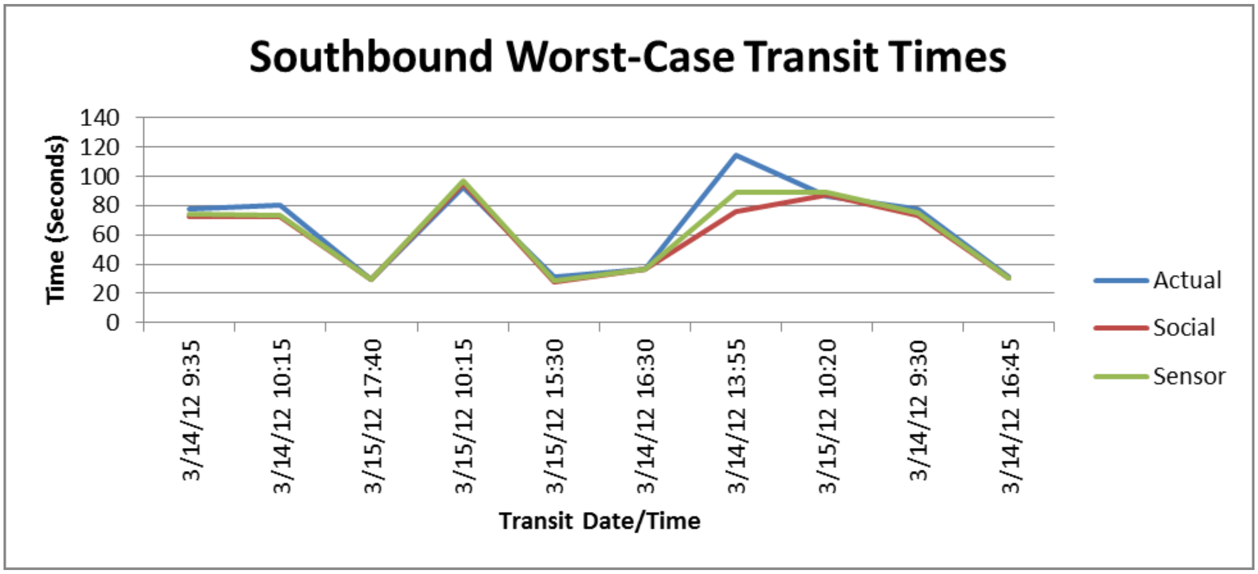


Figure 27 - Southbound Worst-Case Transit Times

Date/Time	Actual	Social	Sensor
3/14/12 9:35	77	73	74
3/14/12 10:15	80	73	73
3/15/12 17:40	30	30	30
3/15/12 10:15	93	95	97
3/15/12 15:30	31	28	29
3/14/12 16:30	37	37	37
3/14/12 13:55	114	76	89
3/15/12 10:20	86	87	89
3/14/12 9:30	78	73	75
3/14/12 16:45	31	30	31

Table 29 - Southbound Worst-Case Transit Times (in seconds)

The southbound case is somewhat unexpected, as it's quite accurate for a worst-case transit time. The worst case showcases times when the social models were less accurate than the sensor models without respect for the magnitude of the errors. It follows that there were a number of instances where both models were quite close but the social one lost out by a slim margin.

Chapter IV – Findings and Discussion

Overall, the results do not point to an overall significant increase in modeling accuracy when incorporating social signals into the datasets. It is important to point out that the various spikes in accuracy seen in HOV sensors for incidents at various distances and modest improvement in best-case transit time studies do not meet the bar of statistical significance to assert that they are caused by anything but randomness in the data. It is especially important to reiterate that while co-occurrence of specific conditions might prove to have a better than .5 probability of producing a win for sensors with social signals, this does not prove beyond a doubt that the impact is due to actual differences, as opposed to random noise.

Instead, this research provides a basis for understanding how to better model future experiments in this domain. The focus of that work should be to reduce the uncertainty produced by noisy data and by the rudimentary modeling approach used on the social signals in this experiment.

Data Quality

There are many references to the importance of data quality with respect to traffic modeling in the literature. (Turner, 2004) This experiment reinforced the need to carefully scrub out suspect data, as approximately one third of the examples in both the training and validation datasets needed to be discarded. The problem was compounded by the fact that each training example requires eight sequential samples: one for the current speed, the five previous speeds, the current occupancy, and the future speed to predict. This caused many examples to be discarded due to the absence of one or more of the samples and significantly reduced the dataset sizes. Some sensors, such as sensor 118 had such sparse data (163 examples in the validation set) that their results cannot be expected to generalize to a larger, more representative dataset.

One way to address this issue would be to build larger datasets for both training and validation. Analysis based on whether the day of week has an effect on modeling precision is worthless when the training dataset only contains four weeks of data. That's quite insufficient for a model to learn the difference between a Friday evening commute and Sunday evening at the same time. This is addressed to a large degree by the use of the prior speed data in generating predictions, however the days of the week exhibit different behavior in practice so ideally, the models should reflect that.

Further, collecting a large body of data with the expectation that a large amount of it will be discarded due to questions about its validity incurs additional threats to internal validity. The imbalance of data for sensors raises the question of whether that is a truly fair comparison. The periodic pattern of sensors providing corrupted data (or the failure of sensors to provide data at all) start to become baked into the models as well. When dealing with such noisy, imprecise signals as messages broadcast from Twitter, the introduction of additional noise signals generated by sensor dynamics can cause the models to learn the incorrect signals.

The inconsistencies in Twitter data broadcast by the @WSDOT_Traffic handle reduce its utility for automated consumption. Of the incidents described in the @WSDOT_Traffic broadcasts, very few had notifications of both the start and end of their impact. There were duplicated messages, where one was clearly from an automated tool and the other was an editorialized version of the same incident. The Twitter datasets used in this experiment contain many instances where the first broadcast from the @WSDOT_Traffic alias is an update for an existing event. This is counter-productive to modeling because the social signal was effectively telling the model that everything is clear prior to the message, when there was actually a known incident.

Matching multiple tweets about a single incident together to understand the progression of its impact (for example, there could be the initial report, possibly additional blocking characteristics when aid arrives on-scene, partial clearing, followed by complete clearing) is challenging to do in an automated fashion. Adding additional error incurred by fuzzy matching would further obfuscate any positive effects of social signals on traffic prediction. Therefore, it is worth reaching out to the WSDOT to request consistent incident identification (perhaps an incident ID included with each tweet) in addition to more predictable broadcasts, when it comes to the lifecycle of each incident.

Finally, as dataset sizes grow, the number of errant “noise” patterns also grows. Shawe-Taylor, et al reported that as more potential patterns are tested, more spurious ones will be picked up (Shawe-Taylor, De Bie, & Cristianini, 2006). This reinforces the need to have a large and representative verification dataset (comparable to the one that is mined for the initial patterns) to determine whether patterns are actual recurring phenomena vs. random noise that happens to coalesce in an interesting way, without any underlying cause.

Modeling and Re-Modeling Social Signals

Some modeling decisions in this experiment, such as the one to assume that all incidents have a sixty minute lifespan where they affect traffic, were needed to work around shortcomings in the underlying data. However, other modeling decisions should be revisited to further optimize the social signal presented to the models. There are many different avenues for modeling the interaction of multiple incidents to produce aggregate attributes representing the state of all incidents active for a given sample.

For example, using the mean of the direction attribute for aggregating data was not beneficial for analyzing the results. Stratification helped to glean some meaning, wherein incidents were either all in the same direction, all in the opposite direction, or somewhere in the middle. This didn't allow for direct comparison of similar data points and further confused the results. (e.g.: what does it mean when a sensor has an increased win probability when incidents' directionality has a mean of .14?) Sensors report speeds for each lane but the locations of tweets were only modeled by direction and location. The effects of an incident occurring very close to a sensor are likely to vary, based on whether it is in the same lane, or even whether it's in the lane to the right or left of the sensor.

Other attributes were also less effective than they could have been, due to the aggregation scheme when modeling. For example, the collision attribute is important but it is equally important to know whether the collision is in the current direction or the opposite one. The source datasets list these two attributes separately, so models can learn to correlate them but teasing out the probability of a win, given a set of prior conditions is difficult, at best.

There are signals that would be useful to model and could deliver a significant benefit when they are present, such as when a fatality accident triggers a multi-hour operation of taking measurements and the required documentation for such a tragic event. These events are very infrequent but have a disproportionately large impact on traffic. The "extraordinary distraction" attribute was aimed at addressing this, in addition to including it as a stratus of the "capacity impact" attribute. However, there were no occurrences of this in the validation dataset, so perhaps a restructuring of how the training and validation datasets are partitioned would allow sufficient representation in both sets.

Improvements in accuracy are also possible by restructuring the experiment to only train models on time samples that include social data. Restricting to this time rage allows the models to only learn the behavior observed when traffic incidents occur, rather than attempting to generalize across the whole spectrum of behavior. This approach assumes that we can train general purpose models that can generate accurate predictions for the nominal case and then switch to the special social models when an incident tweet is received from @WSDOT_Traffic. The signal-to-noise ratio would be significantly stronger for the social signals, so a more exact fitting of the data could be expected. However, this approach requires that we know when an incident is no longer impacting traffic. The problem could be remedied by reaching out to the WSDOT to request that they be more disciplined about broadcasting an end to incident impact or by using a heuristic, such as when the speed is back to within ten percent of the limit, the incident is deemed to be over.

Appendix 1 – WSDOT Data Extraction Method

In the tool, the Raw Data option provides the data in a format that is closest to what is needed to train an ANN (the tool is designed to generate reports, so it provides single and multi-day aggregation capabilities as well. These are not used in this experiment) (See Figure 28 - Selecting raw data for export using CDR)

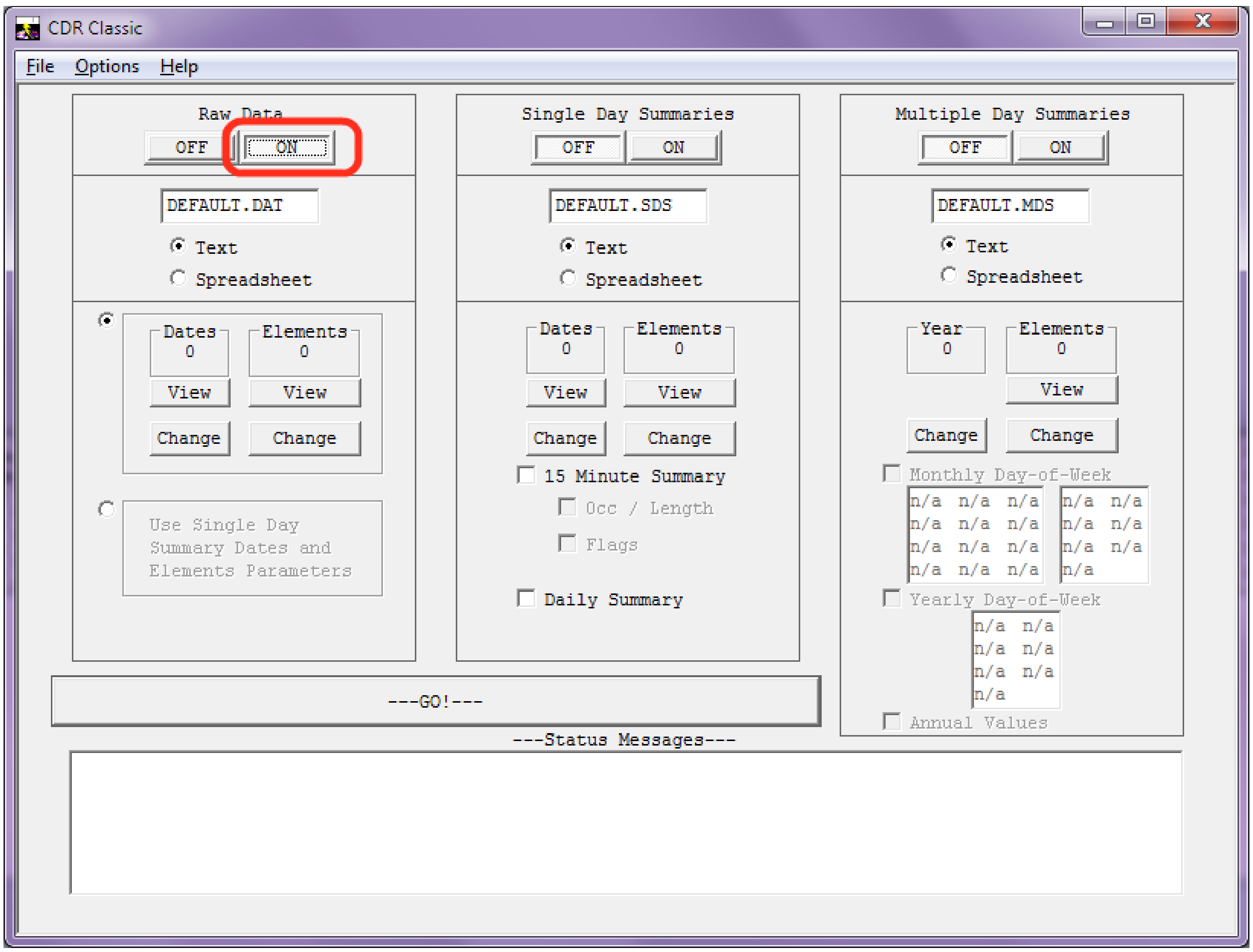


Figure 28 - Selecting raw data for export using CDR

All available dates need to be exported. Dates are selected pushing the “Change” button in the Dates control (See Figure 29 - Selecting dates in CDR)

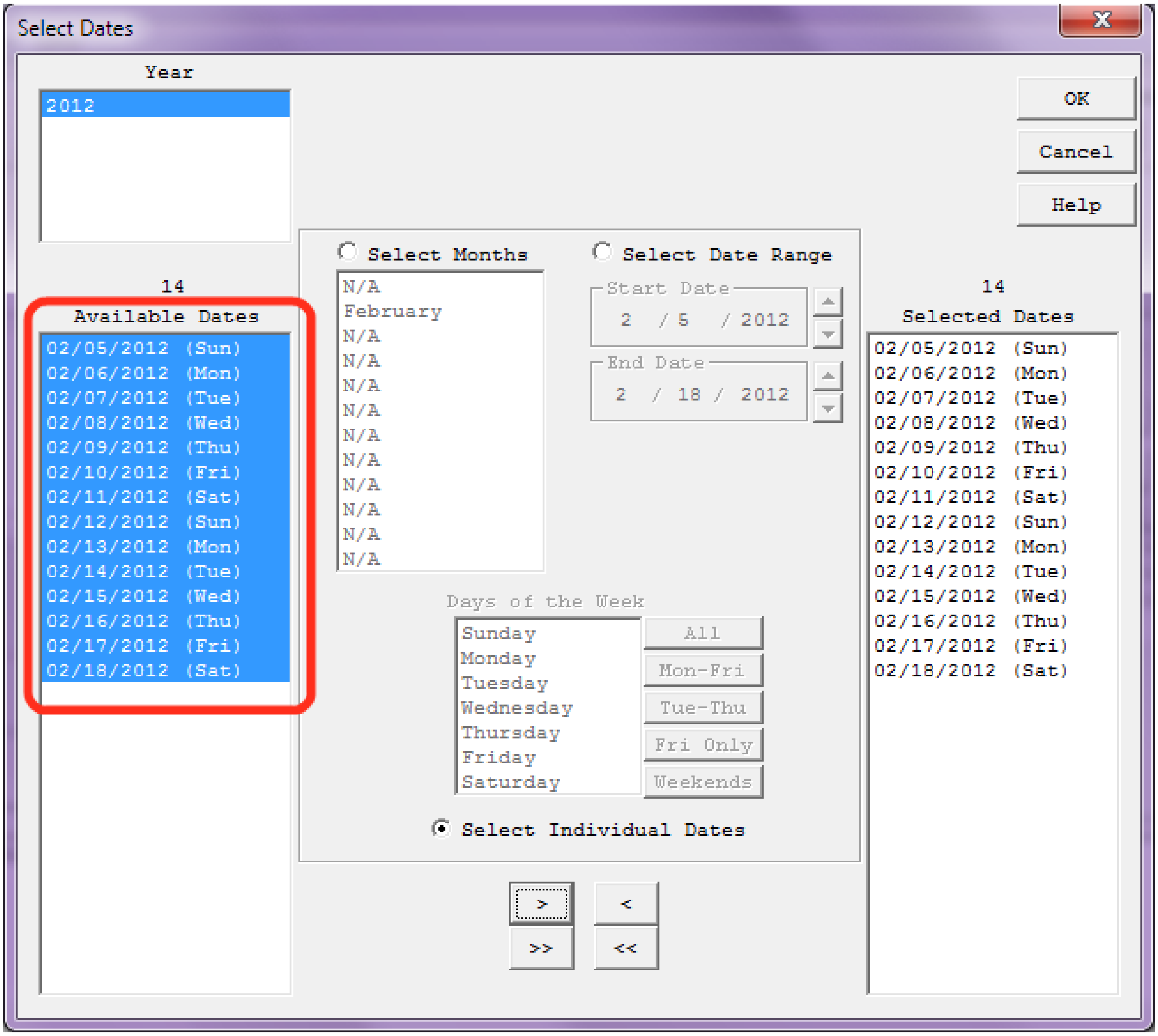


Figure 29 - Selecting dates in CDR

Finally, the "Elements" (speed sensors) are selected by pushing the "Change" button in the Elements control. The resulting dialog allows users to choose the roadway (1), the cabinet along the roadway (2) (which is like a recording station where the nearby sensors' data is collected and routed to the WSDOT), and the individual sensors themselves (3). The sensors selected for export are shown in the pane at the right (4). (See Figure 30 - Selecting sensors to export in CDR)

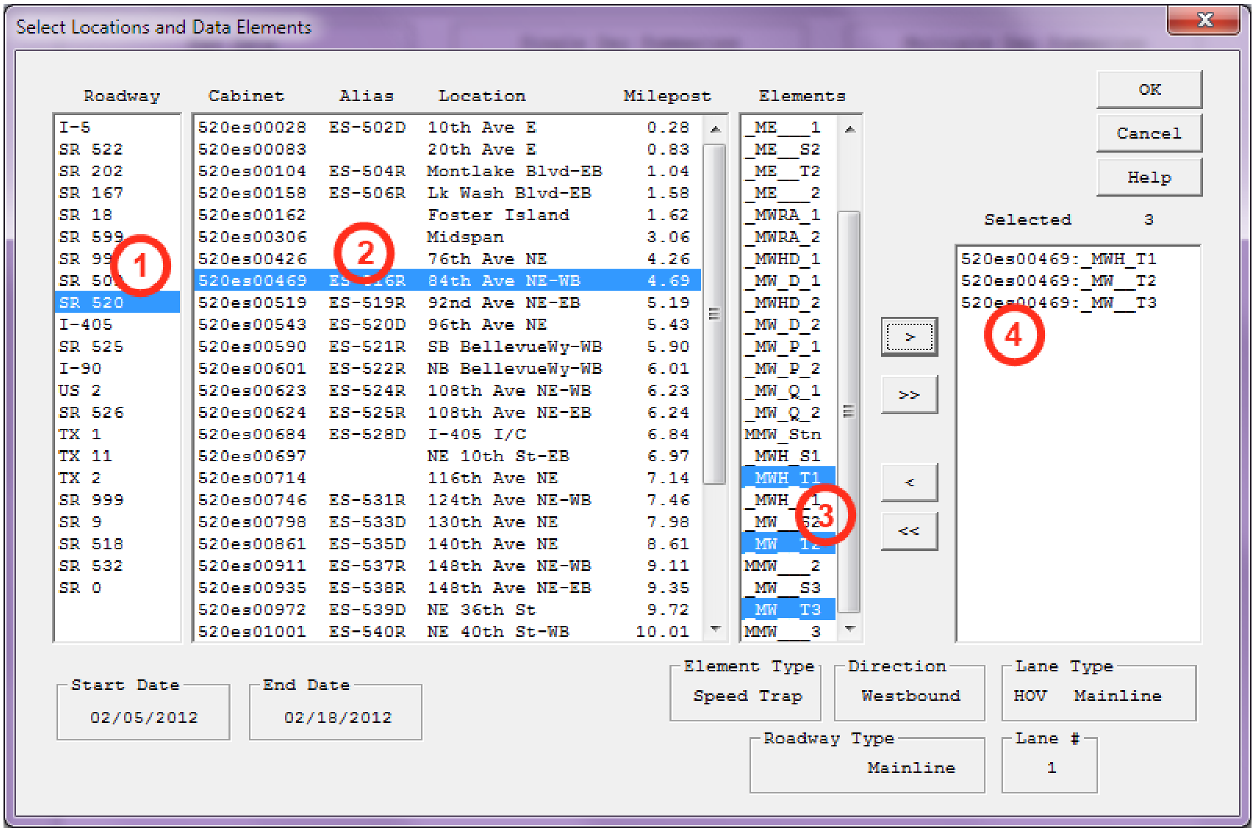


Figure 30 - Selecting sensors to export in CDR

The export process is completed by dismissing the dialogs and pressing the large “GO” button in CDR.

The reports generated by CDR are in a human readable format so they contain rich header information preceding each day’s set of sensor data.

```
     CDR Output Report
     ***********************************
Filename: DEFAULT.DAT

Creation Date: 02/19/12 (Sun)
Creation Time: 10:46:06
    File Type: TEXT
***********************************
     520es00469:_MWH_S1  SR 520  84th Ave NE-WB      4.69
     02/05/2012 (Sun)
     ---Raw Loop Data Listing---
     Time    Vol    Occ   Flg  nPds
     ------------------------------
      0:00      0    0.0%   1    15
      0:05      0    0.0%   1    15
```

Formatting for Import

An AWK script was used to reformat the data into a tab-delimited file where each line contained the sensor name, the timestamp, and the data. This makes it easier to import into a database, as each line in the report file has a uniform format and contains the timestamp, sensor name, and sensor data. The WSDOT uses the Flg (flag) parameter to indicate when data is suspect or missing. The road sensors regularly drop samples, so zero values may be reported when there is actually no data. To eliminate the possibility of error introduced by suspect data, samples that are listed as anything other than correct are not included in this experiment.

```
     Database Import Format
     Sensor    Day  Time Vol  Occ  Flg  nPds
520es00469:_MWH_S1  02/05/2012    0:00 0    0.0% 1    15
520es00469:_MWH_S1  02/05/2012    0:05 0    0.0% 1    15
```

The data is then imported into a Microsoft Access database, where it is normalized for efficient storage and retrieval.

Appendix 2 – Testing Training Parameter Effects on Error

Learning Rate	Momentum	Epochs	RMS Error Training*	RMS Error Eval*
0.7	0.7	1000	0.06934	0.06529
0.7	0.7	2500	0.06868	0.064977
0.7	0.7	5000	0.06841	0.06493
0.7	0.7	10000	0.06814	0.0649
0.7	0.7	20000	0.06776	0.06477
0.7	0.7	50000	0.0661	0.06538
0.5	0.7	1000	0.06931	0.06528
0.5	0.7	2500	0.06889	0.065
0.5	0.7	5000	0.0686	0.06495
0.5	0.7	10000	0.06828	0.06492
0.5	0.7	20000	0.06766	0.06482
0.5	0.7	50000	0.06651	0.06518
0.3	0.7	1000	0.06911	0.0653
0.3	0.7	2500	0.06879	0.06513
0.3	0.7	5000	0.06861	0.065
0.3	0.7	10000	0.06838	0.06494
0.3	0.7	20000	0.06796	0.06488
0.3	0.7	50000	0.06711	0.0651
0.7	0.5	1000	0.06916	0.0659
0.7	0.5	2500	0.06886	0.06559
0.7	0.5	5000	0.06868	0.06542
0.7	0.5	10000	0.06848	0.06529
0.7	0.5	20000	0.06818	0.06511
0.7	0.5	50000	0.06756	0.06481
0.5	0.5	1000	0.0693	0.0655
0.5	0.5	2500	0.06973	0.06504
0.5	0.5	5000	0.06857	0.06492
0.5	0.5	10000	0.06841	0.0649
0.5	0.5	20000	0.06808	0.06489
0.5	0.5	50000	0.06697	0.06497
0.3	0.5	1000	0.0695	0.066
0.3	0.5	2500	0.06885	0.06526
0.3	0.5	5000	0.06866	0.065
0.3	0.5	10000	0.0685	0.065
0.3	0.5	20000	0.0681	0.0649
0.3	0.5	50000	0.0673	0.0648
0.7	0.3	1000	0.0693	0.0647
0.7	0.3	2500	0.06885	0.0648

0.7	0.3	5000	0.06869	0.0648
0.7	0.3	10000	0.06849	0.0648
0.7	0.3	20000	0.0682	0.0647
0.7	0.3	50000	0.0677	0.06462
0.5	0.3	1000	0.0694	0.065
0.5	0.3	2500	0.069	0.065
0.5	0.3	5000	0.0687	0.065
0.5	0.3	10000	0.0685	0.0651
0.5	0.3	20000	0.0683	0.0651
0.5	0.3	50000	0.0675	0.0652
0.3	0.3	1000	0.0689	0.0652
0.3	0.3	2500	0.0687	0.06524
0.3	0.3	5000	0.0686	0.065
0.3	0.3	10000	0.0684	0.065
0.3	0.3	20000	0.0681	0.065
0.3	0.3	50000	0.0677	0.065
0.7	0.7	1000	0.06934	0.06529
0.7	0.7	2500	0.06868	0.064977
0.7	0.7	5000	0.06841	0.06493
0.7	0.7	10000	0.06814	0.0649

Table 30 - ANN Training without Social Signals

Learning Rate	Momentum	Epochs	RMS Error Training*	RMS Error Eval*
0.7	0.7	1000	0.06539	0.06316
0.7	0.7	2500	0.06474	0.0625
0.7	0.7	5000	0.06446	0.0621
0.7	0.7	10000	0.0641	0.0619
0.7	0.7	20000	0.0637	0.0617
0.7	0.7	50000	0.0627	0.062
0.5	0.7	1000	0.06531	0.06306
0.5	0.7	2500	0.0648	0.06296
0.5	0.7	5000	0.06461	0.06288
0.5	0.7	10000	0.06432	0.0626
0.5	0.7	20000	0.0638	0.0623
0.5	0.7	50000	0.0624	0.0625
0.3	0.7	1000	0.06542	0.0632
0.3	0.7	2500	0.06477	0.06293
0.3	0.7	5000	0.0645	0.0626
0.3	0.7	10000	0.06426	0.0624
0.3	0.7	20000	0.0638	0.0623
0.3	0.7	50000	0.06271	0.06258

0.7	0.5	1000	0.0651	0.0633
0.7	0.5	2500	0.0648	0.0627
0.7	0.5	5000	0.0646	0.0624
0.7	0.5	10000	0.0643	0.0622
0.7	0.5	20000	0.064	0.0621
0.7	0.5	50000	0.0632	0.062
0.5	0.5	1000	0.067	0.0653
0.5	0.5	2500	0.0653	0.0639
0.5	0.5	5000	0.0647	0.0633
0.5	0.5	10000	0.0644	0.063
0.5	0.5	20000	0.064	0.0627
0.5	0.5	50000	0.0632	0.0625
0.3	0.5	1000	0.0654	0.0636
0.3	0.5	2500	0.0649	0.0629
0.3	0.5	5000	0.0645	0.0626
0.3	0.5	10000	0.0643	0.0625
0.3	0.5	20000	0.064	0.0624
0.3	0.5	50000	0.0634	0.06212
0.7	0.3	1000	0.0657	0.0628
0.7	0.3	2500	0.0651	0.0626
0.7	0.3	5000	0.0648	0.0625
0.7	0.3	10000	0.0646	0.0625
0.7	0.3	20000	0.0644	0.0625
0.7	0.3	50000	0.0637	0.0626
0.5	0.3	1000	0.0652	0.0631
0.5	0.3	2500	0.0648	0.0627
0.5	0.3	5000	0.0646	0.0625
0.5	0.3	10000	0.0644	0.0623
0.5	0.3	20000	0.06419	0.0621
0.5	0.3	50000	0.0634	0.0617
0.3	0.3	1000	0.0659	0.0644
0.3	0.3	2500	0.0652	0.0632
0.3	0.3	5000	0.0648	0.0628
0.3	0.3	10000	0.0646	0.0625
0.3	0.3	20000	0.0643	0.0623
0.3	0.3	50000	0.0638	0.0621
0.7	0.7	1000	0.06539	0.06316
0.7	0.7	2500	0.06474	0.0625
0.7	0.7	5000	0.06446	0.0621
0.7	0.7	10000	0.0641	0.0619
0.7	0.7	20000	0.0637	0.0617

0.7	0.7	50000	0.0627	0.062
0.5	0.7	1000	0.06531	0.06306

Table 31 - ANN Training with Social Signals

* RMS error values depicted in the tables above are scaled between 0 and 1, per the convention used for quantifying ANN error in the literature. In addition, the learning rate is set to decay by 1% every seven epochs. This feature optimizing the training process; as training progresses outliers can prevent weights from converging at optimal values for the training set. Decreasing the learning rate over time reduces this impact as the ANN fits a curve that best maps to the training set which will have the greatest error when measured against outlier data points. If the learning rate is too high, weights are adjusted excessively in each epoch, causing an oscillation that may never converge. This training process is configured to cut the learning rate in half if the RMS error increases by more than .1%, to prevent oscillations.

Appendix 3 – Sensor Data Speed Statistics

Sensor 38

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	66.32	66.34	66.34	66.33	66.32
StDev	3.46	3.40	3.35	3.44	3.38
Mode	67	67	67	67	67
Example Count	5053	5054	5059	5072	5057
Min	15	15	15	15	15
Max	72	72	72	72	72
Q1	66	66	66	66	66
Median	67	67	67	67	67
Q3	68	68	68	68	68

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	66.32	66.34	66.34	66.33	66.32
StDev	3.26	3.40	3.34	3.43	3.38
Mode	67	67	67	67	67
Example Count	6026	5072	5077	5090	5075
Min	15	15	15	15	15
Max	72	72	72	72	72
Q1	66	66	66	66	66
Median	67	67	67	67	67
Q3	68	68	68	68	68

Validation	Without Social Data	With Social Data
Mean	65.98	65.53
StDev	4.33	5.51
Mode	67	67
Example Count	1298	1573
Min	22	22
Max	73	73
Q1	65	65
Median	67	67
Q3	68	68

Sensor 39

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5

Mean	62.87	62.87	62.87	62.88	62.86
StDev	3.02	3.02	3.02	3.02	3.02
Mode	63	63	63	63	63
Example Count	1790	1771	1778	1764	1778
Min	14	14	14	14	14
Max	73	73	73	73	73
Q1	61	61	61	61	61
Median	63	63	63	63	63
Q3	65	64	65	65	65

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.88	62.87	62.87	62.88	62.86
StDev	3.01	3.02	3.02	3.02	3.02
Mode	63	63	63	63	63
Example Count	1797	1771	1778	1764	1778
Min	14	14	14	14	14
Max	73	73	73	73	73
Q1	61	61	61	61	61
Median	63	63	63	63	63
Q3	65	64	65	65	65

Validation	Without Social Data	With Social Data
Mean	61.71	61.27
StDev	5.84	6.89
Mode	63	63
Example Count	452	460
Min	21	21
Max	70	70
Q1	61	61
Median	63	63
Q3	64	64

Sensor 49

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	64.61	64.63	64.62	64.61	64.61
StDev	4.08	4.04	4.03	4.06	4.08
Mode	66	66	66	66	66
Example Count	7570	7556	7576	7565	7557
Min	15	15	15	15	15

Max	74	74	74	74	74
Q1	64	64	64	64	64
Median	65	65	65	65	65
Q3	66	66	66	66	66

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	64.49	64.63	64.62	64.61	64.60
StDev	4.02	4.04	4.02	4.06	4.08
Mode	65	66	66	66	66
Example Count	8482	7574	7594	7583	7575
Min	15	15	15	15	15
Max	74	74	74	74	74
Q1	64	64	64	64	64
Median	65	65	65	65	65
Q3	66	66	66	66	66

Validation	Without Social Data	With Social Data
Mean	63.52	63.05
StDev	6.33	7.18
Mode	66	66
Example Count	1934	2197
Min	16	16
Max	71	71
Q1	63	63
Median	65	65
Q3	66	66

Sensor 72
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.77	62.76	62.76	62.75	62.75
StDev	3.13	3.15	3.19	3.19	3.16
Mode	64	64	64	64	64
Example Count	7900	7887	7909	7902	7889
Min	13	13	13	13	13
Max	71	71	71	71	71
Q1	62	62	62	62	62
Median	63	63	63	63	63
Q3	64	64	64	64	64

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.61	62.75	62.75	62.74	62.74
StDev	3.35	3.18	3.22	3.22	3.19
Mode	63	64	64	64	64
Example Count	8900	7907	7929	7922	7909
Min	13	13	13	13	13
Max	71	71	71	71	71
Q1	62	62	62	62	62
Median	63	63	63	63	63
Q3	64	64	64	64	64

Validation	Without Social Data	With Social Data
Mean	62.47	62.38
StDev	2.16	2.12
Mode	63	63
Example Count	2004	2279
Min	40	40
Max	73	73
Q1	61	61
Median	63	63
Q3	64	64

Sensor 77

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	59.38	59.44	59.41	59.41	59.40
StDev	8.60	8.51	8.55	8.52	8.57
Mode	63	63	63	63	63
Example Count	6842	6848	6872	6861	6847
Min	12	12	12	12	12
Max	69	69	69	69	69
Q1	60	60	60	60	60
Median	62	62	62	62	62
Q3	63	63	63	63	63

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	58.70	59.41	59.37	59.38	59.37
StDev	9.55	8.56	8.60	8.57	8.62
Mode	63	63	63	63	63

Example Count	7842	6868	6892	6881	6867
Min	12	12	12	12	12
Max	69	69	69	69	69
Q1	60	60	60	60	60
Median	62	62	62	62	62
Q3	63	63	63	63	63

Validation	Without Social Data	With Social Data
Mean	58.49	57.73
StDev	8.53	9.22
Mode	62,63	62
Example Count	1737	2012
Min	15	15
Max	67	67
Q1	58	58
Median	61	61
Q3	63	63

Sensor 80

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	63.15	63.24	63.17	63.20	63.15
StDev	11.25	11.12	11.20	11.17	11.25
Mode	68	68	68	68	68
Example Count	5108	5104	5111	5121	5100
Min	13	13	13	13	13
Max	72	72	72	72	72
Q1	65	65	65	65	65
Median	67	67	67	67	67
Q3	68	68	68	68	68

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.45	63.20	63.12	63.15	63.11
StDev	12.13	11.18	11.27	11.23	11.31
Mode	68	68	68	68	68
Example Count	6076	5122	5129	5139	5118
Min	13	13	13	13	13
Max	72	72	72	72	72
Q1	64	65	65	65	65
Median	67	67	67	67	67

Q3	68	68	68	68	68

Validation	Without Social Data	With Social Data
Mean	62.85	61.90
StDev	11.15	11.89
Mode	68	67
Example Count	1335	1610
Min	13	13
Max	72	72
Q1	64	63
Median	67	66
Q3	68	68

Sensor 87

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	59.48	59.48	59.47	59.47	59.47
StDev	3.15	3.19	3.18	3.16	3.16
Mode	60	60	60	60	60
Example Count	7109	7094	7106	7105	7094
Min	15	15	15	15	15
Max	66	66	66	66	66
Q1	59	59	59	59	59
Median	60	60	60	60	60
Q3	61	61	61	61	61

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	59.41	59.47	59.47	59.47	59.46
StDev	3.30	3.22	3.20	3.19	3.19
Mode	60	60	60	60	60
Example Count	8044	7112	7124	7123	7112
Min	15	15	15	15	15
Max	66	66	66	66	66
Q1	59	59	59	59	59
Median	60	60	60	60	60
Q3	61	61	61	61	61

Validation	Without Social Data	With Social Data
Mean	59.01	58.98

StDev	3.37	3.20
Mode	60	60
Example Count	1997	2272
Min	0	0
Max	64	64
Q1	58	58
Median	59	59
Q3	60	60

Sensor 91

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.73	61.77	61.72	61.72	61.74
StDev	10.02	9.92	10.03	10.02	10.01
Mode	65	65	65	65	65
Example Count	6407	6423	6444	6436	6419
Min	11	11	11	11	11
Max	70	70	70	70	70
Q1	63	63	63	63	63
Median	65	65	65	65	65
Q3	66	66	66	66	66

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.28	61.75	61.69	61.70	61.71
StDev	10.58	9.97	10.09	10.06	10.06
Mode	65	65	65	65	65
Example Count	7216	6439	6460	6452	6435
Min	11	11	11	11	11
Max	70	70	70	70	70
Q1	63	63	63	63	63
Median	64	65	65	65	65
Q3	65	66	66	66	66

Validation	Without Social Data	With Social Data
Mean	60.16	59.13
StDev	12.02	13.02
Mode	65	65
Example Count	1622	1862
Min	12	12
Max	69	69

Q1	62	62
Median	64	64
Q3	65	65

Sensor 94
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	58.12	58.05	58.10	58.05	58.05
StDev	9.61	9.65	9.66	9.69	9.65
Mode	61	61	61	61	61
Example Count	2712	2696	2701	2694	2708
Min	0	0	0	0	0
Max	82	82	82	82	82
Q1	58	57	57	58	57
Median	60	60	60	60	60
Q3	62	62	62	62	62

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	57.79	58.03	58.08	58.03	58.03
StDev	9.93	9.66	9.68	9.70	9.67
Mode	61	61	61	61	61
Example Count	2808	2702	2706	2700	2714
Min	0	0	0	0	0
Max	82	82	82	82	82
Q1	57	57	57	57	57
Median	60	60	60	60	60
Q3	62	62	62	62	62

Validation	Without Social Data	With Social Data
Mean	56.72	55.44
StDev	10.18	11.52
Mode	58	58
Example Count	684	721
Min	0	0
Max	82	82
Q1	57	56
Median	59	59
Q3	61	61

Sensor 98

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	66.10	66.27	66.01	65.97	65.88
StDev	26.90	26.85	27.05	26.99	27.09
Mode	87	87	87	87	87
Example Count	2278	2250	2256	2253	2252
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	42	43	42	42	42
Median	78	78	78	78	78
Q3	86	86	86	86	86

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	63.19	66.14	65.87	65.84	65.75
StDev	27.04	26.86	27.06	27.01	27.10
Mode	87	87	87	87	87
Example Count	2538	2260	2267	2263	2262
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	39	43	42	42	42
Median	76	78	78	78	78
Q3	85	86	86	86	86

Validation	Without Social Data	With Social Data
Mean	66.42	63.01
StDev	24.97	24.89
Mode	87	87
Example Count	651	760
Min	0	0
Max	100	100
Q1	45	40
Median	77	73
Q3	85	84

Sensor 105

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	51.56	51.53	51.49	51.42	51.50
StDev	16.13	16.15	16.18	16.25	16.16
Mode	62	62	62	62	62

Example Count	2853	2834	2842	2819	2827
Min	11	11	11	11	11
Max	71	71	71	71	71
Q1	41	41	39	39	40
Median	60	60	60	60	60
Q3	62	62	62	62	62

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	48.66	51.43	51.40	51.33	51.41
StDev	17.47	16.21	16.24	16.32	16.22
Mode	62	62	62	62	62
Example Count	3231	2845	2853	2830	2838
Min	11	11	11	11	11
Max	71	71	71	71	71
Q1	28	39	38	38	39
Median	59	60	60	60	60
Q3	62	62	62	62	62

Validation	Without Social Data	With Social Data
Mean	48.89	45.53
StDev	16.66	17.49
Mode	60	60
Example Count	775	926
Min	14	14
Max	68	68
Q1	29	25
Median	59	56
Q3	61	61

Sensor 108
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.66	62.76	62.72	62.71	62.74
StDev	6.41	6.20	6.22	6.25	6.30
Mode	63	63	63	63	63
Example Count	1729	1706	1710	1703	1703
Min	0	0	0	0	0
Max	76	76	74	76	76
Q1	62	62	62	62	62
Median	63	63	63	63	63

Q3	65	65	65	65	65

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.57	62.74	62.69	62.69	62.71
StDev	6.59	6.28	6.31	6.33	6.39
Mode	63	63	63	63	63
Example Count	1738	1707	1711	1704	1704
Min	0	0	0	0	0
Max	76	76	74	76	76
Q1	61	62	62	62	62
Median	63	63	63	63	63
Q3	65	65	65	65	65

Validation	Without Social Data	With Social Data
Mean	63.00	62.91
StDev	3.38	3.44
Mode	63	63
Example Count	399	405
Min	41	41
Max	76	76
Q1	61	61
Median	63	63
Q3	65	65

Sensor 109

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	63.79	63.83	63.80	63.76	63.79
StDev	6.01	5.89	5.99	5.93	5.99
Mode	65	65	65	65	65
Example Count	1262	1248	1248	1243	1236
Min	16	16	16	16	16
Max	74	74	73	74	74
Q1	63	63	63	63	63
Median	65	65	65	65	65
Q3	66	66	66	66	66

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	63.58	63.80	63.76	63.72	63.76

StDev	6.64	6.03	6.13	6.08	6.14
Mode	65	65	65	65	65
Example Count	1275	1249	1249	1244	1237
Min	16	16	16	16	16
Max	74	74	73	74	74
Q1	63	63	63	63	63
Median	65	65	65	65	65
Q3	66	66	66	66	66

Validation	Without Social Data	With Social Data
Mean	63.68	63.61
StDev	4.09	4.15
Mode	65	65
Example Count	273	275
Min	38	38
Max	72	72
Q1	62	62
Median	64	64
Q3	66	66

Sensor 111

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.51	62.61	62.60	62.58	62.57
StDev	16.86	16.57	16.68	16.67	16.73
Mode	66	66	66	66	66
Example Count	3830	3817	3825	3815	3815
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	63	63	63	63	63
Median	66	66	66	66	66
Q3	69	69	69	69	69

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	62.61	62.60	62.59	62.57	62.56
StDev	16.37	16.57	16.68	16.68	16.73
Mode	66	66	66	66	66
Example Count	4133	3825	3833	3823	3823
Min	0	0	0	0	0
Max	100	100	100	100	100

Q1	63	63	63	63	63
Median	66	66	66	66	66
Q3	69	69	69	69	69

Validation	Without Social Data	With Social Data
Mean	62.06	62.16
StDev	16.25	15.69
Mode	66	66
Example Count	974	1049
Min	0	0
Max	87	87
Q1	62	62
Median	65	65
Q3	68	68

Sensor 118

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	60.47	60.51	60.48	60.46	60.49
StDev	2.14	2.11	2.14	2.14	2.13
Mode	60	60	60	60	60
Example Count	733	717	711	722	707
Min	51	51	51	51	51
Max	66	66	66	66	66
Q1	59	59	59	59	59
Median	61	61	60	60	61
Q3	62	62	62	62	62

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	60.47	60.51	60.48	60.46	60.49
StDev	2.14	2.11	2.14	2.14	2.13
Mode	60	60	60	60	60
Example Count	733	717	711	722	707
Min	51	51	51	51	51
Max	66	66	66	66	66
Q1	59	59	59	59	59
Median	61	61	60	60	61
Q3	62	62	62	62	62

Validation	Without Social	With Social

	Data	Data
Mean	60.71	60.71
StDev	2.17	2.17
Mode	61	61
Example Count	163	163
Min	52	52
Max	68	68
Q1	59	59
Median	61	61
Q3	62	62

Sensor 119

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	57.57	57.86	57.82	57.78	57.66
StDev	21.32	20.96	21.08	21.05	21.19
Mode	0	0	0	0	0
Example Count	2588	2569	2583	2569	2571
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	61	61	61	61	61
Median	64	64	64	64	64
Q3	68	68	68	68	68

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	57.63	57.86	57.82	57.78	57.66
StDev	21.18	20.96	21.08	21.05	21.19
Mode	0	0	0	0	0
Example Count	2632	2569	2583	2569	2571
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	61	61	61	61	61
Median	64	64	64	64	64
Q3	68	68	68	68	68

Validation	Without Social Data	With Social Data
Mean	56.62	56.61
StDev	22.46	22.34
Mode	0	0
Example Count	602	609

Min	0	0
Max	93	93
Q1	60	60
Median	64	64
Q3	67	67

Sensor 132

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	50.80	50.81	50.81	50.84	50.83
StDev	11.43	11.40	11.43	11.40	11.41
Mode	61	61	61	61	61
Example Count	5711	5711	5728	5735	5723
Min	15	15	15	15	15
Max	67	67	67	67	67
Q1	40	40	40	40	40
Median	56	56	56	56	56
Q3	60	60	60	60	60

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	49.36	50.78	50.78	50.81	50.80
StDev	11.81	11.40	11.44	11.41	11.42
Mode	61	61	61	61	61
Example Count	6648	5730	5747	5754	5742
Min	15	15	15	15	15
Max	67	67	67	67	67
Q1	37	40	40	40	40
Median	54	56	56	56	56
Q3	60	60	60	60	60

Validation	Without Social Data	With Social Data
Mean	49.99	48.34
StDev	12.03	12.36
Mode	61	61
Example Count	1488	1763
Min	15	15
Max	65	65
Q1	38	36
Median	56	53
Q3	60	60

Sensor 149

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	52.39	52.38	52.38	52.36	52.41
StDev	10.74	10.77	10.78	10.78	10.72
Mode	59	59	59	59	59
Example Count	6228	6230	6244	6246	6235
Min	11	11	11	11	11
Max	66	66	66	66	66
Q1	50	50	50	50	50
Median	56	56	56	56	56
Q3	59	59	59	59	59

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	50.84	52.34	52.34	52.32	52.37
StDev	11.62	10.80	10.81	10.81	10.75
Mode	58	59	59	59	59
Example Count	7204	6250	6264	6266	6255
Min	11	11	11	11	11
Max	66	66	66	66	66
Q1	48	50	50	50	50
Median	55	56	56	56	56
Q3	59	59	59	59	59

Validation	Without Social Data	With Social Data
Mean	50.66	49.37
StDev	11.66	12.02
Mode	58	58
Example Count	1592	1859
Min	14	14
Max	65	65
Q1	48	45
Median	56	54
Q3	59	58

Sensor 151

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	63.15	63.14	63.14	63.16	63.14

StDev	16.80	16.85	16.85	16.76	16.82
Mode	71	71	71	71	71
Example Count	7822	7809	7829	7821	7811
Min	0	0	0	0	0
Max	83	83	83	83	83
Q1	64	64	64	64	64
Median	70	70	70	70	70
Q3	72	72	72	72	72

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.92	63.11	63.11	63.13	63.11
StDev	17.34	16.87	16.87	16.77	16.83
Mode	71	71	71	71	71
Example Count	8725	7828	7848	7840	7830
Min	0	0	0	0	0
Max	83	83	83	83	83
Q1	61	64	64	64	64
Median	69	70	70	70	70
Q3	72	72	72	72	72

Validation	Without Social Data	With Social Data
Mean	60.78	59.06
StDev	16.85	17.72
Mode	70	70
Example Count	1992	2257
Min	12	12
Max	78	78
Q1	61	55
Median	68	67
Q3	70	70

Sensor 155
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.94	61.93	61.96	62.13	61.96
StDev	9.63	9.67	9.55	9.01	9.59
Mode	64	64	64	64	64
Example Count	1926	1903	1915	1902	1900
Min	0	0	0	0	0
Max	70	70	70	70	70

Q1	62	62	62	62	62
Median	64	64	64	64	64
Q3	65	65	65	65	65

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.86	61.93	61.96	62.13	61.96
StDev	9.77	9.67	9.55	9.01	9.59
Mode	64	64	64	64	64
Example Count	1930	1903	1915	1902	1900
Min	0	0	0	0	0
Max	70	70	70	70	70
Q1	62	62	62	62	62
Median	64	64	64	64	64
Q3	65	65	65	65	65

Validation	Without Social Data	With Social Data
Mean	62.96	62.96
StDev	1.36	1.36
Mode	63	63
Example Count	515	515
Min	57	57
Max	67	67
Q1	62	62
Median	63	63
Q3	64	64

Sensor 161
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	24.85	24.95	25.00	24.91	24.97
StDev	22.15	22.16	22.12	22.13	22.13
Mode	0	0	0	0	0
Example Count	7530	7514	7541	7520	7521
Min	0	0	0	0	0
Max	87	87	87	87	87
Q1	0	0	0	0	0
Median	27	27	28	27	28
Q3	47	48	48	48	48

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	25.13	24.96	25.00	24.91	24.98
StDev	21.54	22.14	22.11	22.12	22.12
Mode	0	0	0	0	0
Example Count	8399	7532	7559	7538	7539
Min	0	0	0	0	0
Max	87	87	87	87	87
Q1	0	0	0	0	0
Median	26	27	27	27	27
Q3	47	48	48	48	47

Validation	Without Social Data	With Social Data
Mean	22.91	23.10
StDev	21.33	20.76
Mode	0	0
Example Count	1900	2134
Min	0	0
Max	61	61
Q1	0	0
Median	22	23
Q3	46	46

Sensor 168

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	53.77	53.78	53.86	53.89	53.75
StDev	11.12	11.19	10.99	10.96	11.17
Mode	57	57	57	57	57
Example Count	2470	2450	2468	2453	2458
Min	0	0	0	0	0
Max	67	67	67	67	67
Q1	54	55	55	55	54
Median	57	57	57	57	57
Q3	58	59	59	59	59

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	53.07	53.75	53.84	53.87	53.72
StDev	11.67	11.19	10.99	10.97	11.17
Mode	57	57	57	57	57
Example Count	2587	2455	2473	2458	2463

Min	0	0	0	0	0
Max	67	67	67	67	67
Q1	54	55	55	55	54
Median	57	57	57	57	57
Q3	58	59	59	59	59

Validation	Without Social Data	With Social Data
Mean	55.00	54.98
StDev	6.48	6.49
Mode	55	55
Example Count	575	576
Min	12	12
Max	64	64
Q1	55	54
Median	56	56
Q3	58	58

Sensor 169

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	54.79	54.78	54.78	54.75	54.75
StDev	7.31	7.32	7.34	7.38	7.34
Mode	57	57	57	57	57
Example Count	7915	7903	7925	7915	7905
Min	0	0	0	0	0
Max	71	71	71	71	71
Q1	55	55	55	55	54
Median	57	57	57	57	57
Q3	58	58	58	58	58

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	53.66	54.75	54.74	54.72	54.72
StDev	8.65	7.36	7.39	7.42	7.39
Mode	57	57	57	57	57
Example Count	8915	7923	7945	7935	7925
Min	0	0	0	0	0
Max	71	71	71	71	71
Q1	54	55	55	54	54
Median	56	57	57	57	57
Q3	58	58	58	58	58

Validation	Without Social Data	With Social Data
Mean	54.02	53.34
StDev	7.06	7.53
Mode	57	57
Example Count	1997	2265
Min	0	0
Max	67	67
Q1	54	53
Median	56	56
Q3	58	58

Sensor 176

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	53.74	53.68	53.71	53.72	53.70
StDev	7.50	7.69	7.60	7.57	7.63
Mode	56	56	56	56	56
Example Count	6918	6913	6932	6913	6911
Min	0	0	0	0	0
Max	77	77	77	77	77
Q1	54	54	54	54	54
Median	56	56	56	56	56
Q3	57	57	57	57	57

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	53.66	53.67	53.72	53.72	53.70
StDev	7.49	7.70	7.60	7.58	7.64
Mode	56	56	56	56	56
Example Count	7360	6921	6940	6921	6919
Min	0	0	0	0	0
Max	77	77	77	77	77
Q1	53	54	54	54	54
Median	55	56	56	56	56
Q3	57	57	57	57	57

Validation	Without Social Data	With Social Data
Mean	52.26	51.83
StDev	8.24	8.61

Mode	56	56
Example Count	1753	1910
Min	0	0
Max	66	66
Q1	52	52
Median	55	54
Q3	56	56

Sensor 179

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	59.67	59.65	59.69	59.65	59.67
StDev	8.53	8.57	8.55	8.59	8.56
Mode	64	64	64	64	64
Example Count	6286	6281	6297	6303	6260
Min	12	12	12	12	12
Max	68	68	68	68	68
Q1	59	59	59	59	59
Median	62	62	62	62	62
Q3	64	64	64	64	64

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	59.32	59.64	59.68	59.64	59.65
StDev	8.99	8.60	8.56	8.62	8.58
Mode	64	64	64	64	64
Example Count	6738	6289	6304	6311	6268
Min	12	12	12	12	12
Max	68	68	68	68	68
Q1	59	59	59	59	59
Median	62	62	62	62	62
Q3	64	64	64	64	64

Validation	Without Social Data	With Social Data
Mean	58.31	57.71
StDev	9.34	9.72
Mode	63	63
Example Count	1791	1963
Min	13	13
Max	68	68
Q1	57	56

Median	62	61
Q3	64	63

Sensor 200
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.90	61.91	61.92	61.91	61.92
StDev	4.04	3.97	3.99	4.00	3.96
Mode	63	63	63	63	63
Example Count	6830	6825	6848	6829	6823
Min	0	0	0	0	0
Max	67	67	67	67	67
Q1	62	62	62	62	62
Median	63	63	63	63	63
Q3	63	63	63	63	63

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	61.72	61.91	61.91	61.91	61.91
StDev	4.57	3.98	4.00	4.01	3.97
Mode	63	63	63	63	63
Example Count	7269	6831	6855	6835	6829
Min	0	0	0	0	0
Max	67	67	67	67	67
Q1	62	62	62	62	62
Median	63	63	63	63	63
Q3	63	63	63	63	63

Validation	Without Social Data	With Social Data
Mean	61.29	60.89
StDev	4.96	5.45
Mode	63	63
Example Count	1626	1758
Min	22	22
Max	67	67
Q1	61	61
Median	62	62
Q3	63	63

Sensor 206
Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	55.92	55.80	55.95	55.77	55.78
StDev	16.83	16.94	16.82	16.97	16.93
Mode	65	64	65	64	65
Example Count	1154	1139	1131	1132	1146
Min	12	12	12	12	12
Max	76	76	76	76	76
Q1	59	59	59	59	59
Median	64	64	64	64	64
Q3	66	66	66	66	66

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	54.24	55.77	55.92	55.74	55.75
StDev	18.17	16.96	16.84	16.99	16.95
Mode	65	64	65	64	65
Example Count	1210	1140	1132	1133	1147
Min	12	12	12	12	12
Max	76	76	76	76	76
Q1	41	59	59	59	59
Median	64	64	64	64	64
Q3	66	66	66	66	66

Validation	Without Social Data	With Social Data
Mean	48.32	46.99
StDev	18.99	19.30
Mode	64	64
Example Count	356	377
Min	12	12
Max	73	73
Q1	29	28
Median	61	60
Q3	65	64

Sensor 239

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	57.88	58.00	57.87	57.91	57.93
StDev	10.82	10.64	10.83	10.85	10.81
Mode	62	62	62	62	62
Example Count	1479	1446	1448	1458	1453

Min	0	0	0	0	0
Max	77	77	77	77	77
Q1	58	58	58	58	58
Median	61	61	61	61	61
Q3	64	64	64	64	64

Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	56.75	57.98	57.84	57.87	57.89
StDev	11.73	10.65	10.87	10.89	10.85
Mode	62	62	62	62	62
Example Count	1564	1447	1450	1460	1455
Min	0	0	0	0	0
Max	77	77	77	77	77
Q1	56	58	58	58	58
Median	61	61	61	61	61
Q3	63	64	64	64	64

Validation	Without Social Data	With Social Data
Mean	56.97	55.02
StDev	9.23	10.30
Mode	62	62
Example Count	302	340
Min	28	28
Max	71	71
Q1	55	44
Median	61	60
Q3	63	63

Sensor 242

Training Without Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	47.03	47.35	47.26	47.37	47.20
StDev	30.60	30.52	30.53	30.52	30.58
Mode	0	0	0	0	0
Example Count	4064	4056	4078	4047	4059
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	0	0	0	0	0
Median	64	64	64	64	64
Q3	70	70	70	70	70

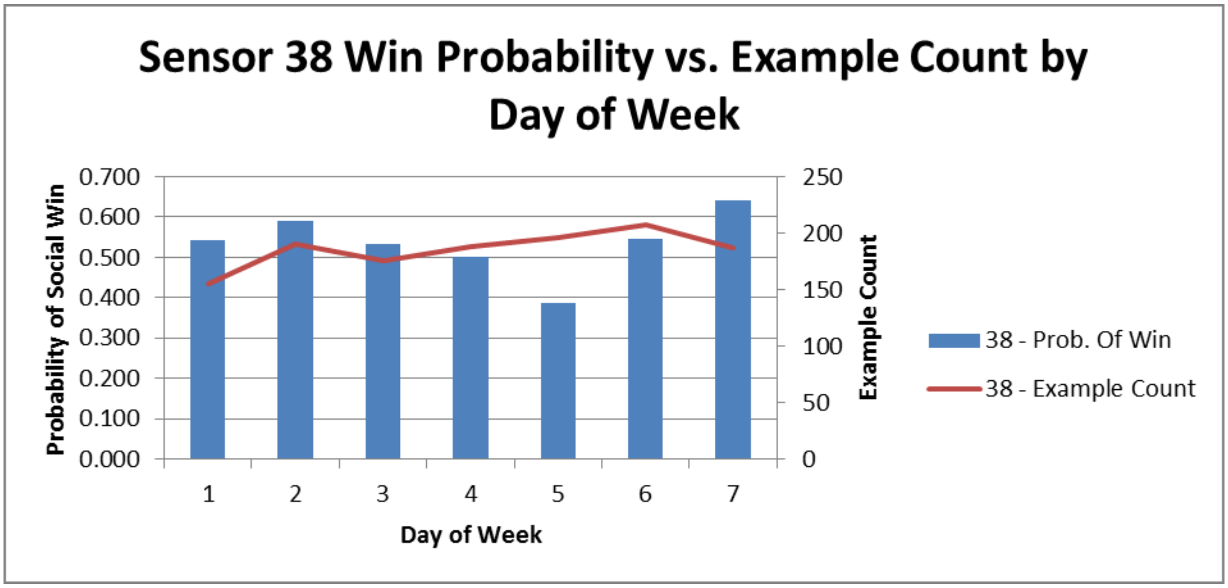
Training With Social Data

Metric	Fold1	Fold2	Fold3	Fold4	Fold5
Mean	47.19	47.34	47.25	47.36	47.19
StDev	29.88	30.50	30.52	30.50	30.57
Mode	0	0	0	0	0
Example Count	4354	4060	4083	4051	4063
Min	0	0	0	0	0
Max	100	100	100	100	100
Q1	0	0	0	0	0
Median	63	64	64	64	64
Q3	70	70	70	70	70

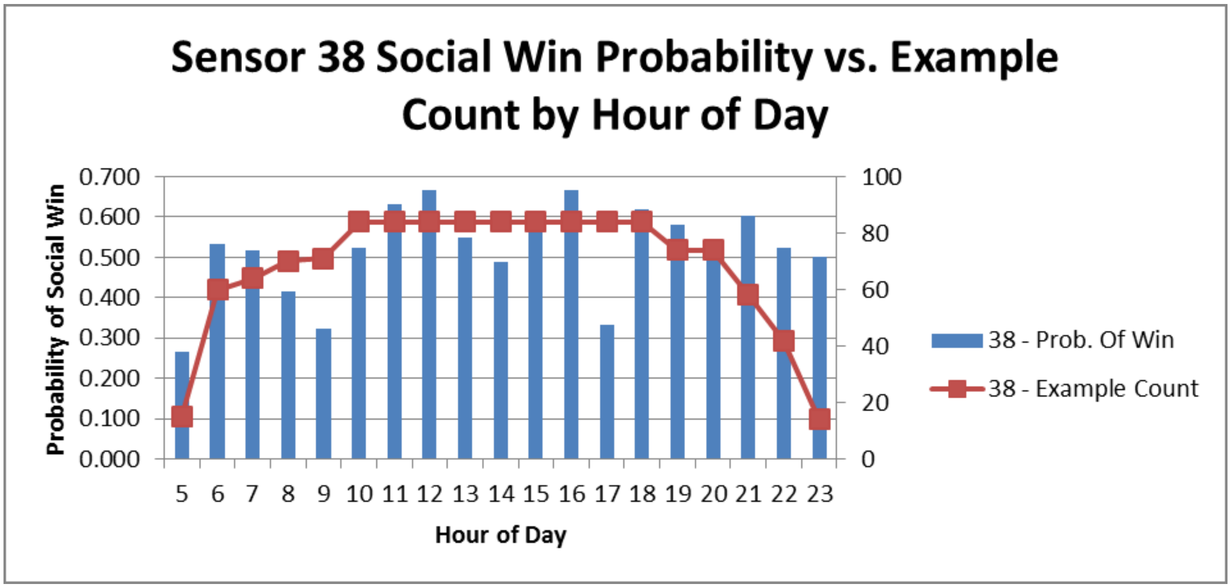
Validation	Without Social Data	With Social Data
Mean	46.51	46.60
StDev	30.47	29.24
Mode	0	0
Example Count	849	933
Min	0	0
Max	100	100
Q1	0	0
Median	63	61
Q3	70	69

Appendix 4 – Detailed Result Breakdown

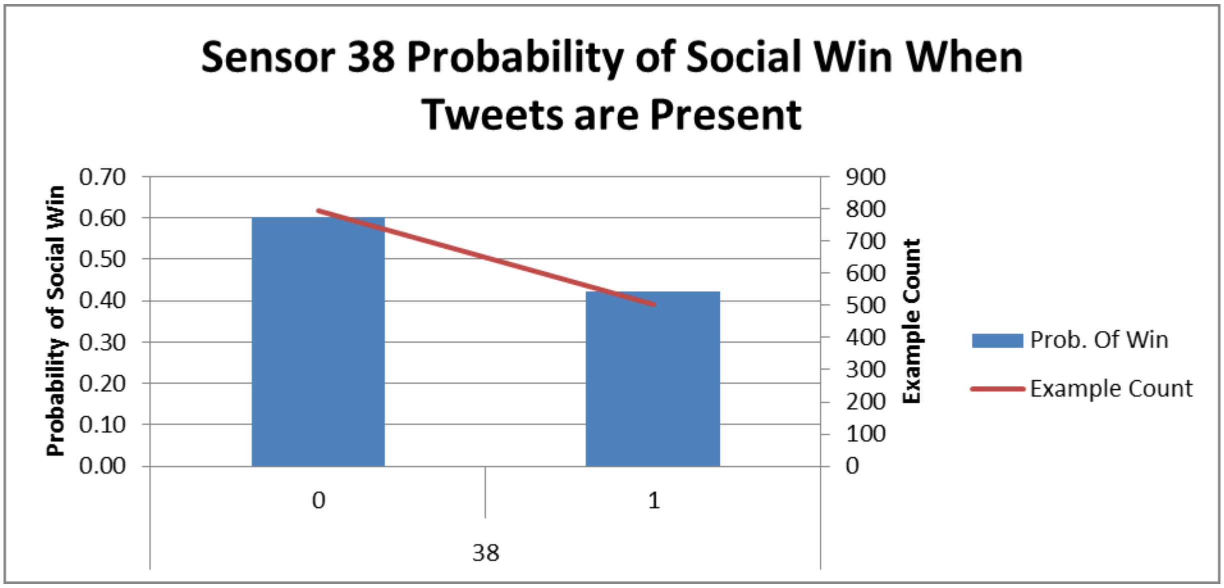
Sensor 38



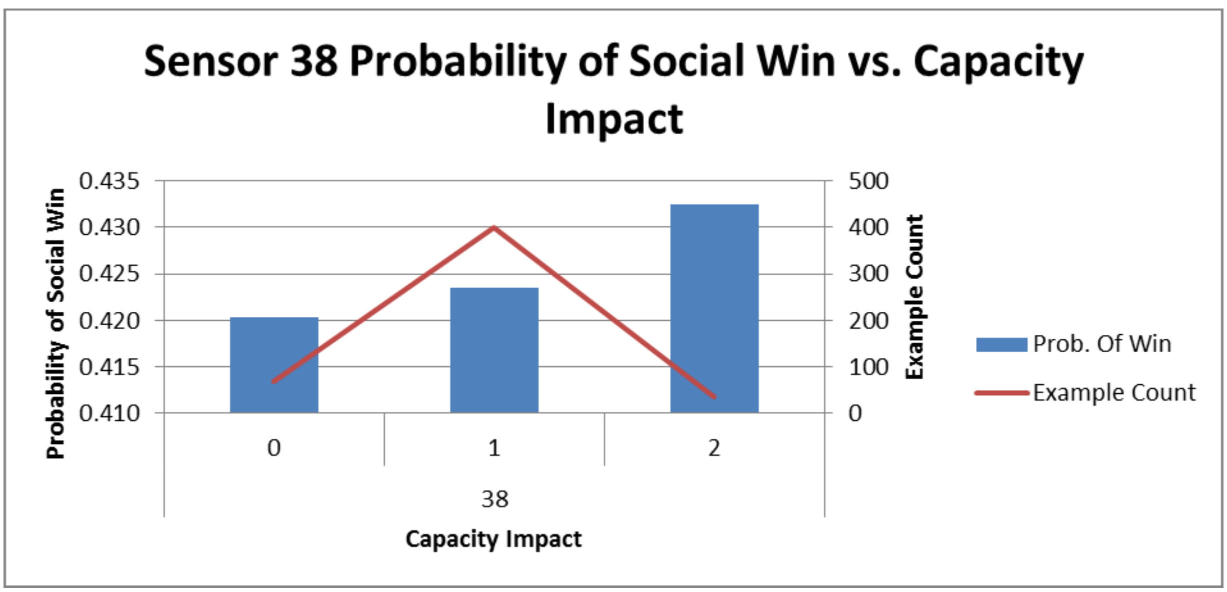
Day of Week	Prob. Of Win	Example Count
1	0.542	155
2	0.589	190
3	0.531	175
4	0.500	188
5	0.388	196
6	0.546	207
7	0.642	187



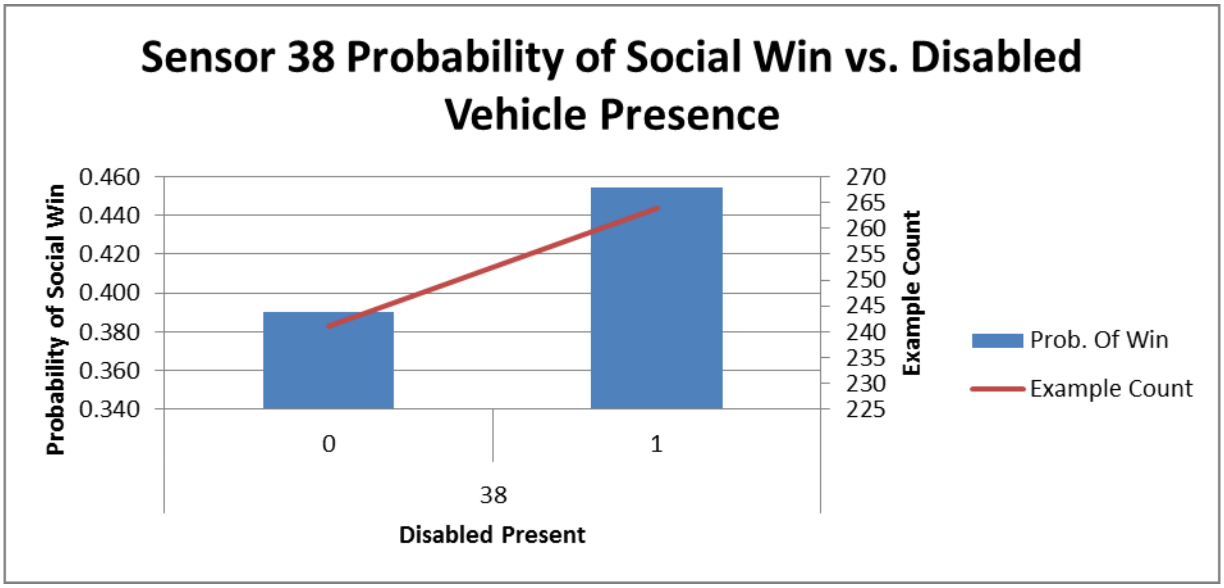
Hour of Day	Prob. Of Win	Example Count
5	0.267	15
6	0.533	60
7	0.516	64
8	0.414	70
9	0.324	71
10	0.524	84
11	0.631	84
12	0.667	84
13	0.548	84
14	0.488	84
15	0.583	84
16	0.667	84
17	0.333	84
18	0.619	84
19	0.581	74
20	0.527	74
21	0.603	58
22	0.524	42
23	0.500	14



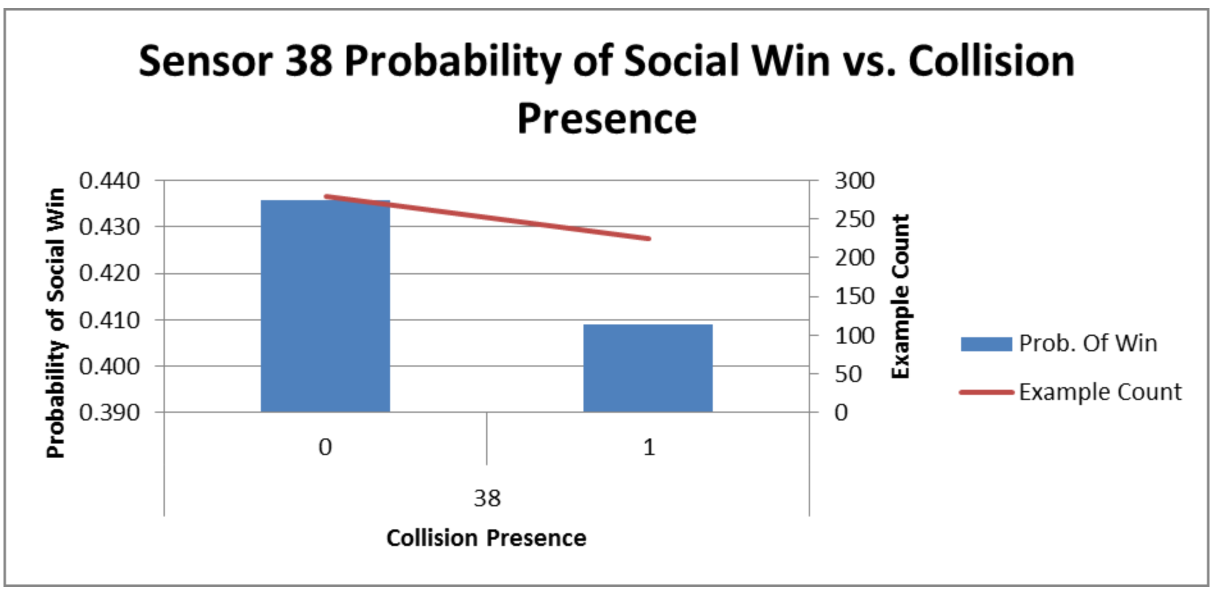
Row Labels	Prob. Of Win	Example Count
0	0.60	793
1	0.42	505



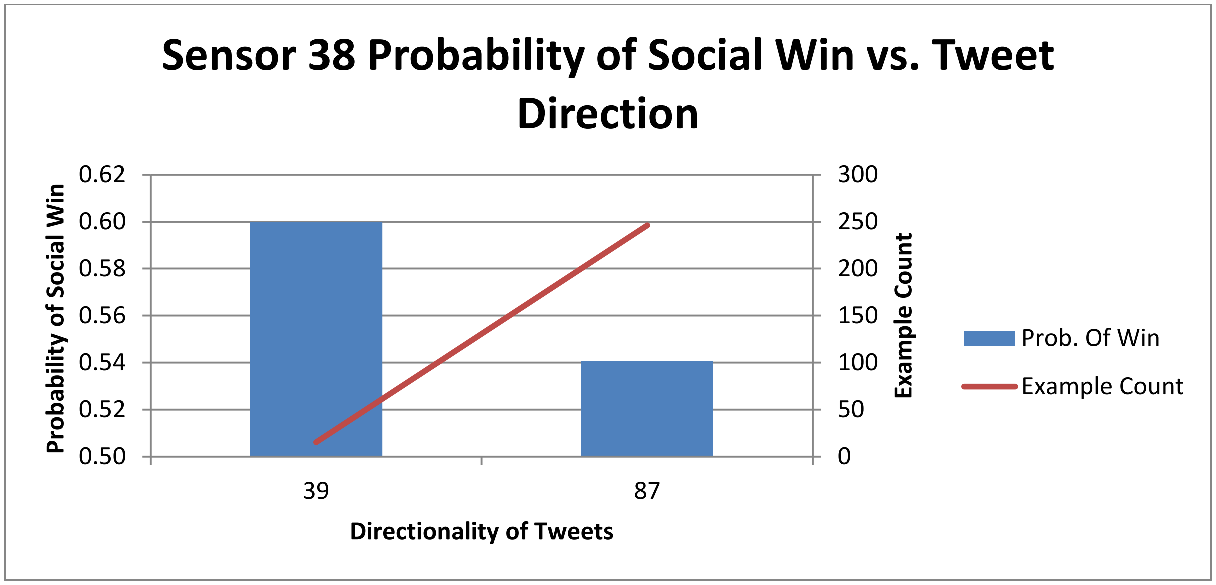
Row Labels	Prob. Of Win	Example Count
0	0.420	69
1	0.424	399
2	0.432	37



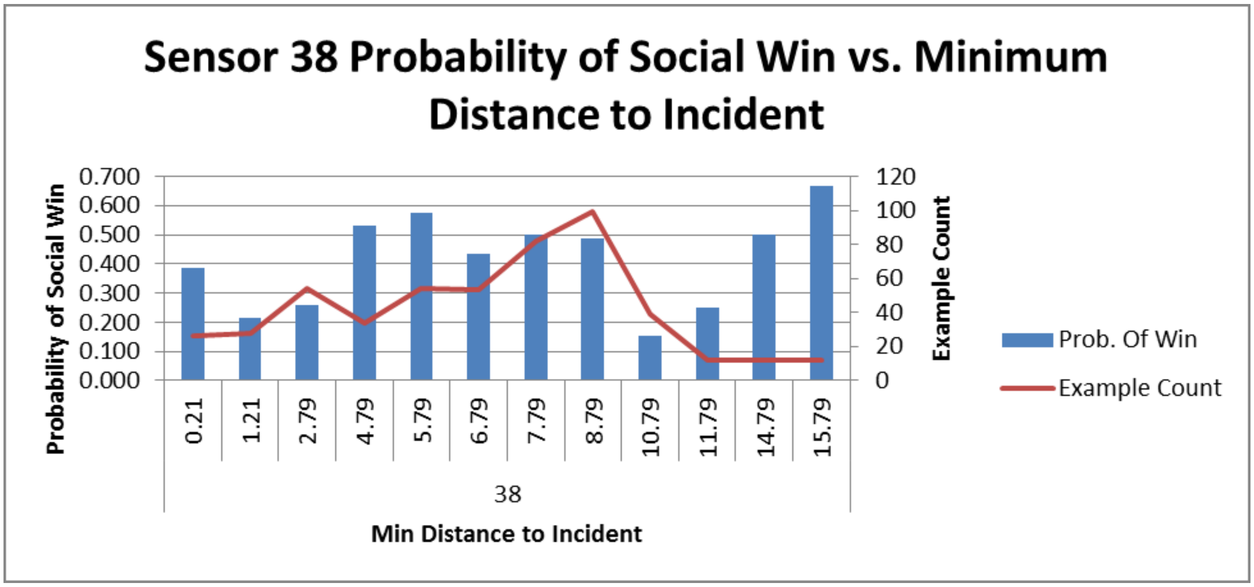
Row Labels	Prob. Of Win	Example Count
0	0.390	241
1	0.455	264



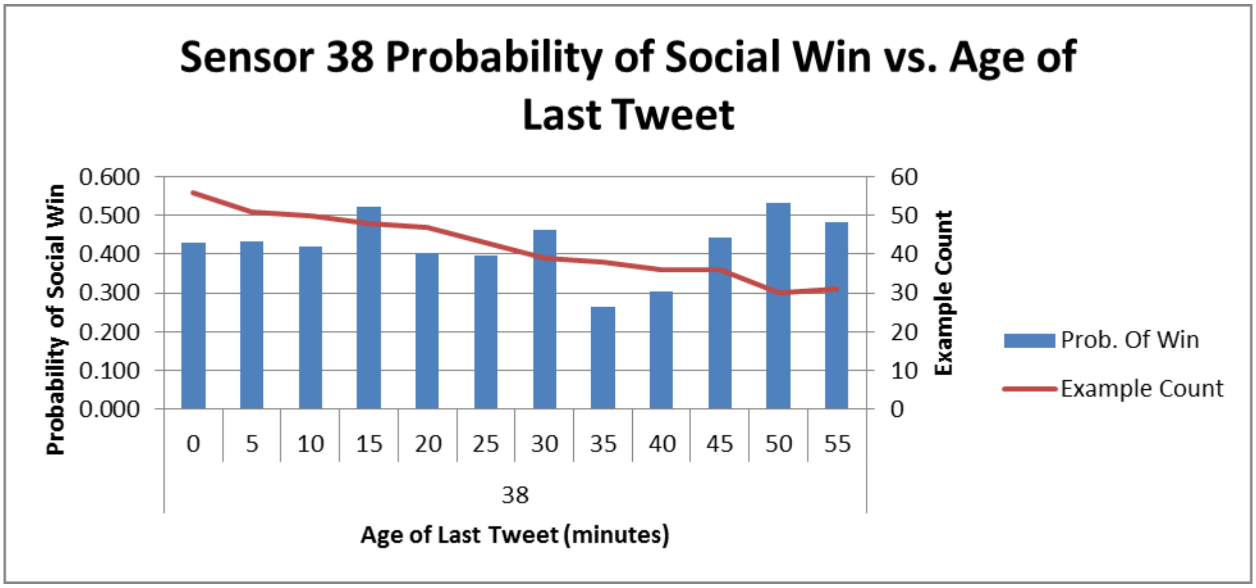
Row Labels	Prob. Of Win	Example Count
0	0.436	280
1	0.409	225



Row Labels	Prob. Of Win	Example Count
0	0.45	197
0.25	0.00	1
0.333	0.00	2
0.4	0.00	6
0.5	0.48	42
0.75	0.00	7
1	0.42	250

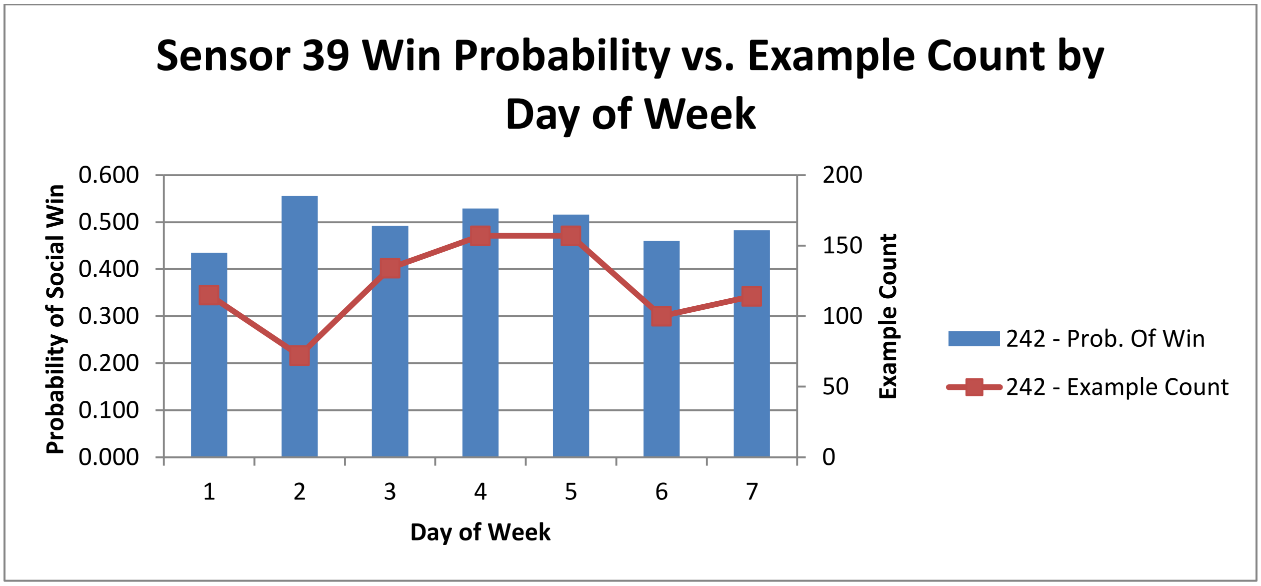


Row Labels	Prob. Of Win	Example Count
0.21	0.385	26
1.21	0.214	28
2.79	0.259	54
4.79	0.529	34
5.79	0.574	54
6.79	0.434	53
7.79	0.500	82
8.79	0.485	99
10.79	0.154	39
11.79	0.250	12
14.79	0.500	12
15.79	0.667	12

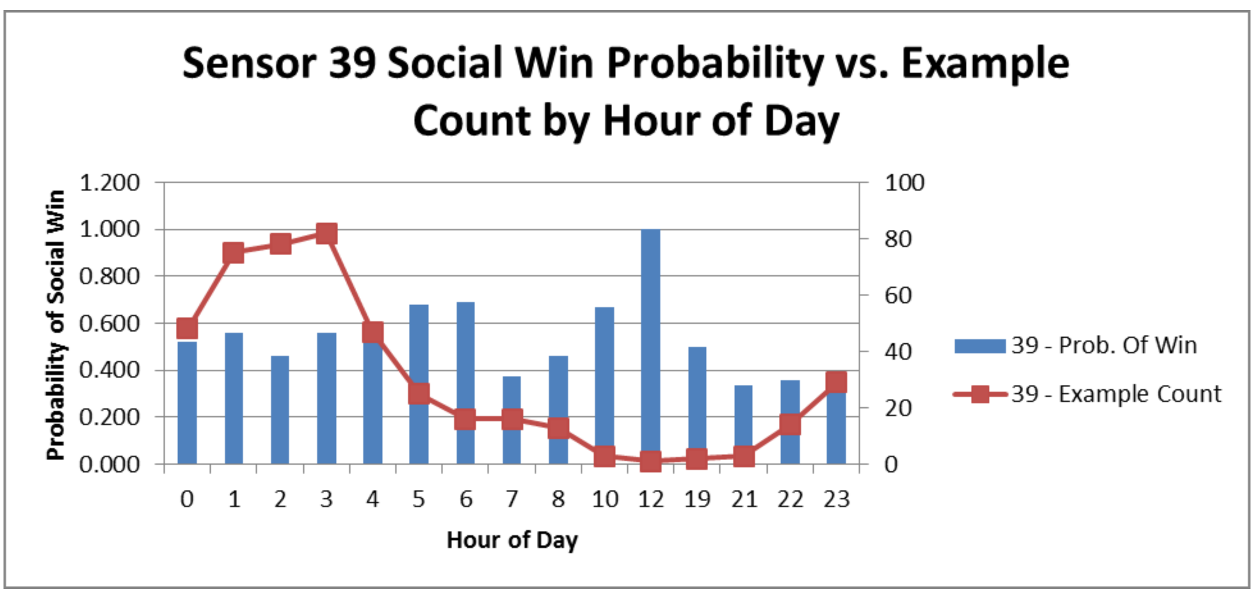


Row Labels	Prob. Of Win	Example Count
0	0.429	56
5	0.431	51
10	0.420	50
15	0.521	48
20	0.404	47
25	0.395	43
30	0.462	39
35	0.263	38
40	0.306	36
45	0.444	36
50	0.533	30
55	0.484	31

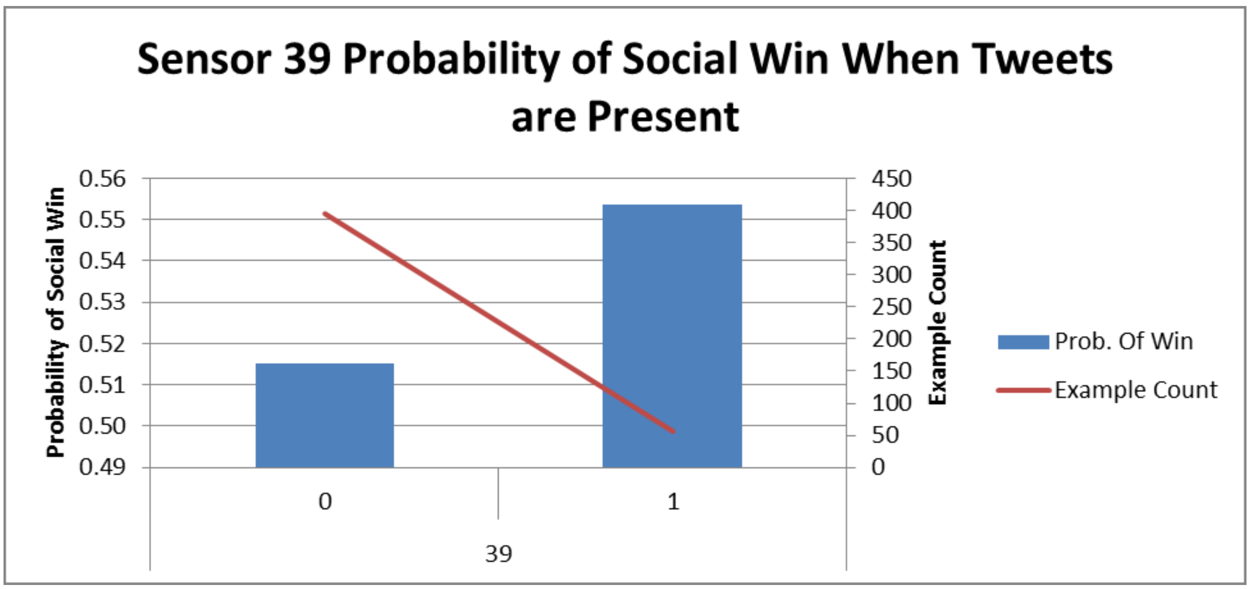
Sensor 39



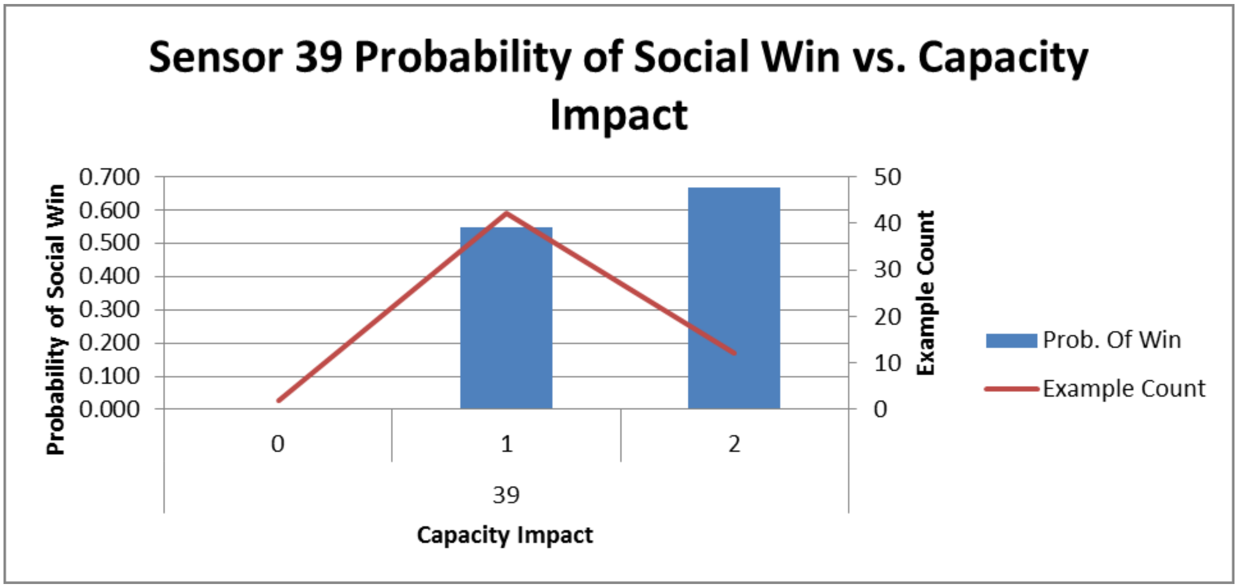
Row Labels	Prob. Of Win	Example Count
1	0.568	95
2	0.348	46
3	0.528	53
4	0.379	58
5	0.481	77
6	0.588	51
7	0.667	72



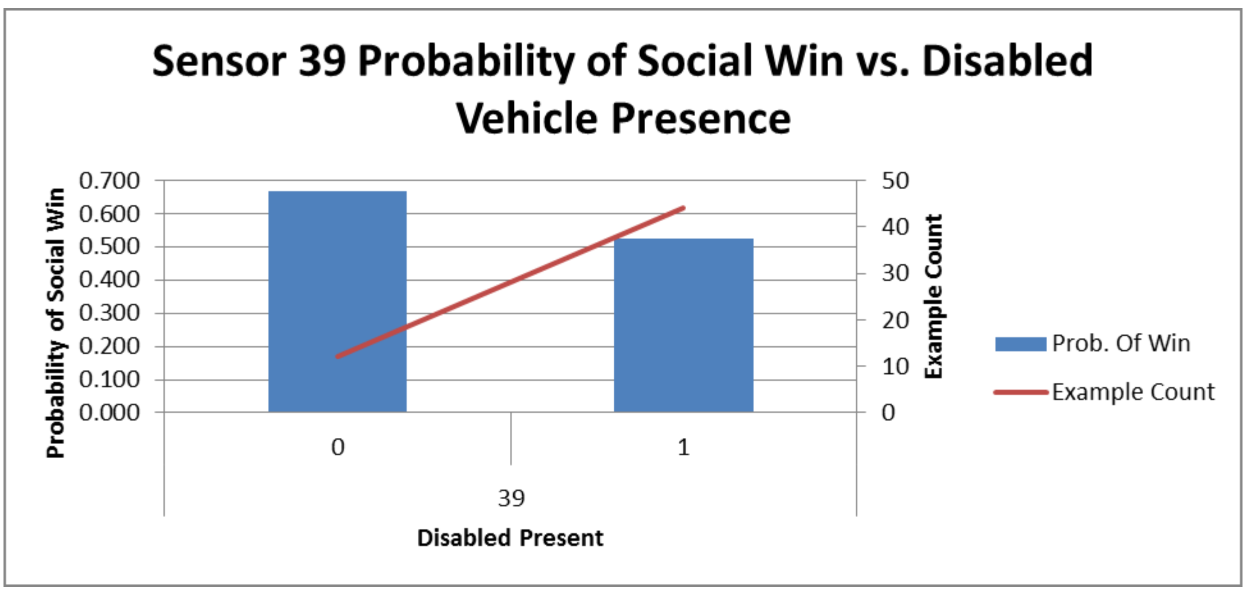
Row Labels	Prob. Of Win	Example Count
0	0.521	48
1	0.560	75
2	0.462	78
3	0.561	82
4	0.532	47
5	0.680	25
6	0.688	16
7	0.375	16
8	0.462	13
10	0.667	3
12	1.000	1
19	0.500	2
21	0.333	3
22	0.357	14
23	0.379	29



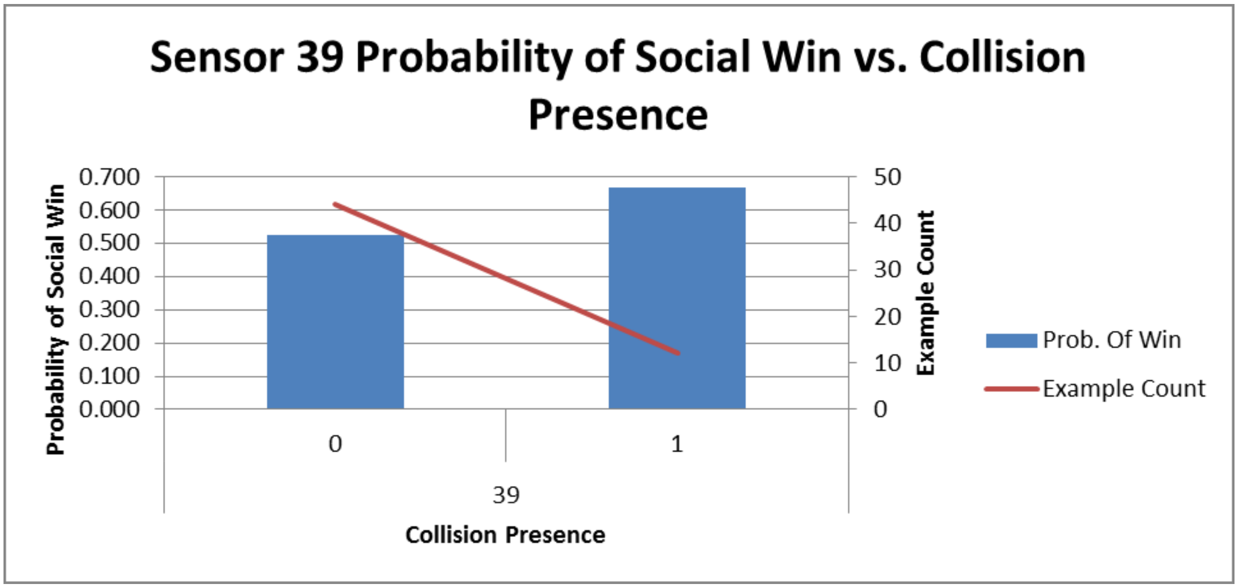
Row Labels	Prob. Of Win	Example Count
0	0.52	396
1	0.55	56



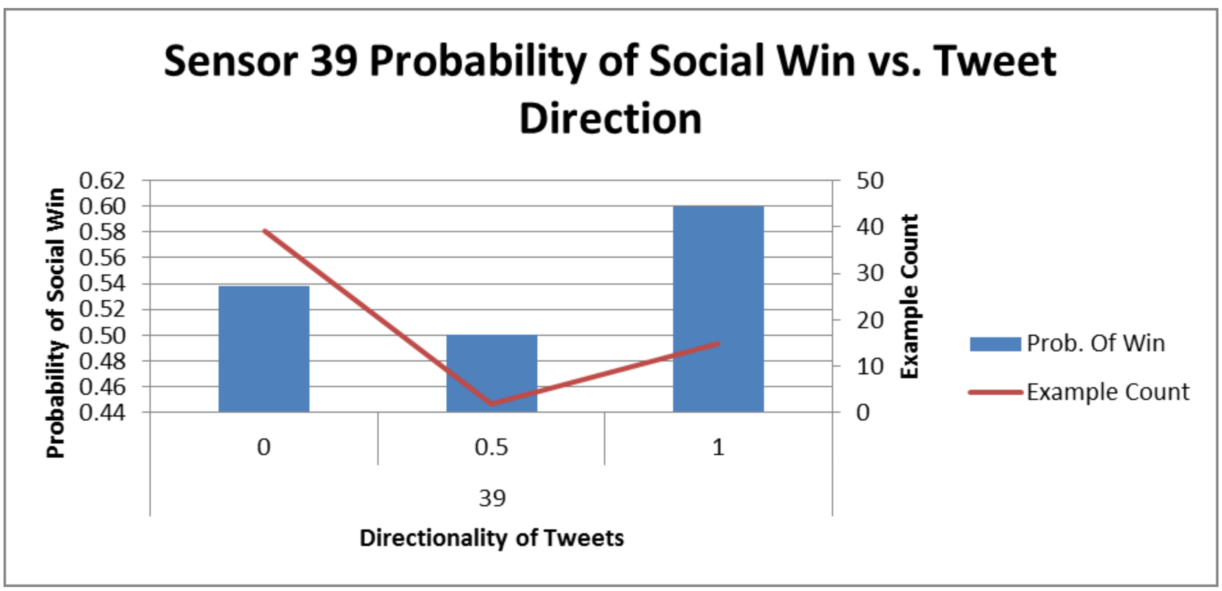
Row Labels	Prob. Of Win	Example Count
0	0.000	2
1	0.548	42
2	0.667	12



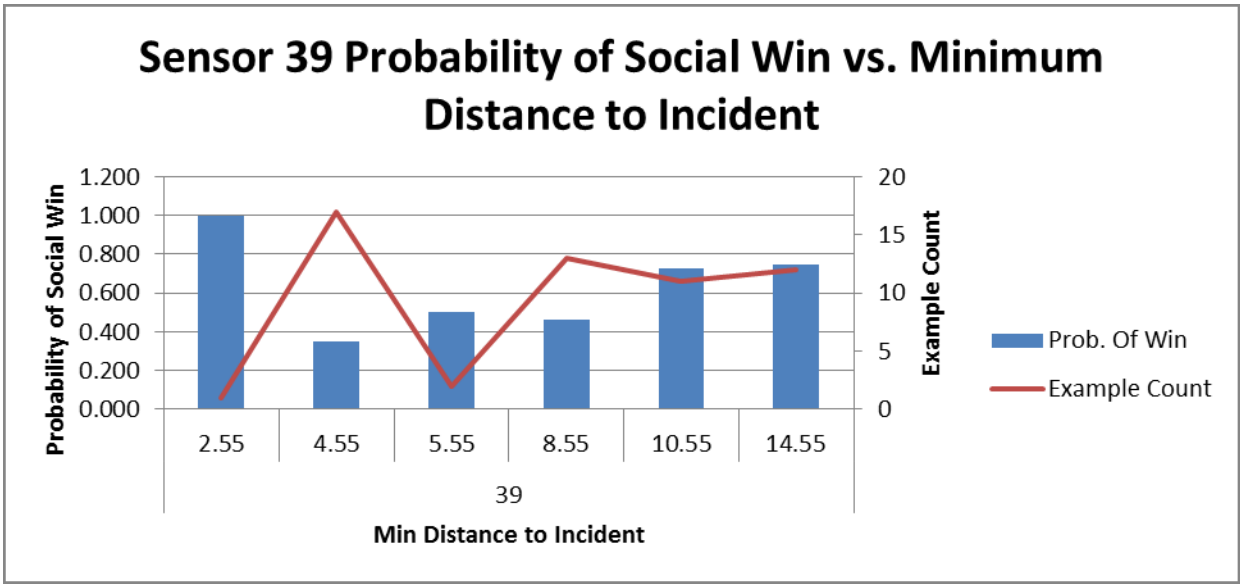
Row Labels	Prob. Of Win	Example Count
0	0.667	12
1	0.523	44



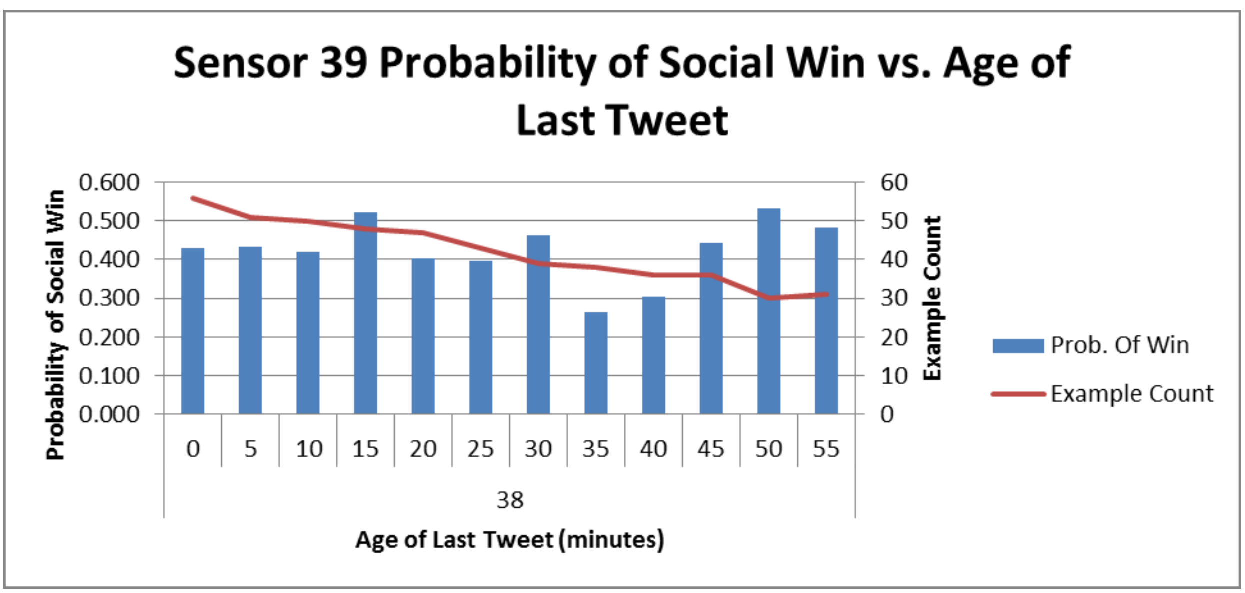
Row Labels	Prob. Of Win	Example Count
0	0.523	44
1	0.667	12



Row Labels	Prob. Of Win	Example Count
0	0.54	39
0.5	0.50	2
1	0.60	15

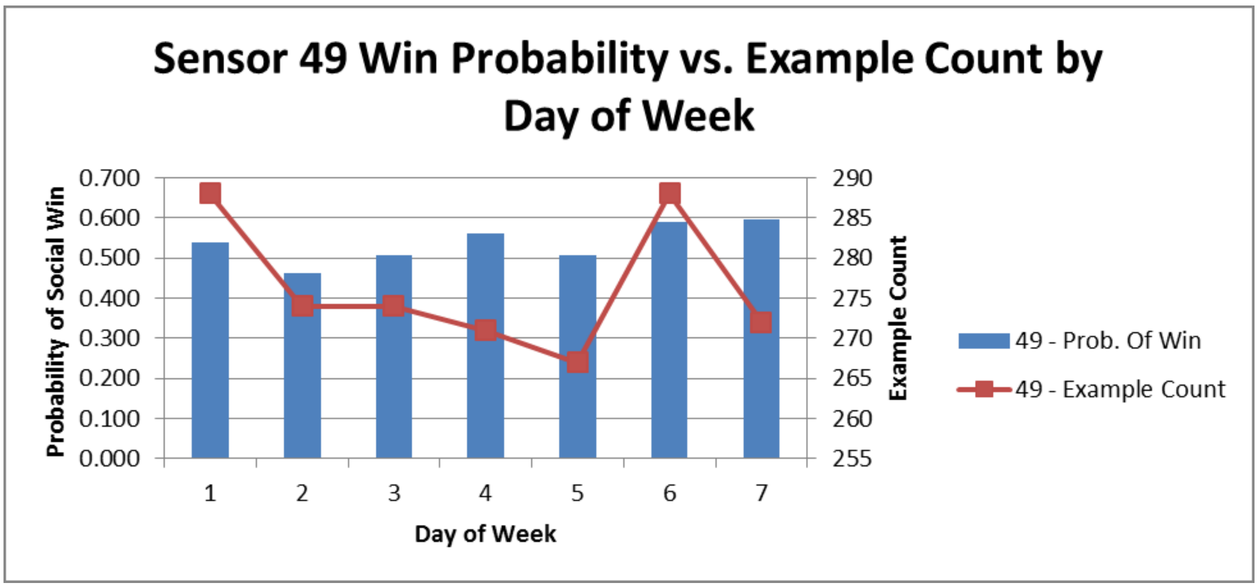


Row Labels	Prob. Of Win	Example Count
2.55	1.000	1
4.55	0.353	17
5.55	0.500	2
8.55	0.462	13
10.55	0.727	11
14.55	0.750	12



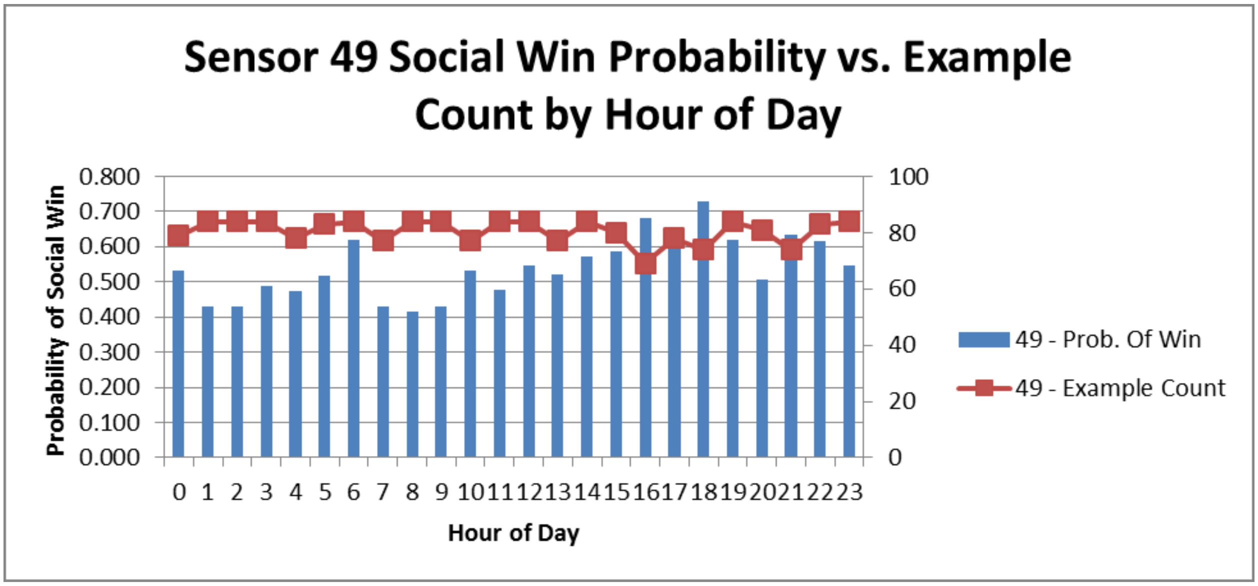
Row Labels	Prob. Of Win	Example Count
0	0.667	6
5	0.600	5
10	0.600	5
15	0.500	4
20	0.500	4
25	0.800	5
30	0.500	4
35	0.500	4
40	0.600	5
45	0.400	5
50	0.200	5
55	0.750	4

Sensor 49



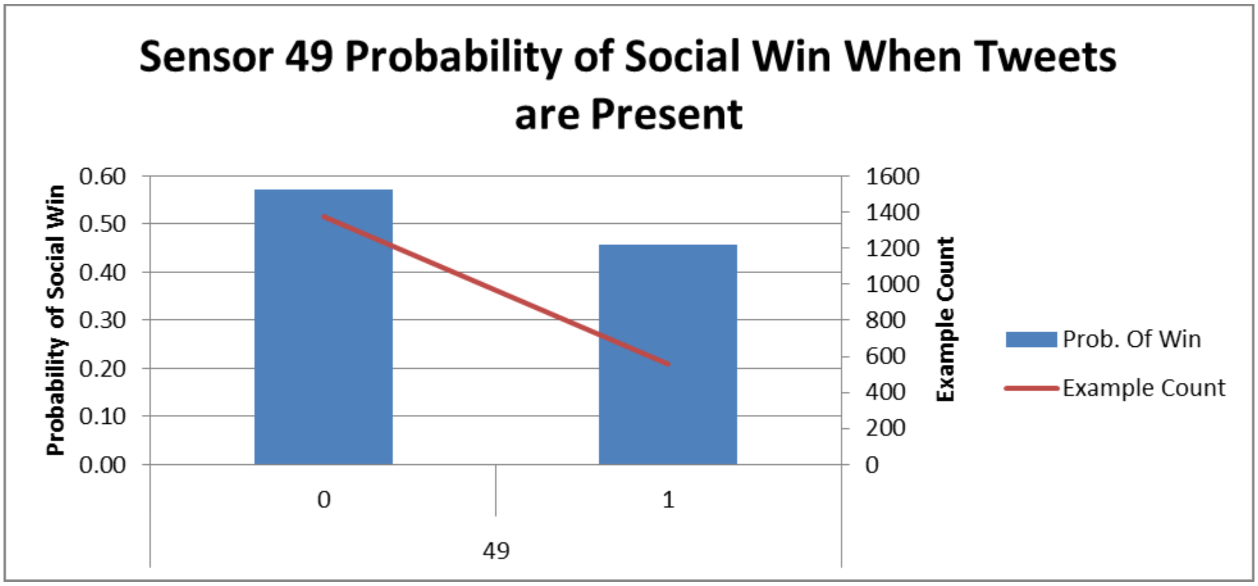
Row Labels	Prob. Of Win	Example Count
1	0.538	288
2	0.464	274
3	0.507	274
4	0.561	271
5	0.506	267
6	0.590	288

7	0.596	272

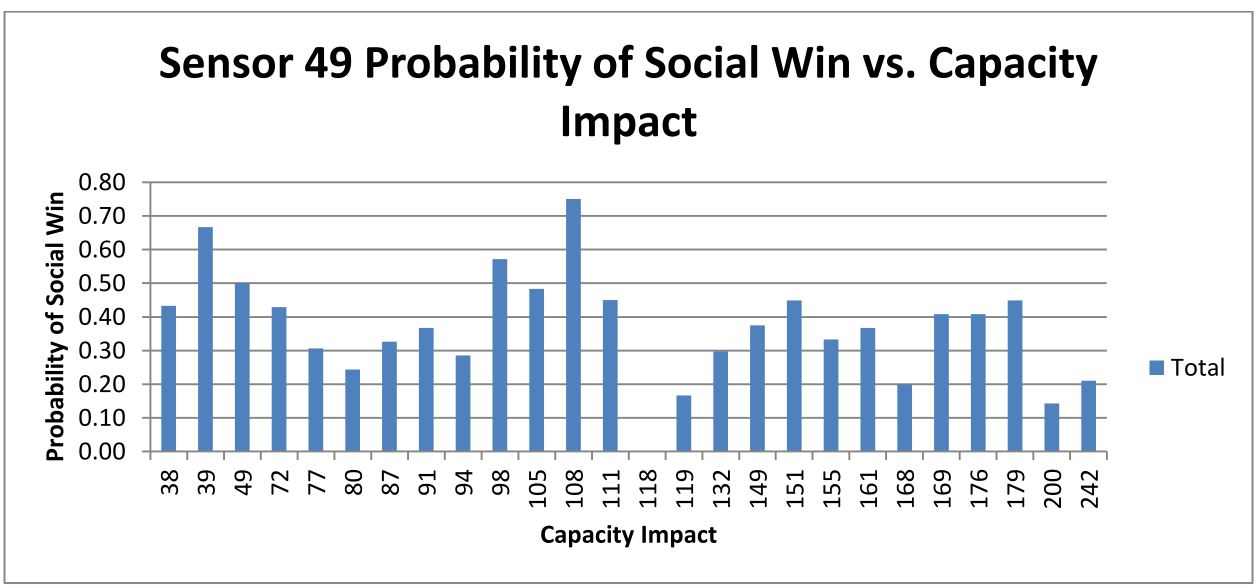


Row Labels	Prob. Of Win	Example Count
0	0.532	79
1	0.429	84
2	0.429	84
3	0.488	84
4	0.474	78
5	0.518	83
6	0.619	84
7	0.429	77
8	0.417	84
9	0.429	84
10	0.532	77
11	0.476	84
12	0.548	84
13	0.519	77
14	0.571	84
15	0.588	80
16	0.681	69
17	0.628	78
18	0.730	74
19	0.619	84
20	0.506	81

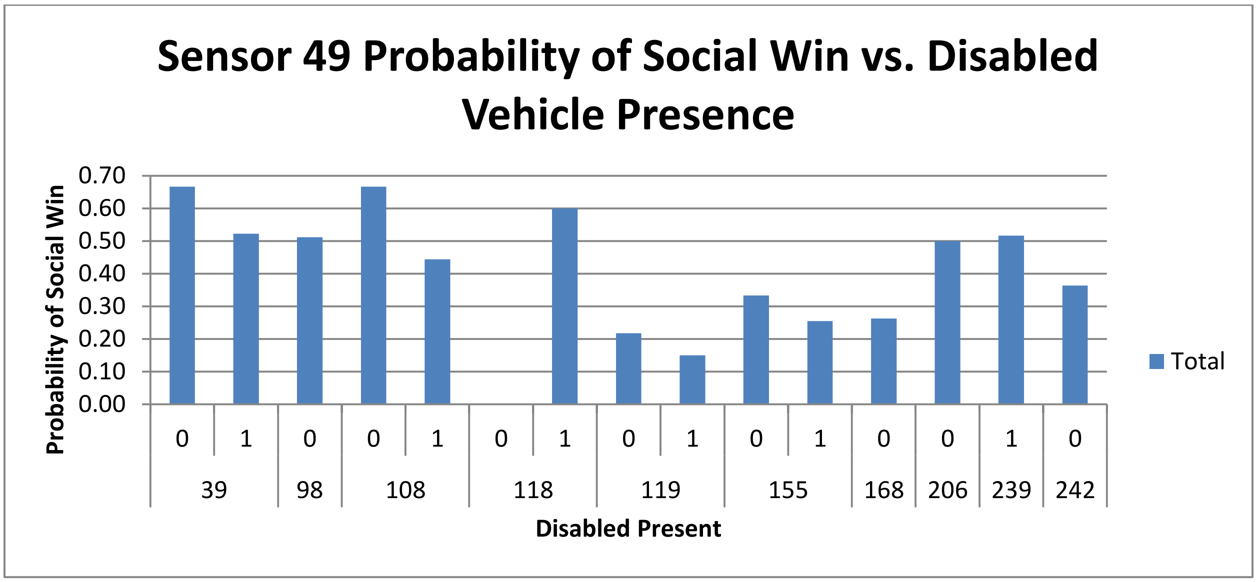
21	0.635	74
22	0.614	83
23	0.548	84



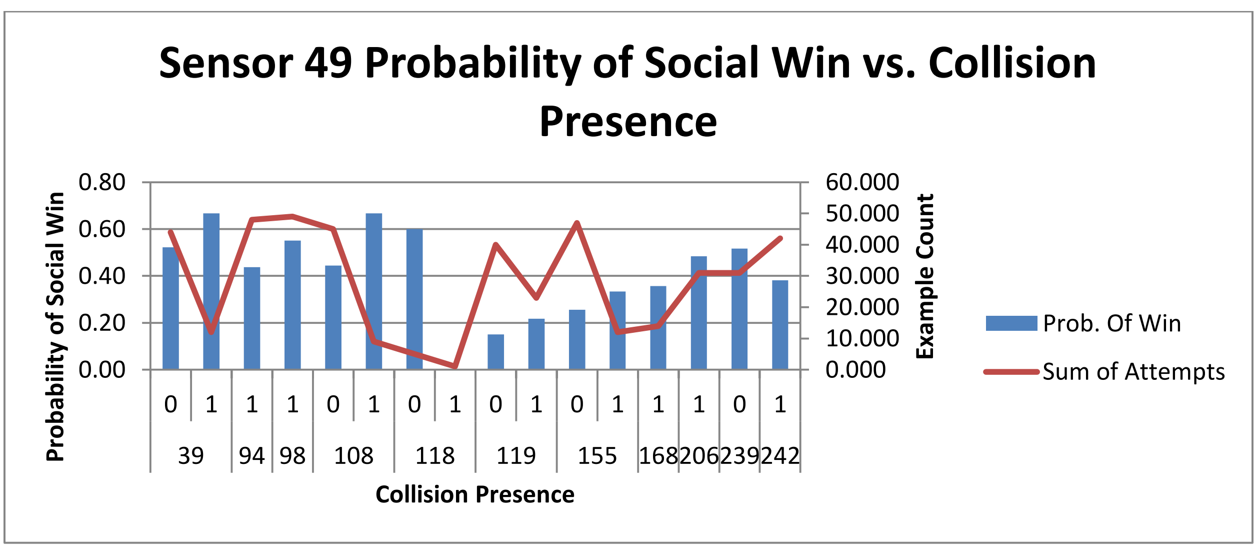
Row Labels	Prob. Of Win	Example Count
0	0.57	1373
1	0.46	561



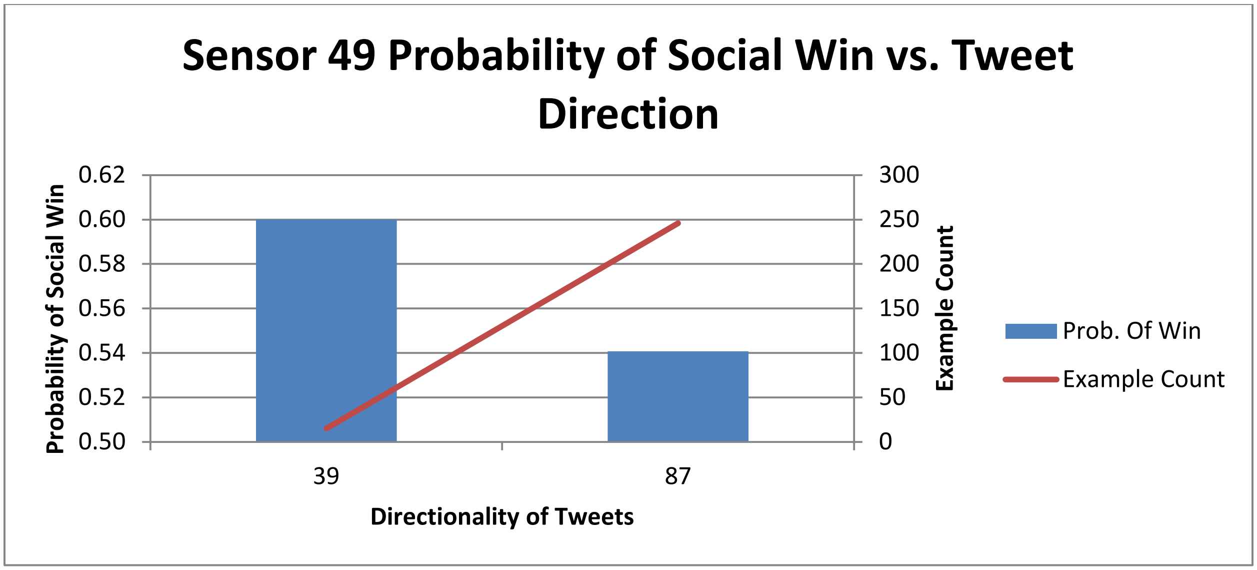
Row Labels	Prob. Of Win	Example Count
0	0.522	69
1	0.442	446
2	0.500	46



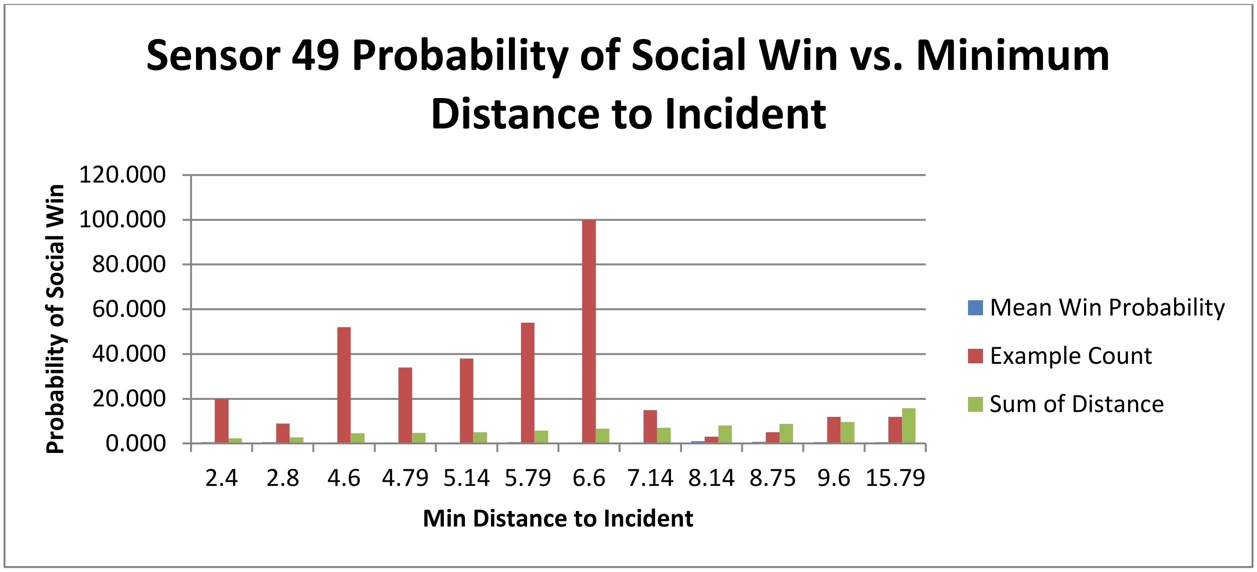
Row Labels	Prob. Of Win	Example Count
0	0.482	251
1	0.435	310



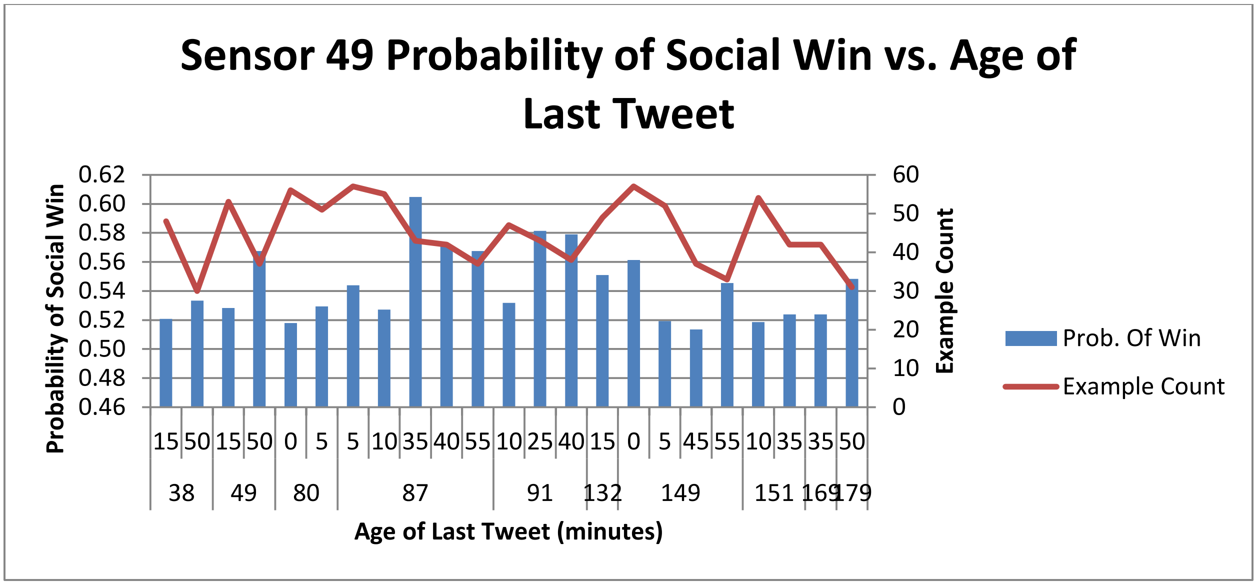
Row Labels	Prob. Of Win	Example Count
0	0.444	322
1	0.473	239



Row Labels	Prob. Of Win	Example Count
0	0.43	240
0.25	0.00	1
0.333	0.50	2
0.4	0.17	6
0.5	0.48	42
0.75	0.00	7
1	0.50	263

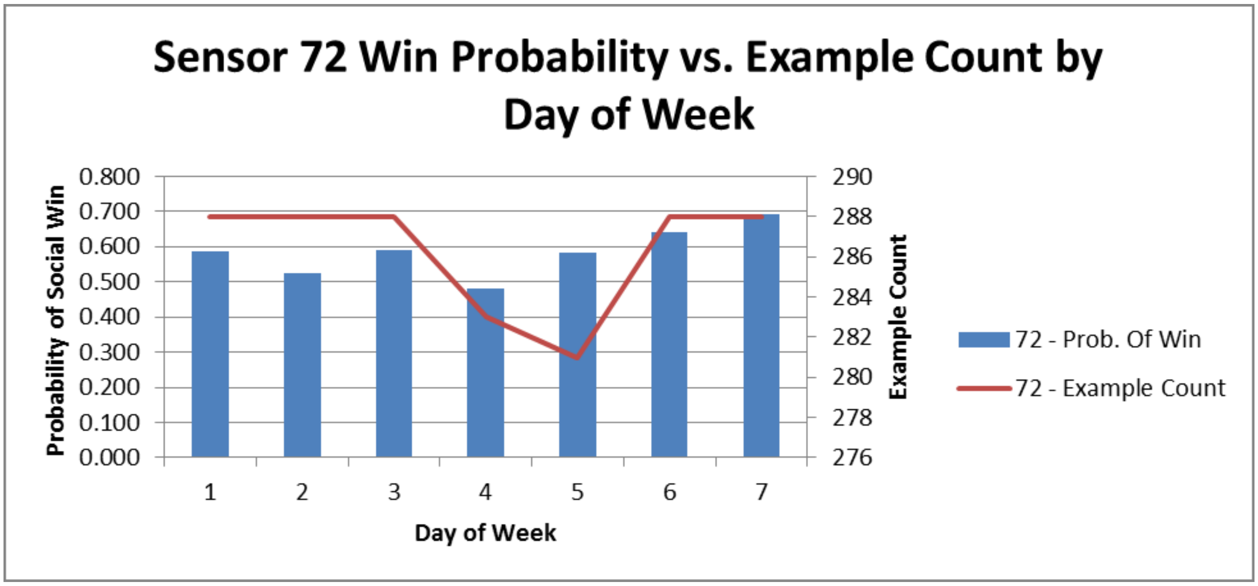


Row Labels	Prob. Of Win	Example Count
0.92	0.538	26
1.92	0.231	26
2.08	0.470	66
4.08	0.283	46
5.08	0.473	55
6.08	0.491	53
7.08	0.388	85
8.08	0.590	105
10.08	0.431	51
11.08	0.417	12
14.08	0.458	24
15.08	0.583	12

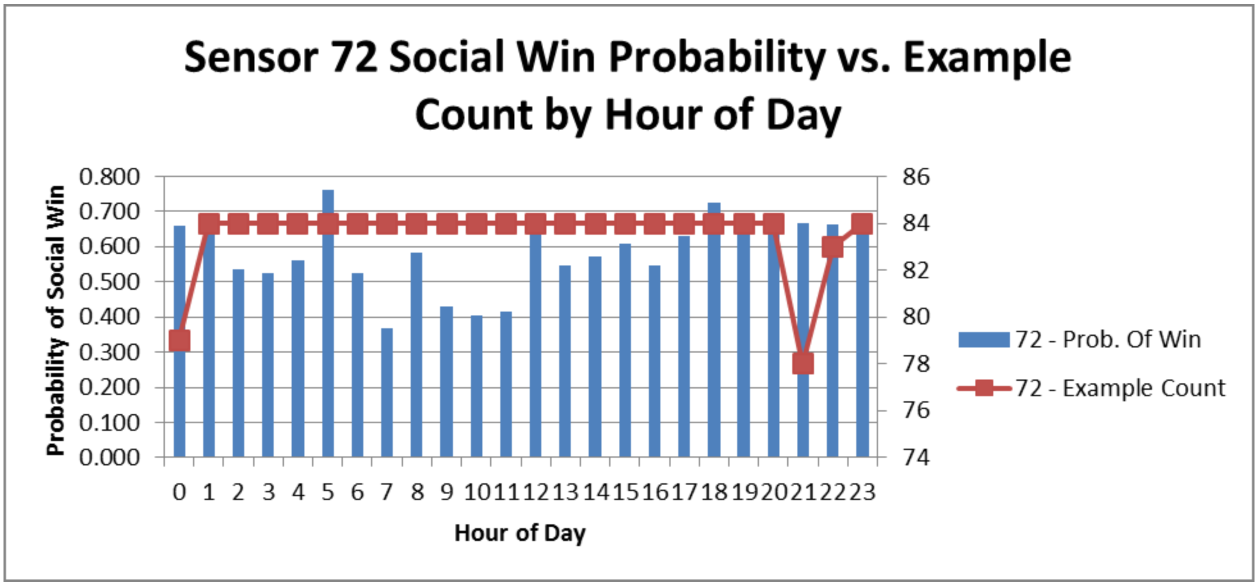


Row Labels	Prob. Of Win	Example Count
0	0.433	60
5	0.411	56
10	0.389	54
15	0.528	53
20	0.462	52
25	0.404	47
30	0.465	43
35	0.415	41
40	0.500	42
45	0.475	40
50	0.568	37
55	0.472	36

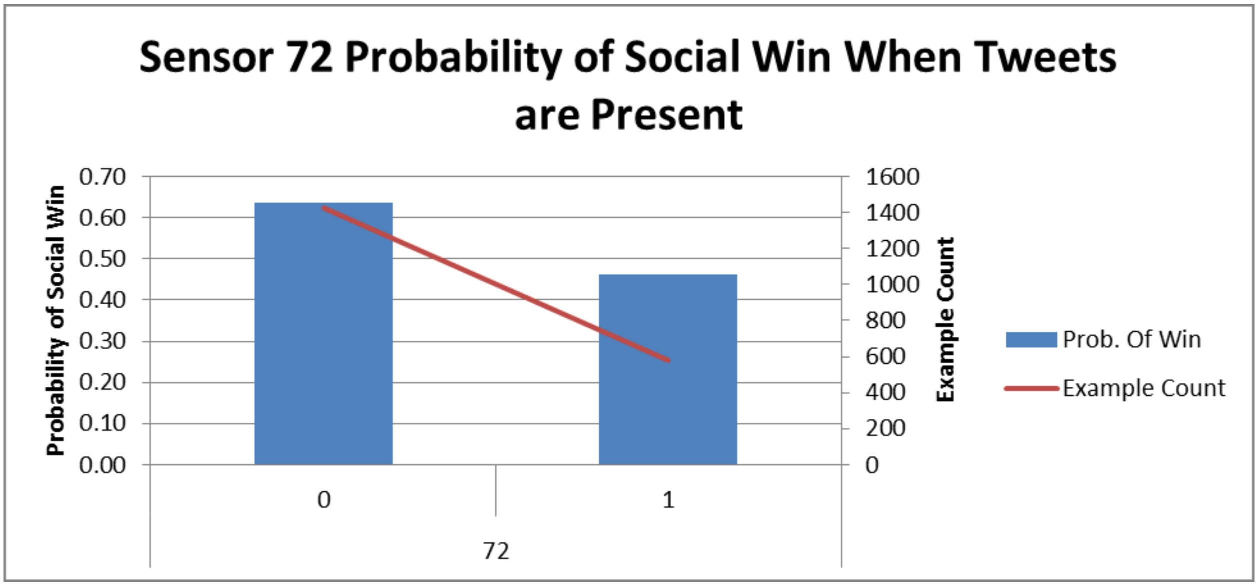
Sensor 72



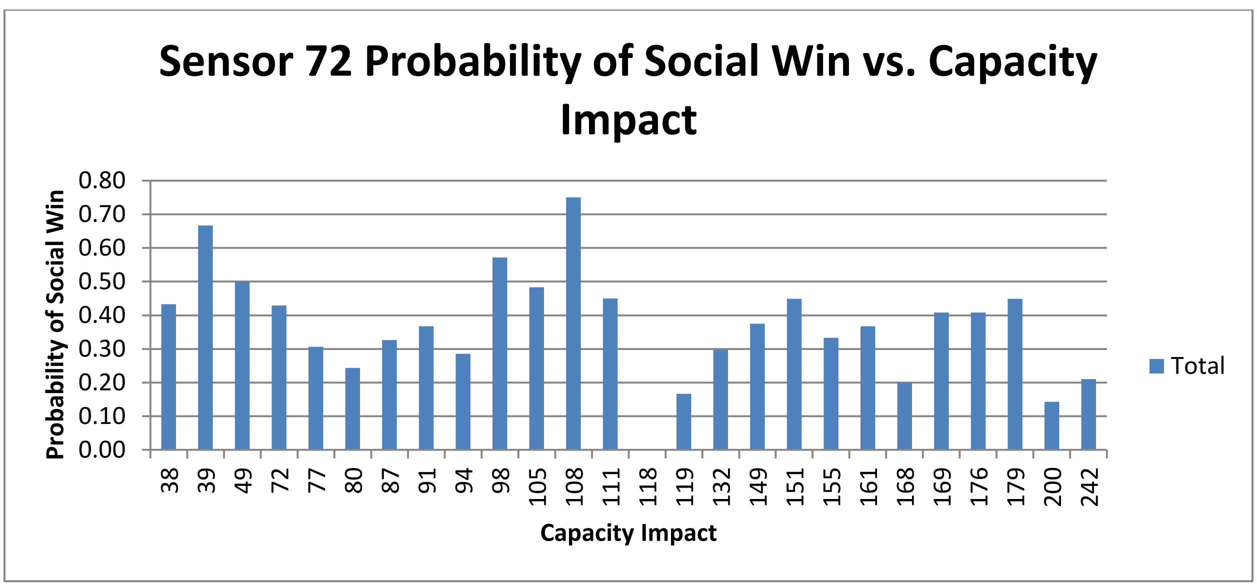
Row Labels	Prob. Of Win	Example Count
1	0.587	288
2	0.524	288
3	0.590	288
4	0.481	283
5	0.584	281
6	0.642	288
7	0.691	288



Row Labels	Prob. Of Win	Example Count
0	0.658	79
1	0.690	84
2	0.536	84
3	0.524	84
4	0.560	84
5	0.762	84
6	0.524	84
7	0.369	84
8	0.583	84
9	0.429	84
10	0.405	84
11	0.417	84
12	0.655	84
13	0.548	84
14	0.571	84
15	0.607	84
16	0.548	84
17	0.631	84
18	0.726	84
19	0.655	84
20	0.655	84
21	0.667	78
22	0.663	83
23	0.690	84

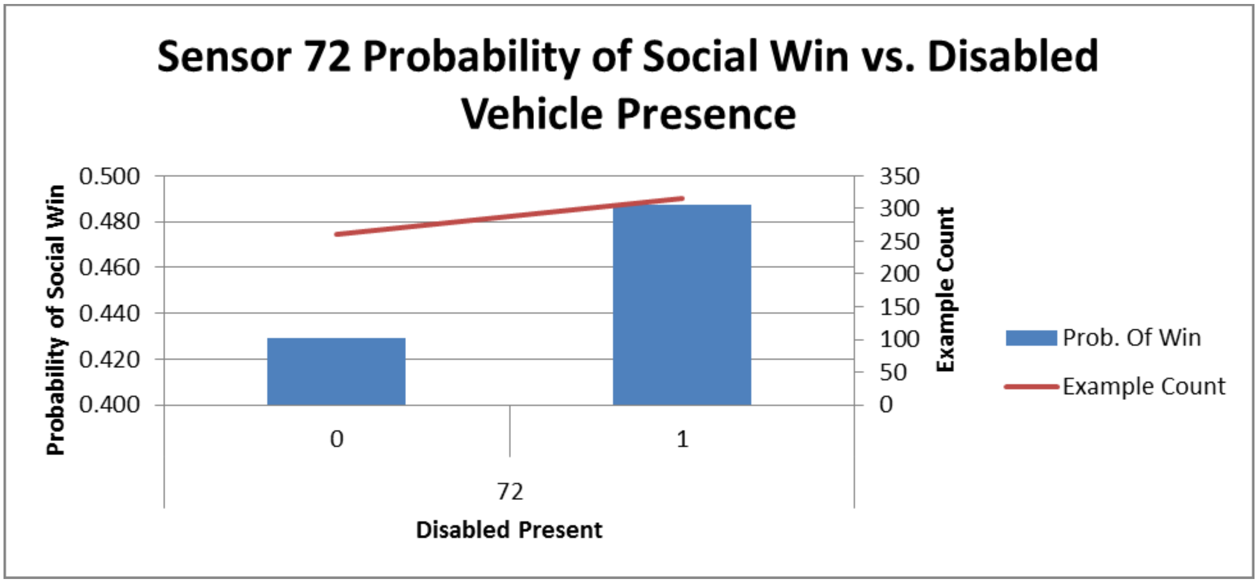


Row Labels	Prob. Of Win	Example Count
0	0.64	1427
1	0.46	577

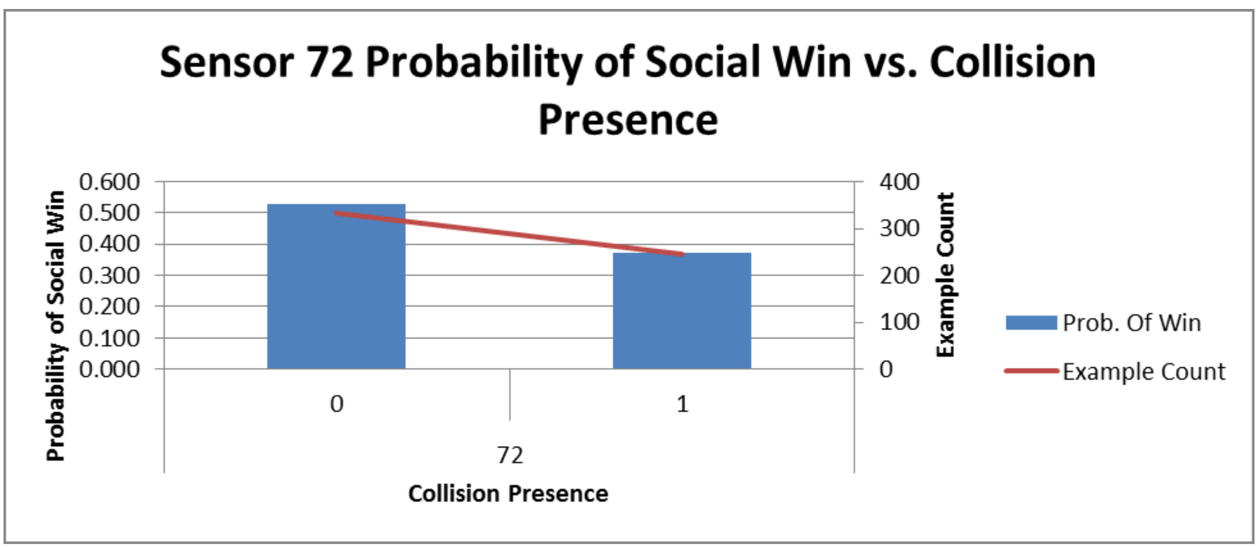


Row Labels	Prob. Of Win	Example Count

0	0.536	69
1	0.453	459
2	0.429	49

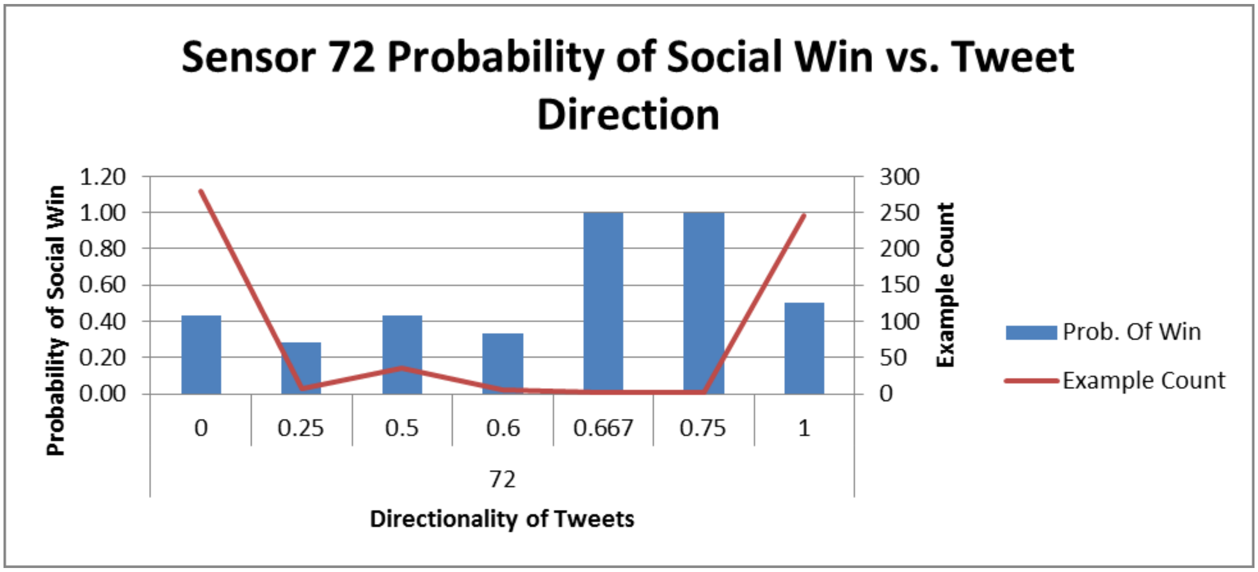


Row Labels	Prob. Of Win	Example Count
0	0.429	261
1	0.487	316

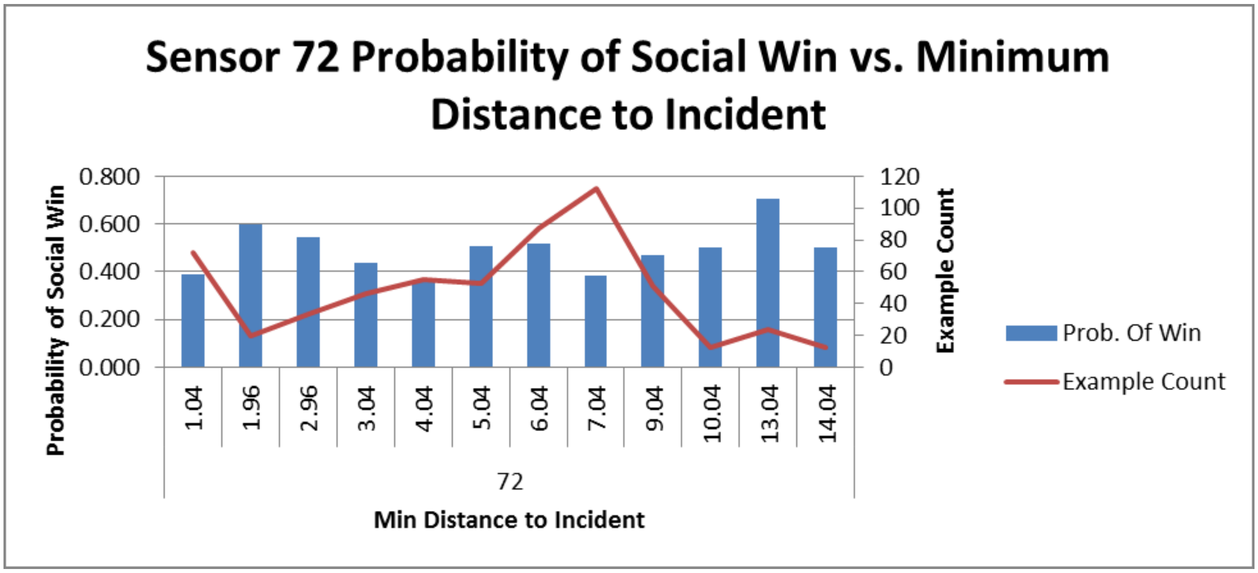


Row Labels	Prob. Of Win	Example Count

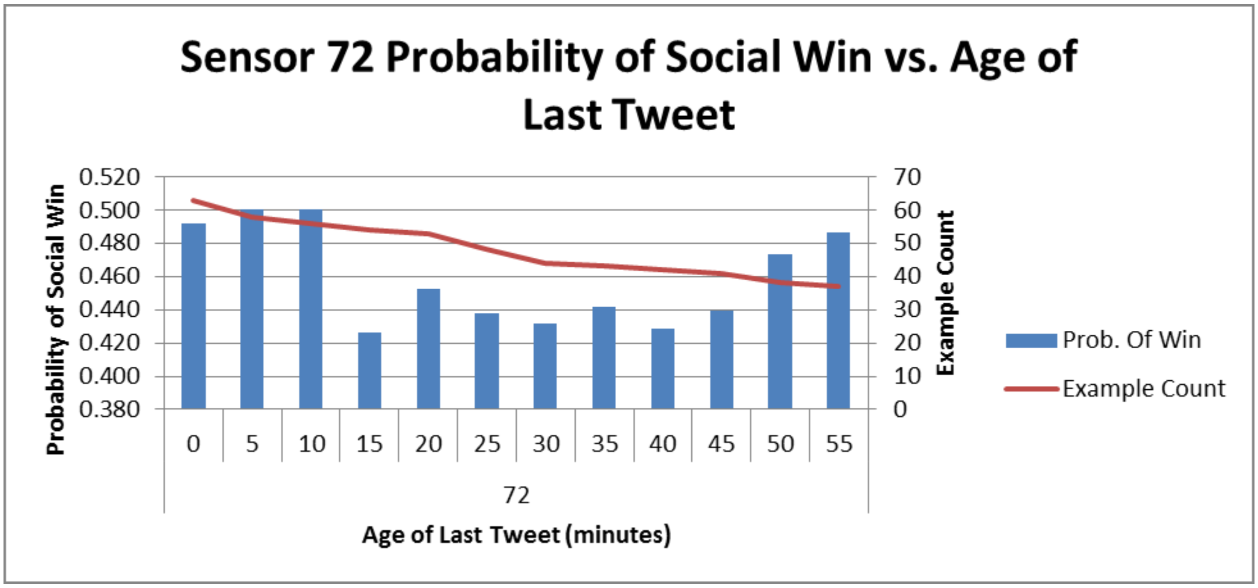
0	0.527	332
1	0.371	245



Row Labels	Prob. Of Win	Example Count
0	0.43	280
0.25	0.29	7
0.5	0.43	35
0.6	0.33	6
0.667	1.00	2
0.75	1.00	1
1	0.50	246

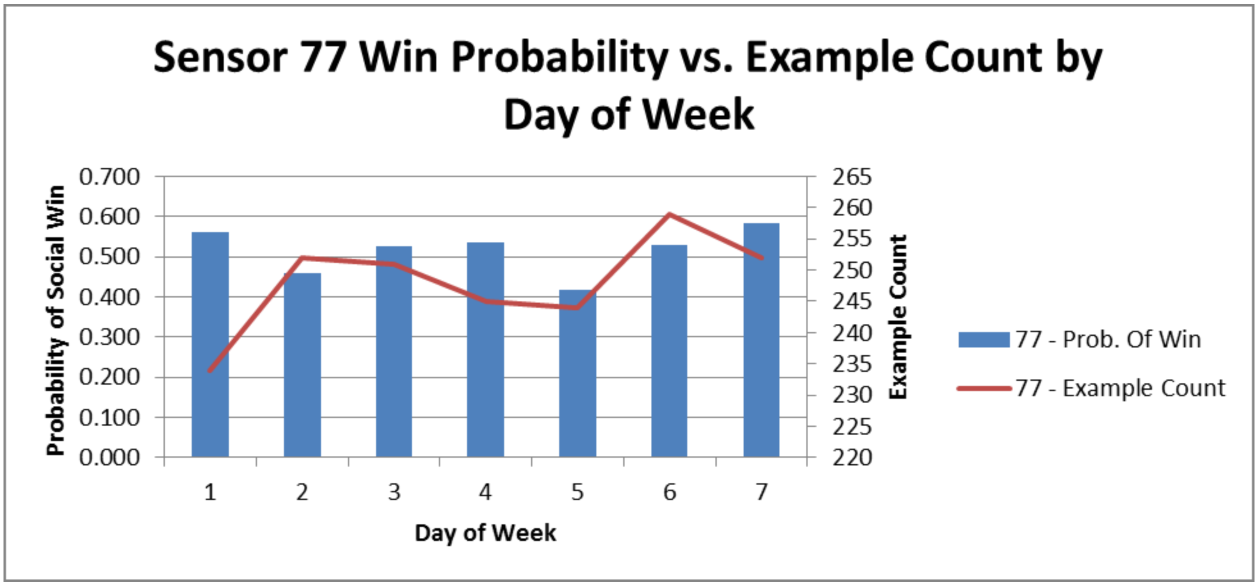


Row Labels	Prob. Of Win	Example Count
1.04	0.389	72
1.96	0.600	20
2.96	0.545	33
3.04	0.435	46
4.04	0.364	55
5.04	0.509	53
6.04	0.517	87
7.04	0.384	112
9.04	0.471	51
10.04	0.500	12
13.04	0.708	24
14.04	0.500	12

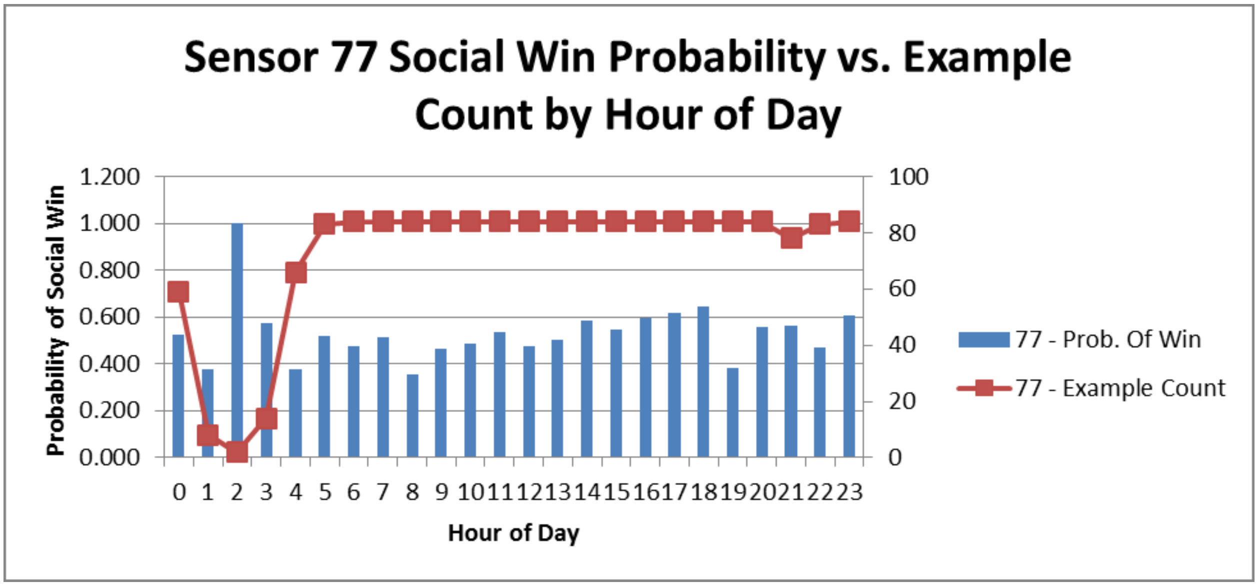


Row Labels	Prob. Of Win	Example Count
0	0.492	63
5	0.500	58
10	0.500	56
15	0.426	54
20	0.453	53
25	0.438	48
30	0.432	44
35	0.442	43
40	0.429	42
45	0.439	41
50	0.474	38
55	0.486	37

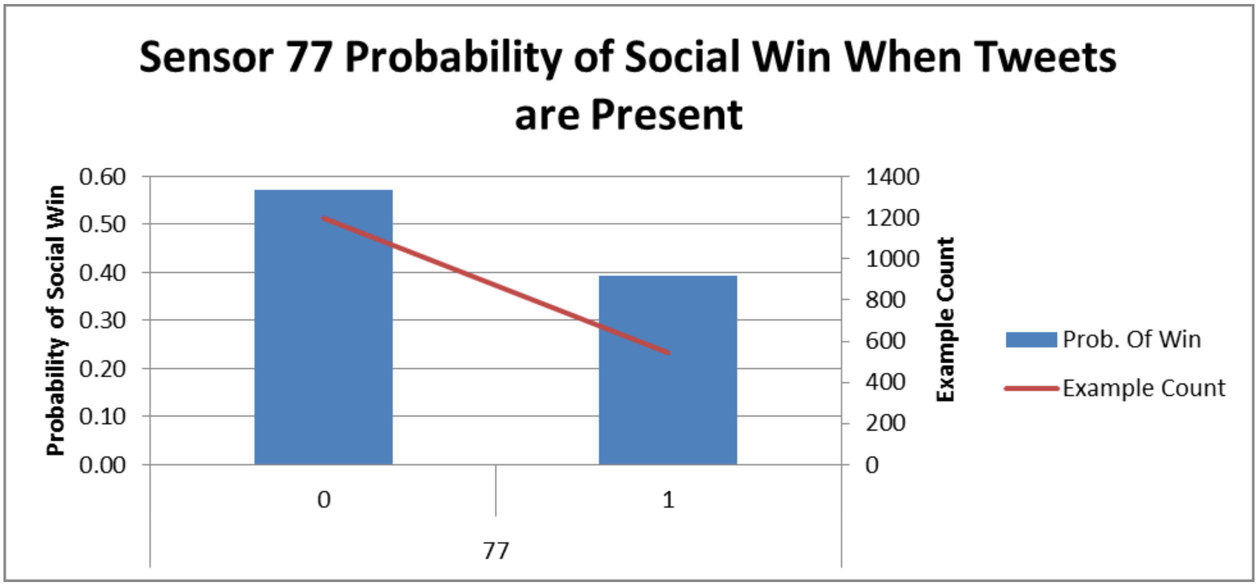
Sensor 77



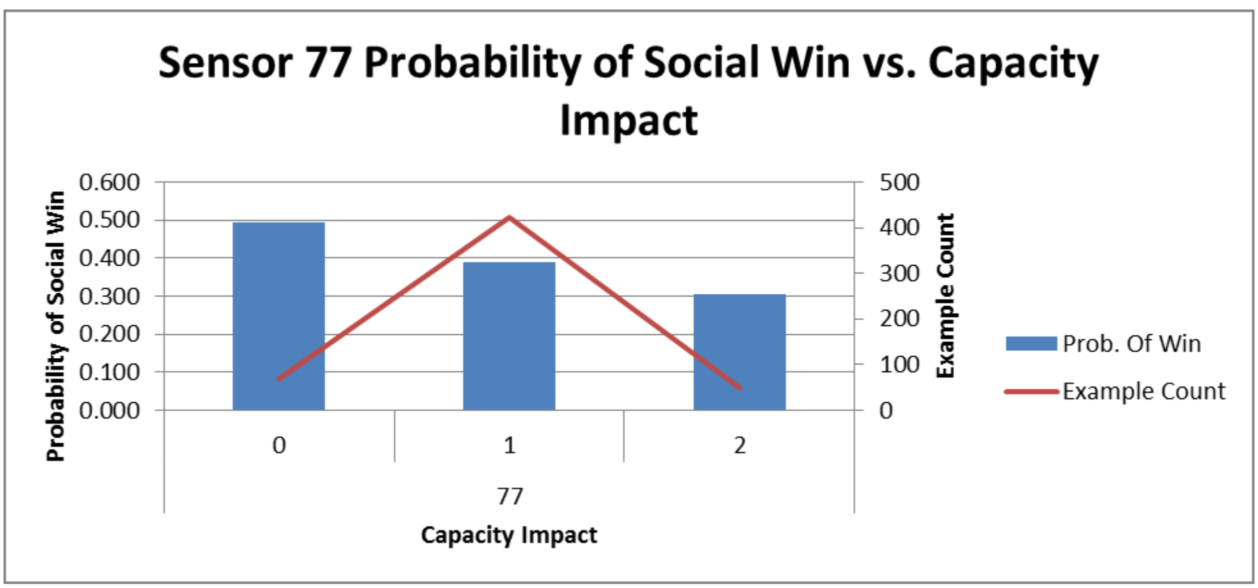
Row Labels	Prob. Of Win	Example Count
1	0.560	234
2	0.460	252
3	0.526	251
4	0.535	245
5	0.418	244
6	0.529	259
7	0.583	252



Row Labels	Prob. Of Win	Example Count
0	0.525	59
1	0.375	8
2	1.000	2
3	0.571	14
4	0.379	66
5	0.518	83
6	0.476	84
7	0.512	84
8	0.357	84
9	0.464	84
10	0.488	84
11	0.536	84
12	0.476	84
13	0.500	84
14	0.583	84
15	0.548	84
16	0.595	84
17	0.619	84
18	0.643	84
19	0.381	84
20	0.560	84
21	0.564	78
22	0.470	83
23	0.607	84

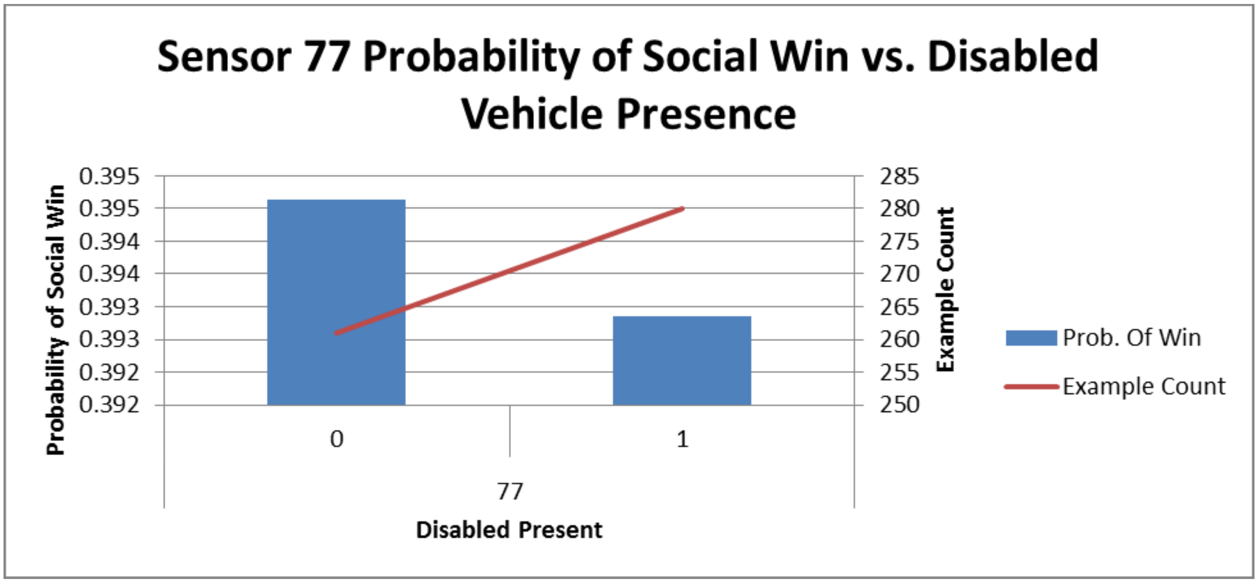


Row Labels	Prob. Of Win	Example Count
0	0.57	1196
1	0.39	541

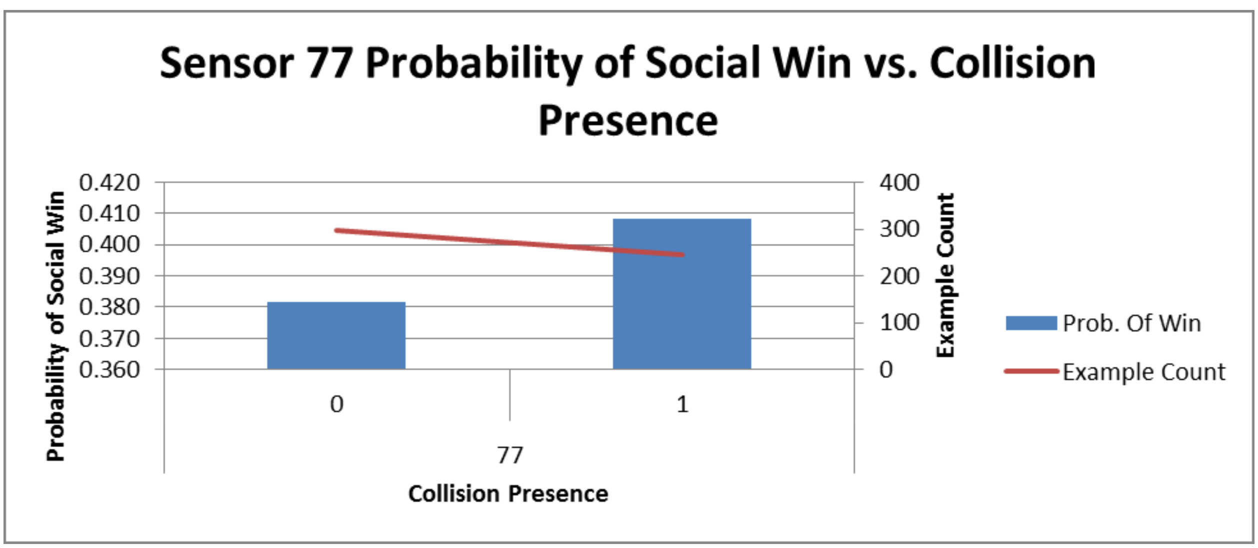


Row Labels	Prob. Of Win	Example Count

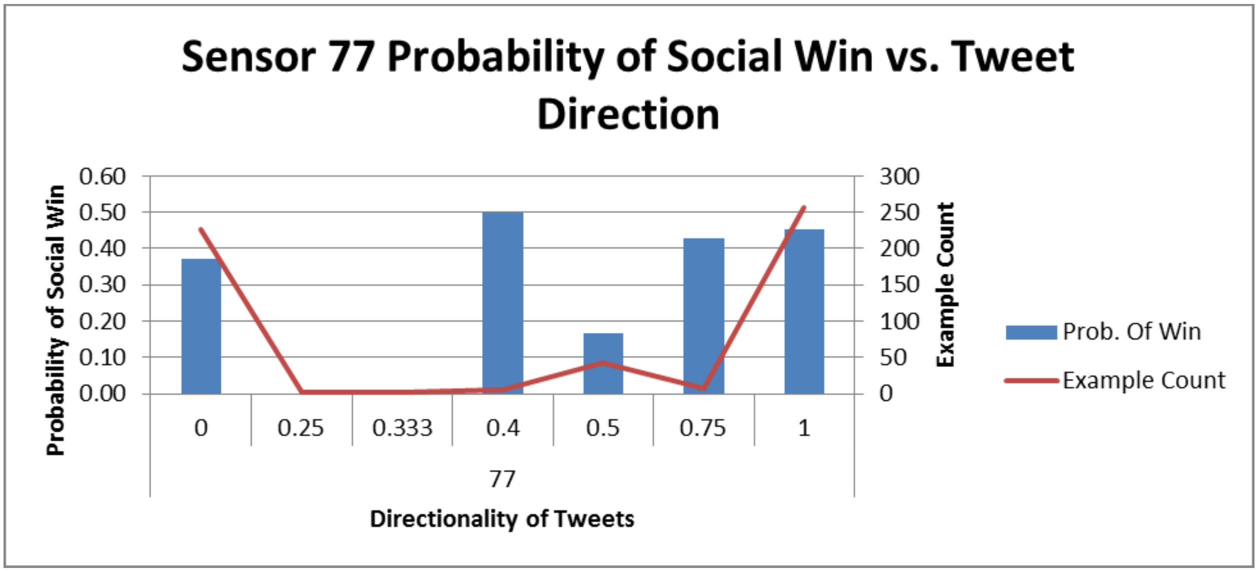
0	0.493	69
1	0.388	423
2	0.306	49



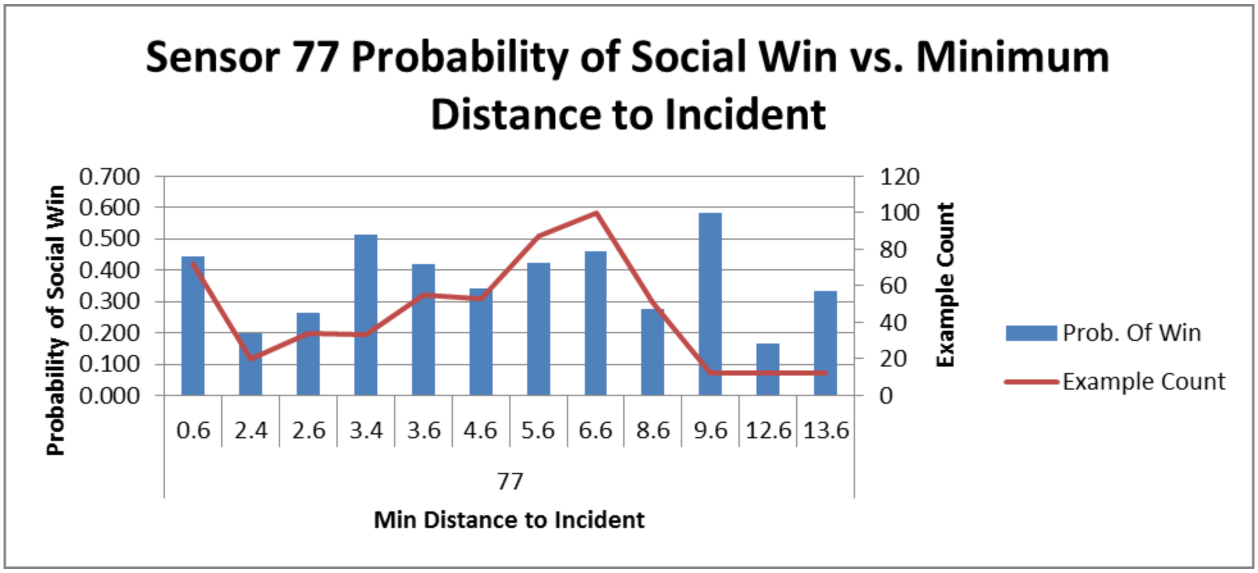
Row Labels	Prob. Of Win	Example Count
0	0.395	261
1	0.393	280



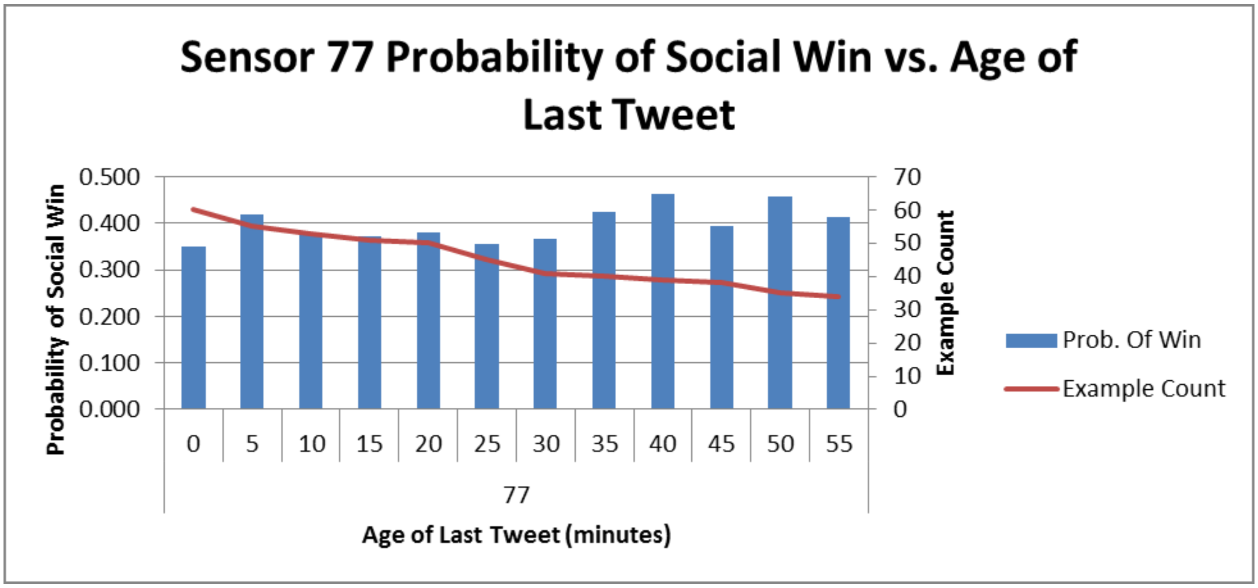
Row Labels	Prob. Of Win	Example Count
0	0.382	296
1	0.408	245



Row Labels	Prob. Of Win	Example Count
0	0.37	227
0.25	0.00	1
0.333	0.00	2
0.4	0.50	6
0.5	0.17	42
0.75	0.43	7
1	0.45	256

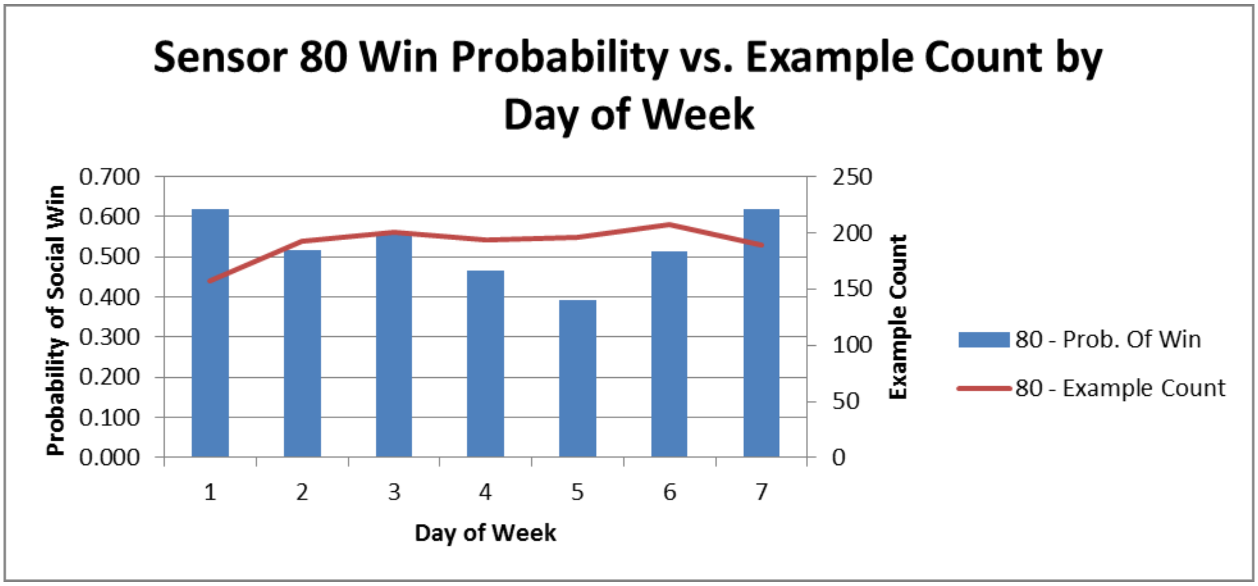


Row Labels	Prob. Of Win	Example Count
0.6	0.444	72
2.4	0.200	20
2.6	0.265	34
3.4	0.515	33
3.6	0.418	55
4.6	0.340	53
5.6	0.425	87
6.6	0.460	100
8.6	0.275	51
9.6	0.583	12
12.6	0.167	12
13.6	0.333	12

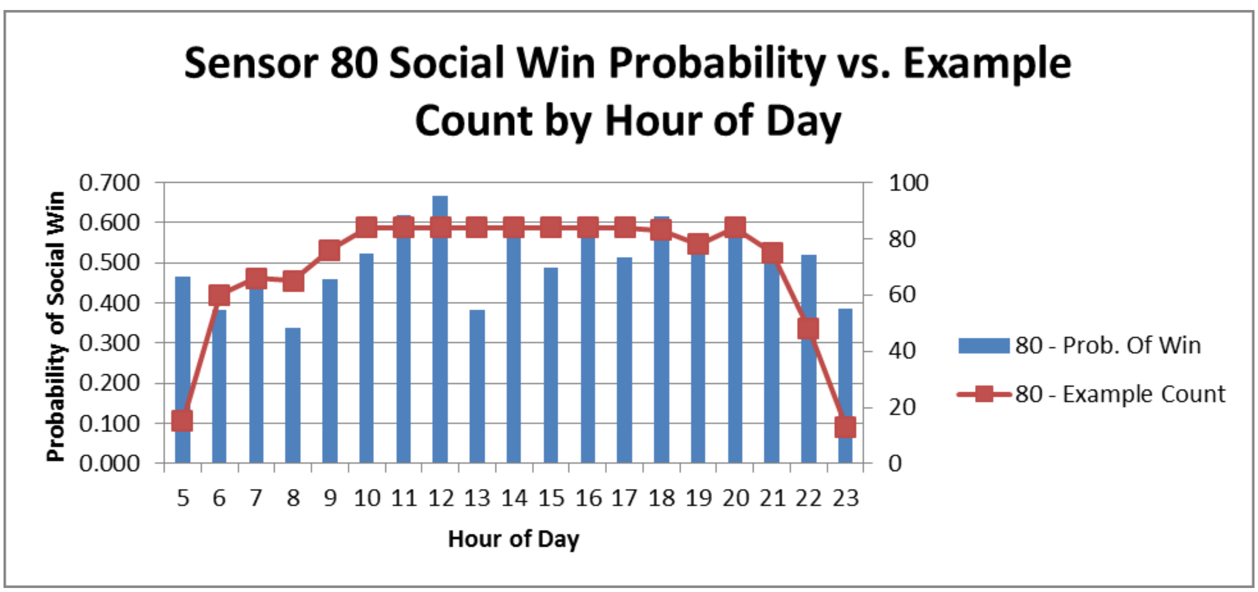


Row Labels	Prob. Of Win	Example Count
0	0.350	60
5	0.418	55
10	0.377	53
15	0.373	51
20	0.380	50
25	0.356	45
30	0.366	41
35	0.425	40
40	0.462	39
45	0.395	38
50	0.457	35
55	0.412	34

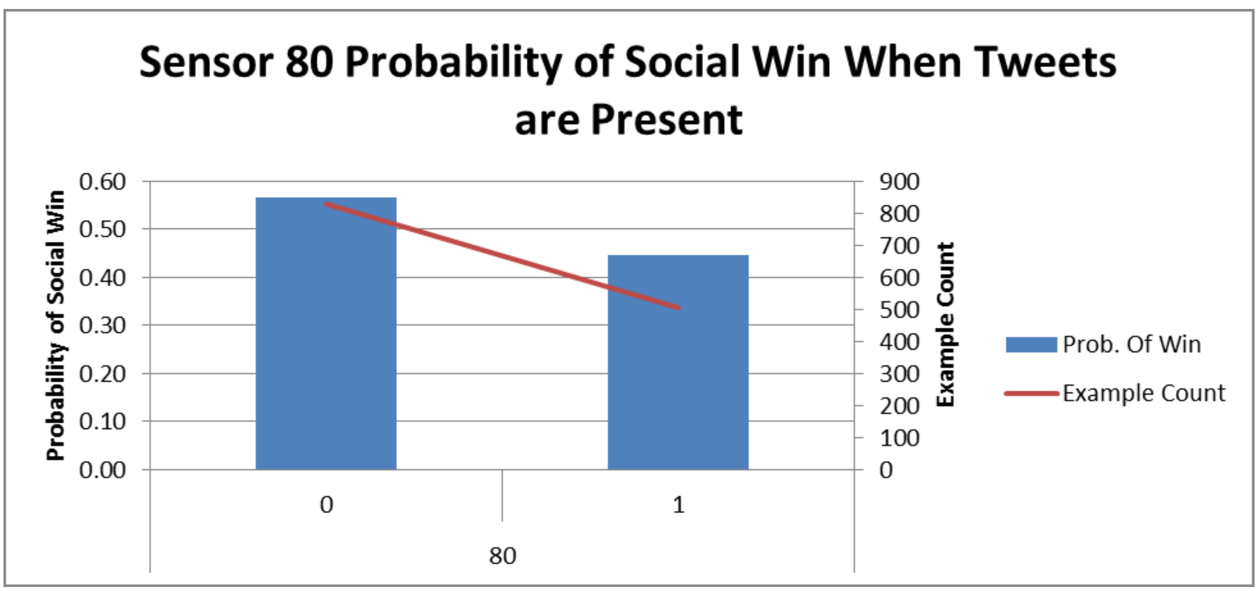
Sensor 80



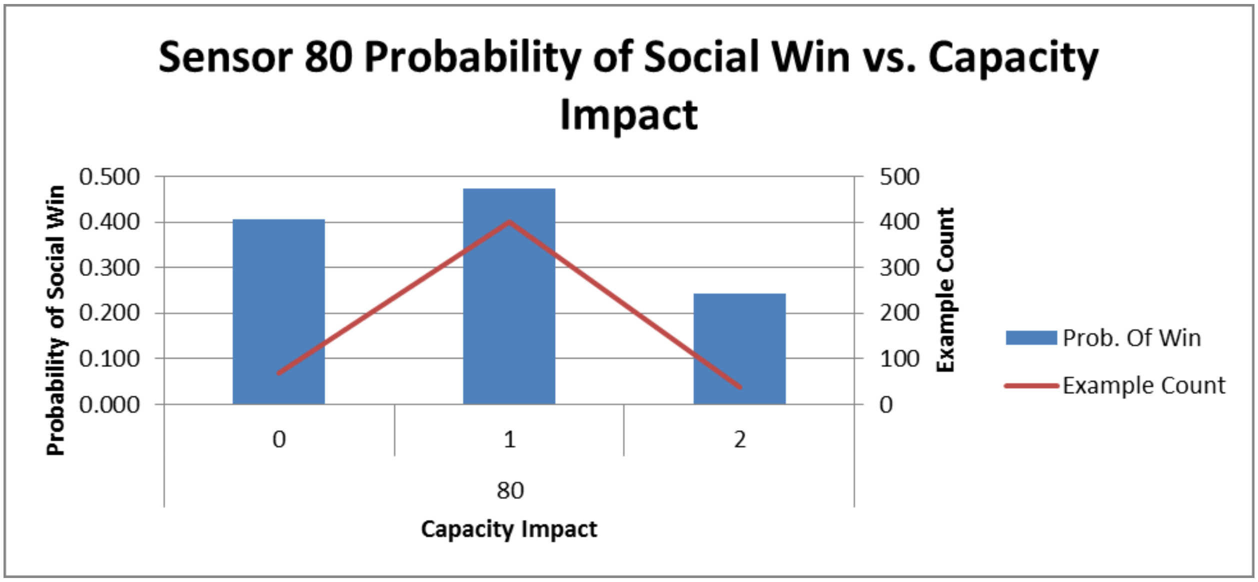
Row Labels	Prob. Of Win	Example Count
1	0.618	157
2	0.516	192
3	0.555	200
4	0.464	194
5	0.393	196
6	0.512	207
7	0.619	189



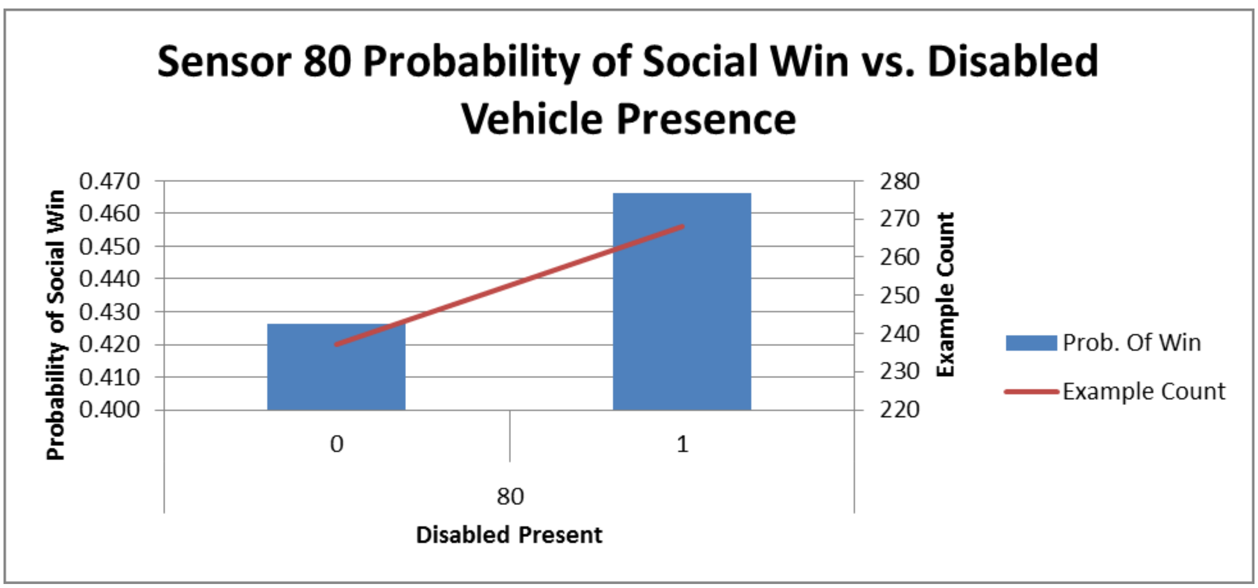
Row Labels	Prob. Of Win	Example Count
5	0.467	15
6	0.383	60
7	0.470	66
8	0.338	65
9	0.461	76
10	0.524	84
11	0.619	84
12	0.667	84
13	0.381	84
14	0.583	84
15	0.488	84
16	0.595	84
17	0.512	84
18	0.614	83
19	0.551	78
20	0.560	84
21	0.547	75
22	0.521	48
23	0.385	13



Row Labels	Prob. Of Win	Example Count
0	0.57	830
1	0.45	505

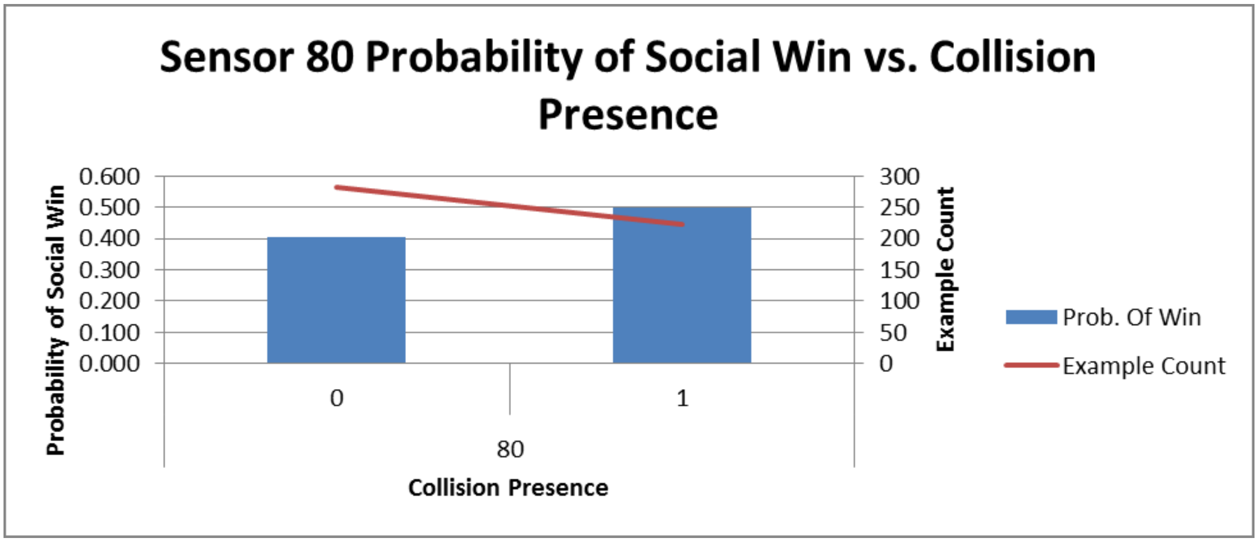


Row Labels	Prob. Of Win	Example Count
0	0.406	69
1	0.474	399
2	0.243	37

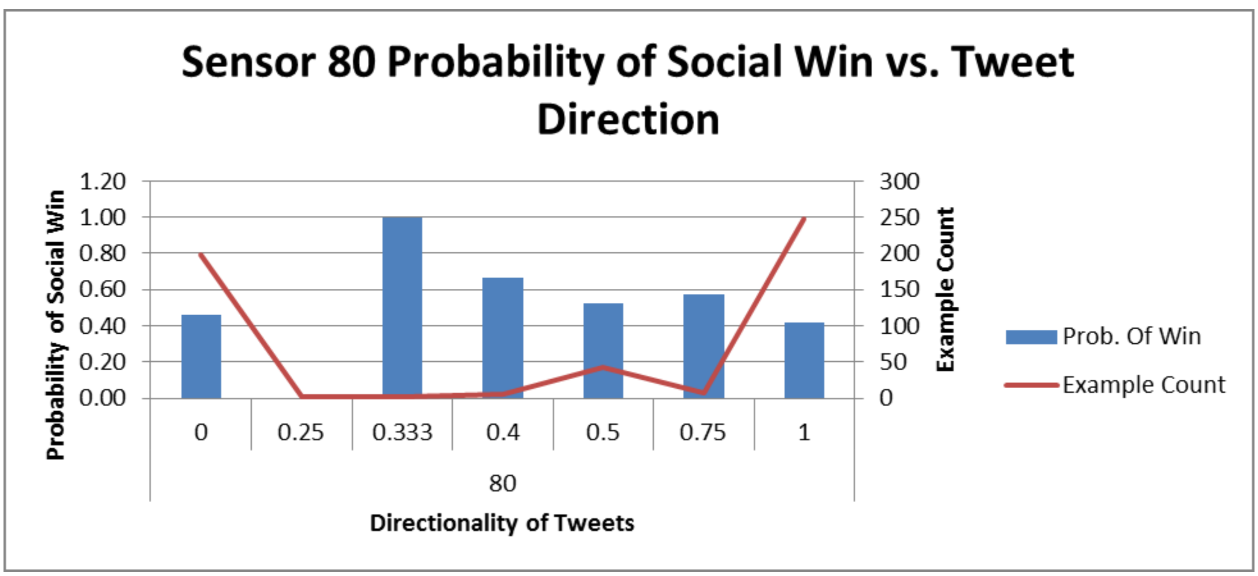


Row Labels	Prob. Of Win	Example Count

0	0.426	237
1	0.466	268

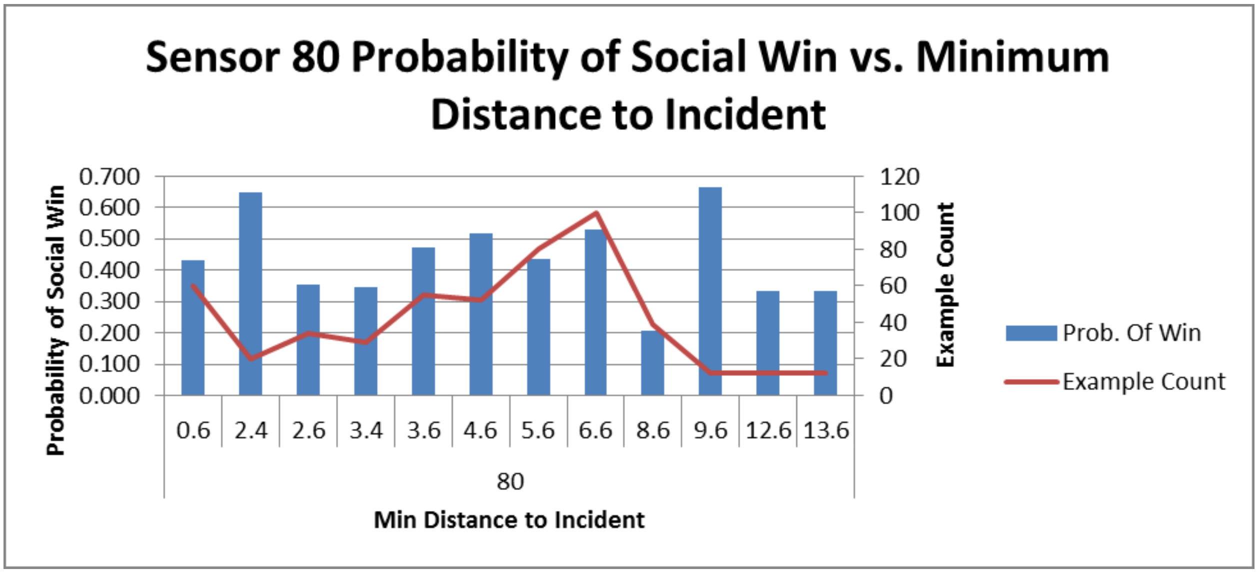


Row Labels	Prob. Of Win	Example Count
0	0.406	283
1	0.500	222

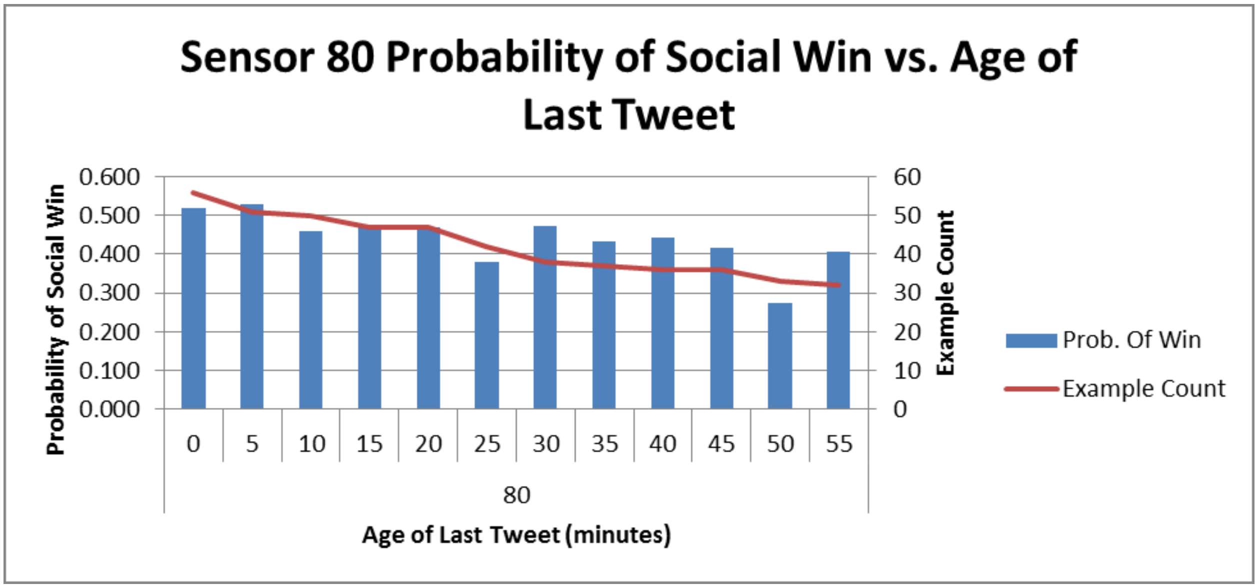


Row Labels	Prob. Of Win	Example Count

0	0.46	199
0.25	0.00	1
0.333	1.00	2
0.4	0.67	6
0.5	0.52	42
0.75	0.57	7
1	0.42	248

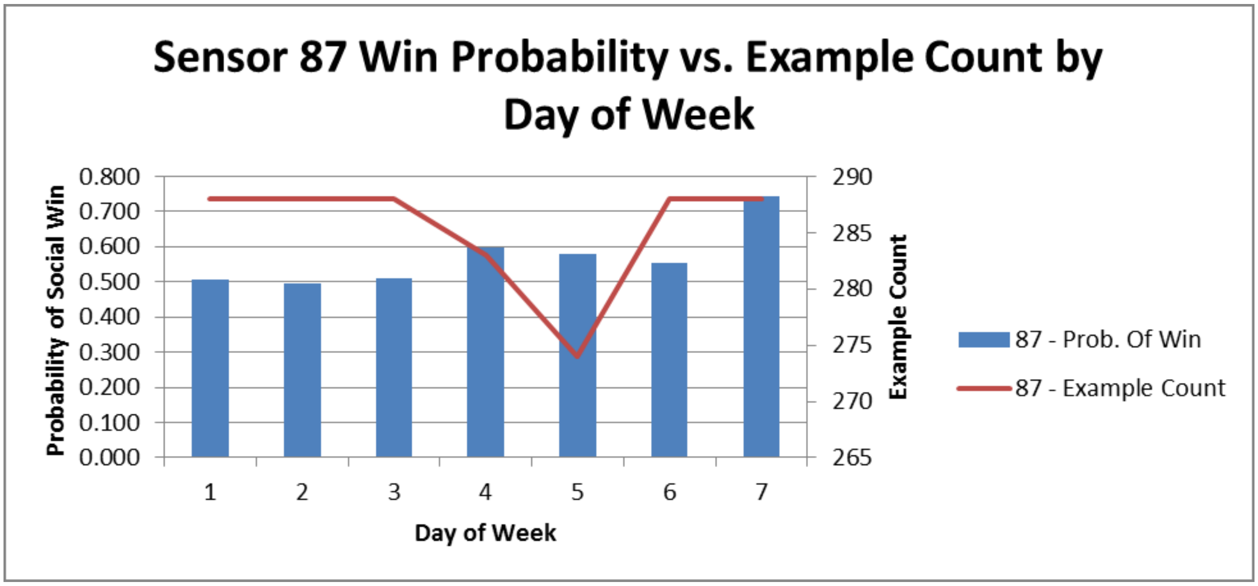


Row Labels	Prob. Of Win	Example Count
0.6	0.433	60
2.4	0.650	20
2.6	0.353	34
3.4	0.345	29
3.6	0.473	55
4.6	0.519	52
5.6	0.438	80
6.6	0.530	100
8.6	0.205	39
9.6	0.667	12
12.6	0.333	12
13.6	0.333	12

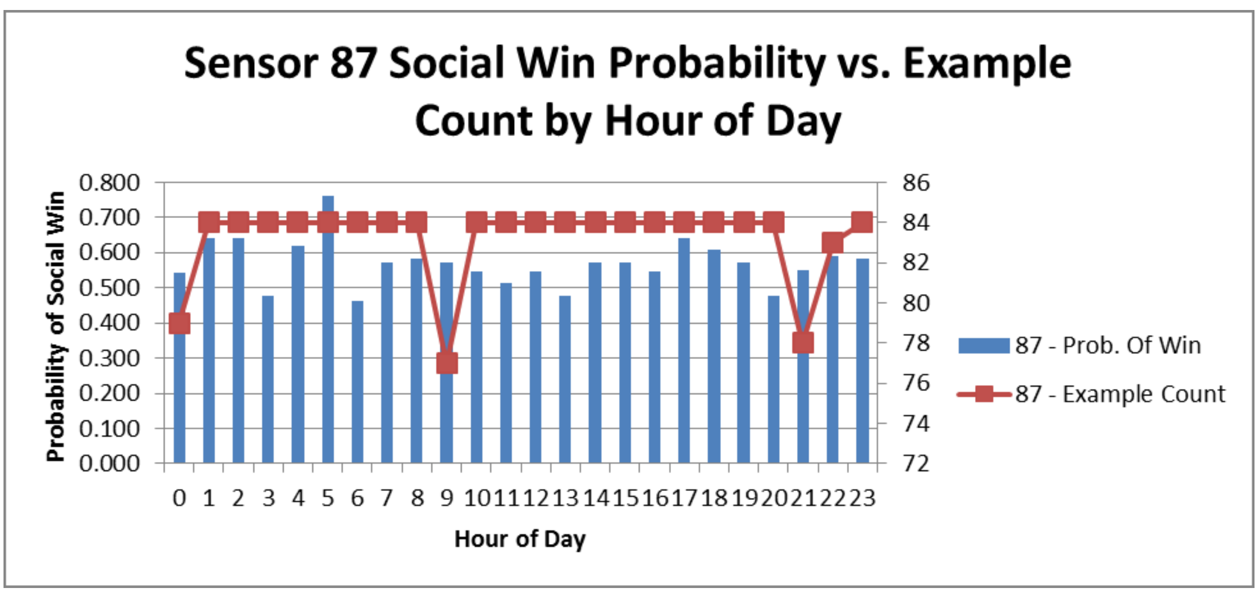


Row Labels	Prob. Of Win	Example Count
0	0.518	56
5	0.529	51
10	0.460	50
15	0.468	47
20	0.468	47
25	0.381	42
30	0.474	38
35	0.432	37
40	0.444	36
45	0.417	36
50	0.273	33
55	0.406	32

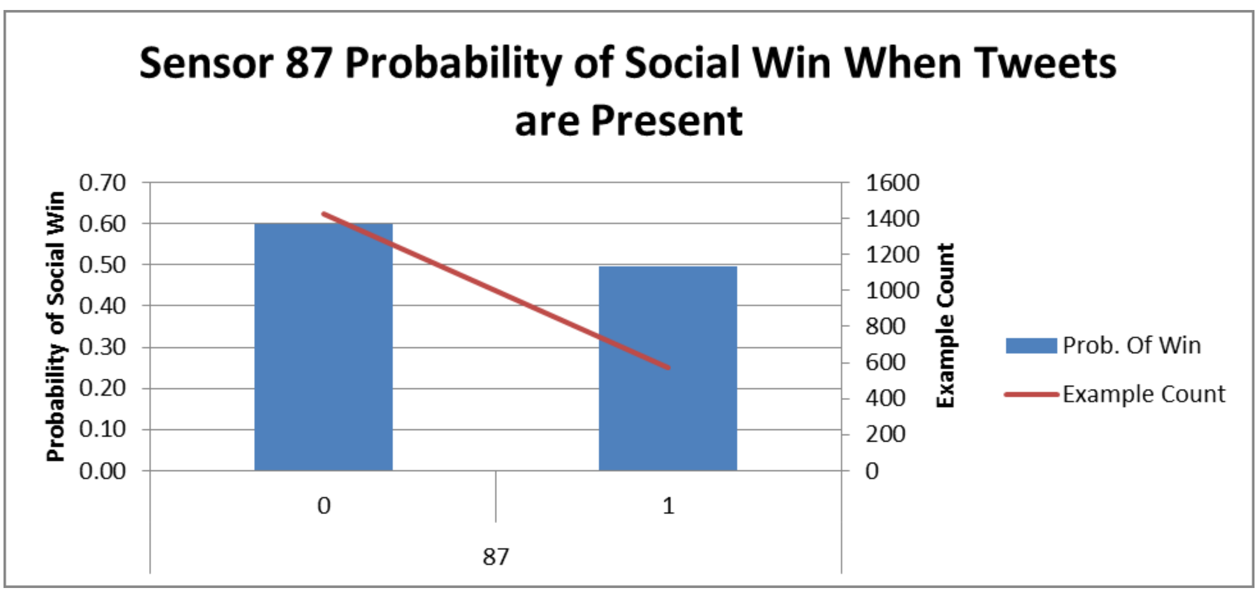
Sensor 87



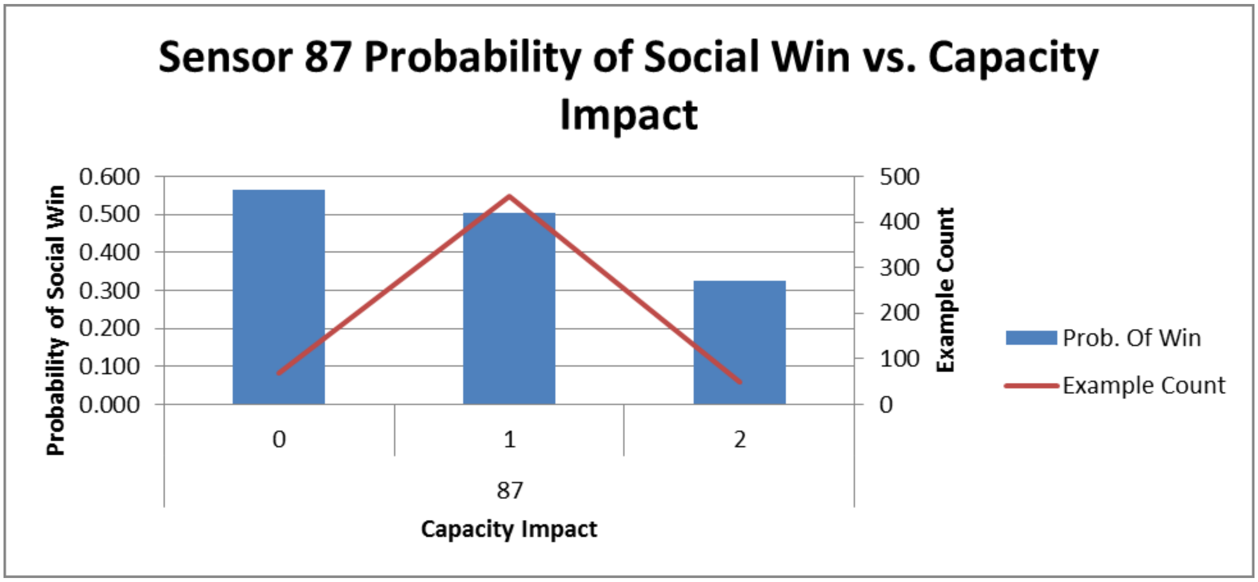
Row Labels	Prob. Of Win	Example Count
1	0.507	288
2	0.497	288
3	0.510	288
4	0.597	283
5	0.580	274
6	0.556	288
7	0.743	288



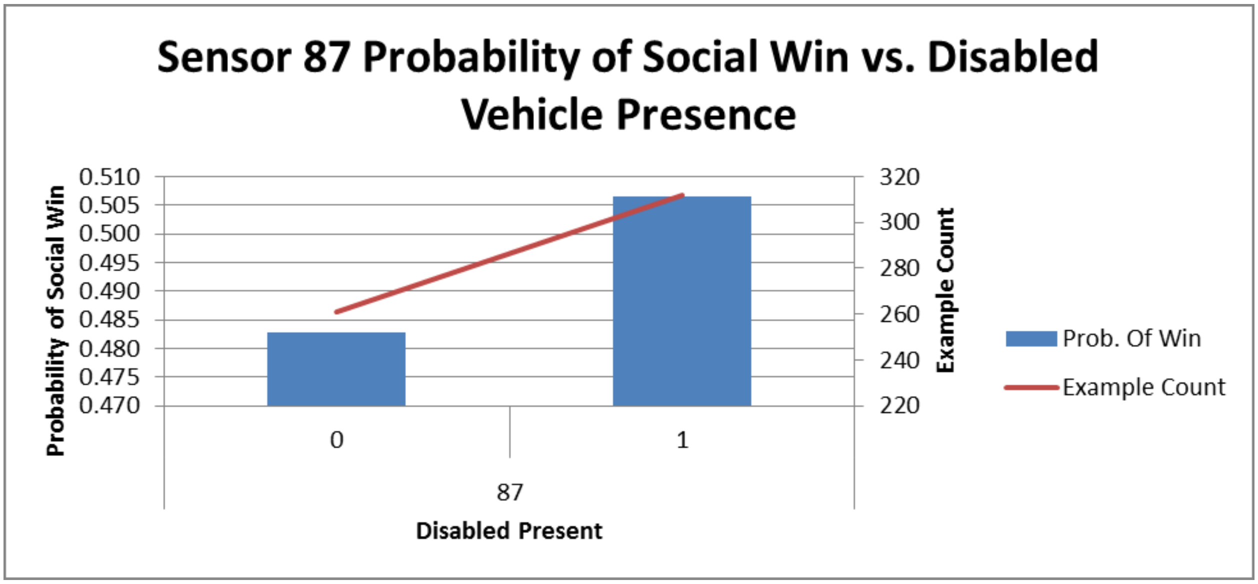
Row Labels	Prob. Of Win	Example Count
0	0.544	79
1	0.643	84
2	0.643	84
3	0.476	84
4	0.619	84
5	0.762	84
6	0.464	84
7	0.571	84
8	0.583	84
9	0.571	77
10	0.548	84
11	0.512	84
12	0.548	84
13	0.476	84
14	0.571	84
15	0.571	84
16	0.548	84
17	0.643	84
18	0.607	84
19	0.571	84
20	0.476	84
21	0.551	78
22	0.590	83
23	0.583	84



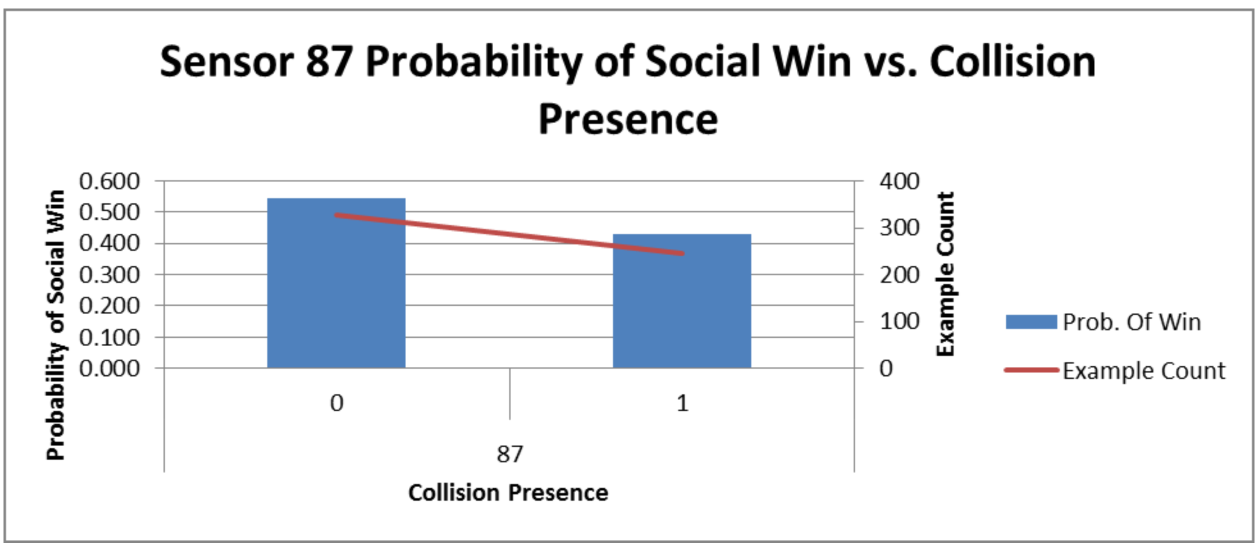
Row Labels	Prob. Of Win	Example Count
0	0.60	1424
1	0.50	573



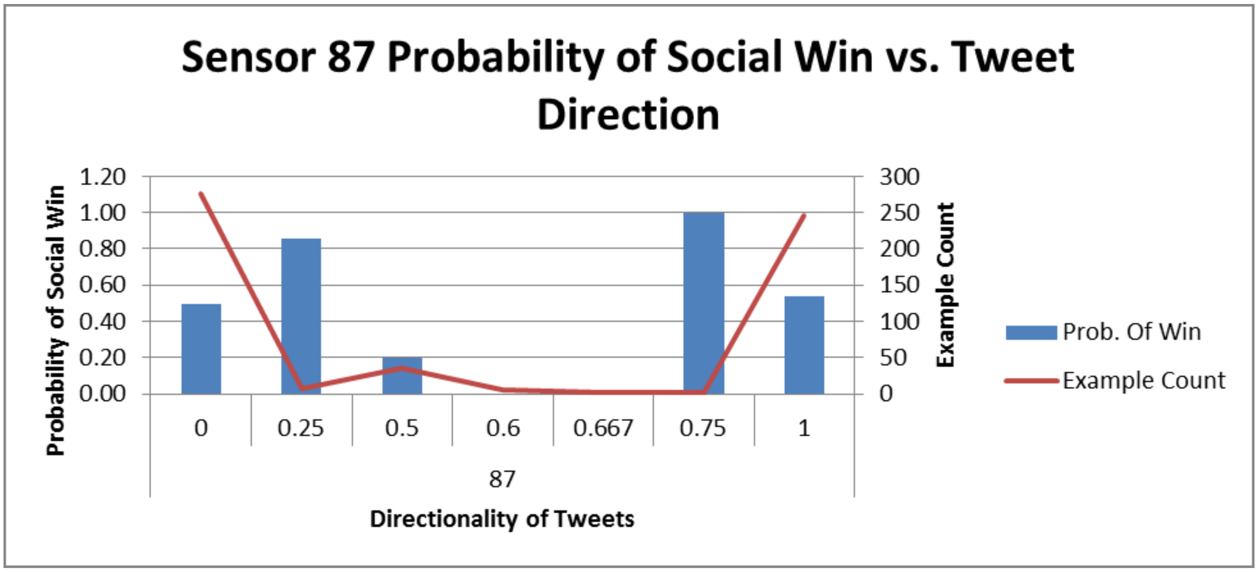
Row Labels	Prob. Of Win	Example Count
0	0.565	69
1	0.503	455
2	0.327	49



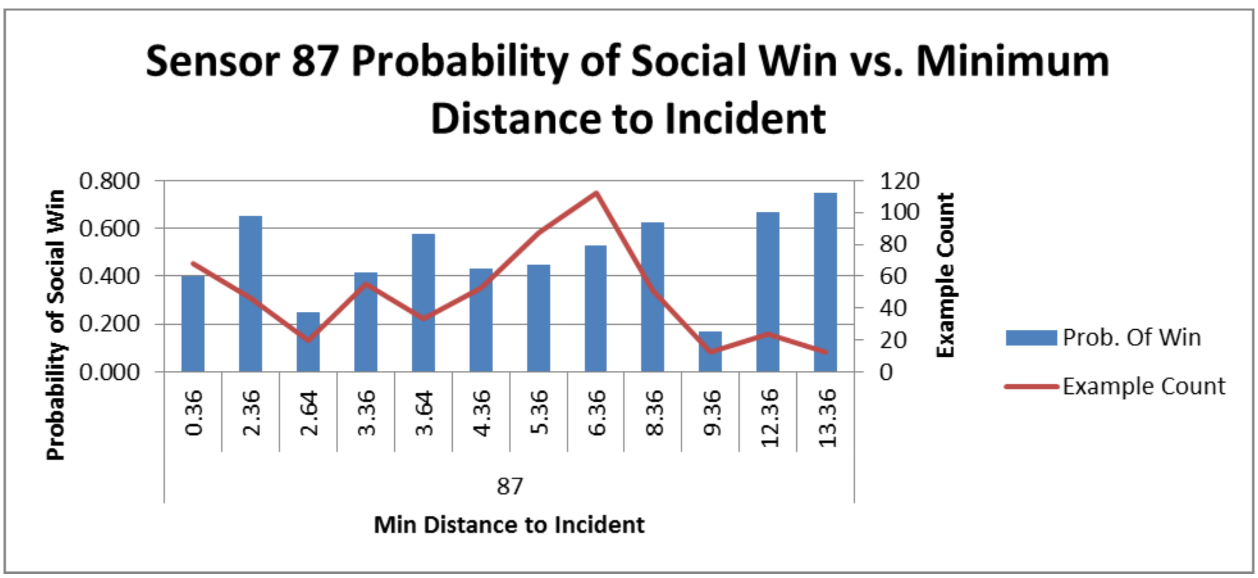
Row Labels	Prob. Of Win	Example Count
0	0.483	261
1	0.506	312



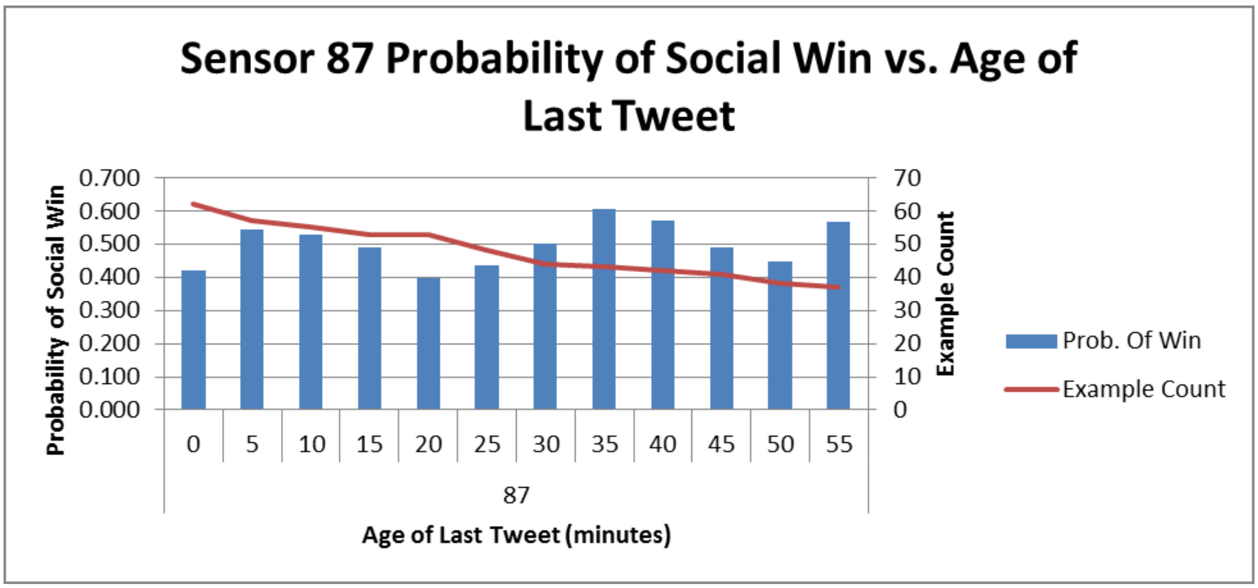
Row Labels	Prob. Of Win	Example Count
0	0.546	328
1	0.429	245



Row Labels	Prob. Of Win	Example Count
0	0.50	276
0.25	0.86	7
0.5	0.20	35
0.6	0.00	6
0.667	0.00	2
0.75	1.00	1
1	0.54	246



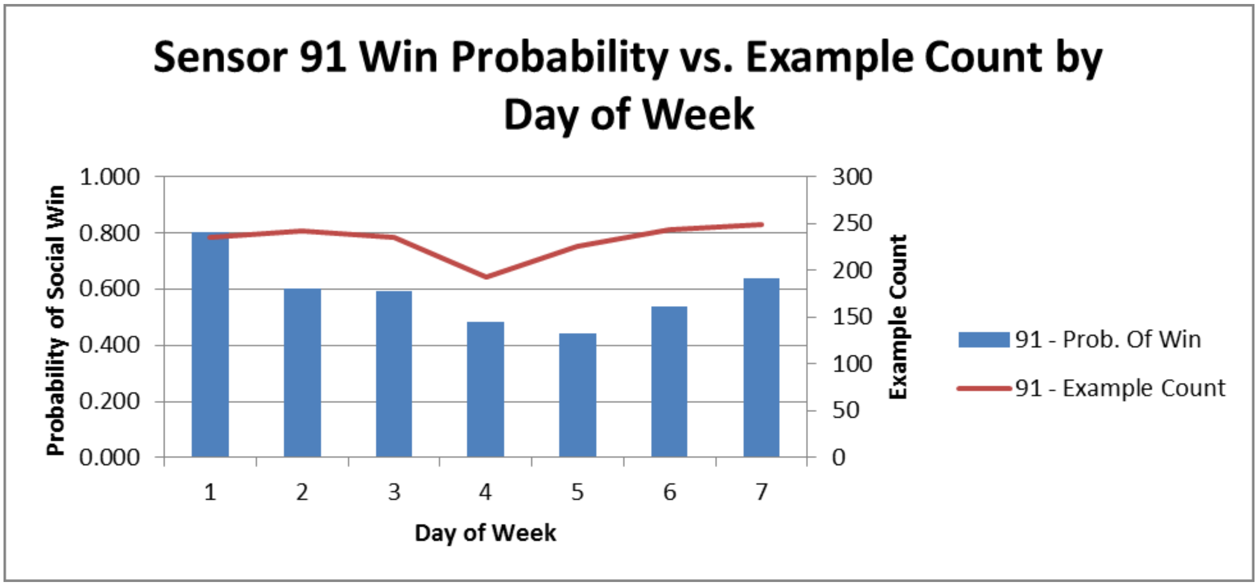
Row Labels	Prob. Of Win	Example Count
0.36	0.397	68
2.36	0.652	46
2.64	0.250	20
3.36	0.418	55
3.64	0.576	33
4.36	0.434	53
5.36	0.448	87
6.36	0.527	112
8.36	0.627	51
9.36	0.167	12
12.36	0.667	24
13.36	0.750	12



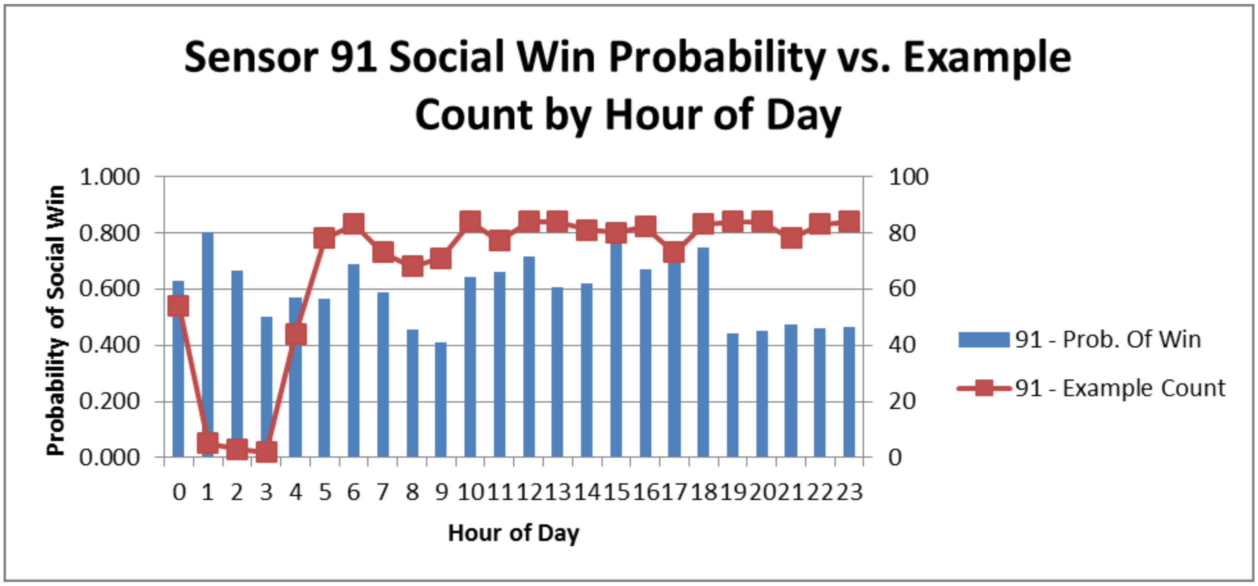
Row Labels	Prob. Of Win	Example Count
0	0.419	62
5	0.544	57
10	0.527	55
15	0.491	53
20	0.396	53
25	0.438	48
30	0.500	44
35	0.605	43
40	0.571	42

45	0.488	41
50	0.447	38
55	0.568	37

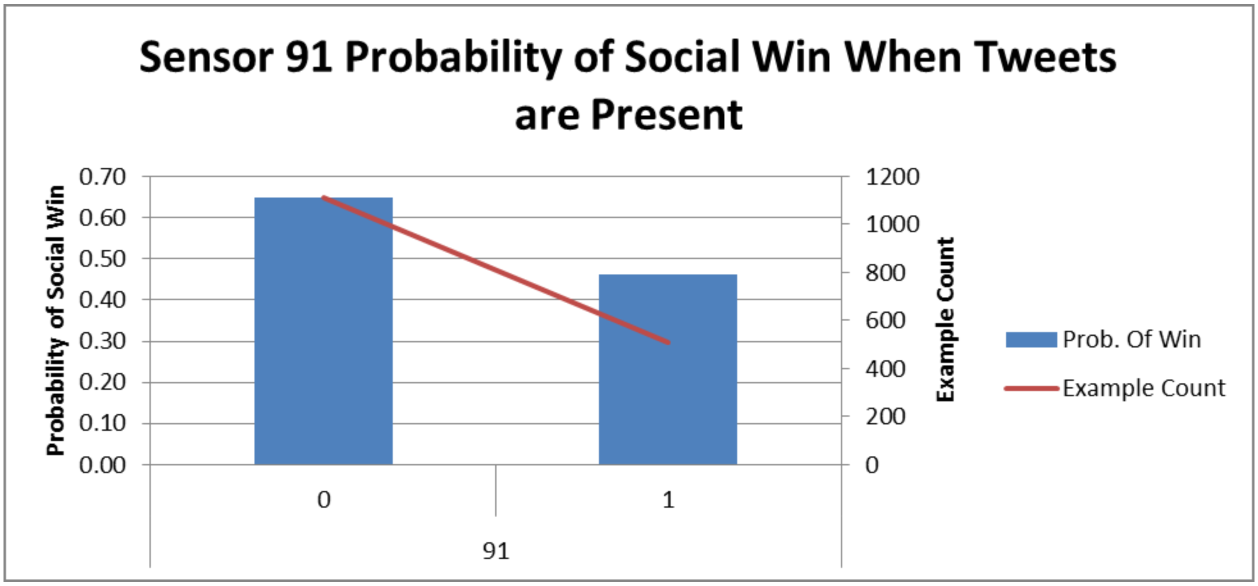
Sensor 91



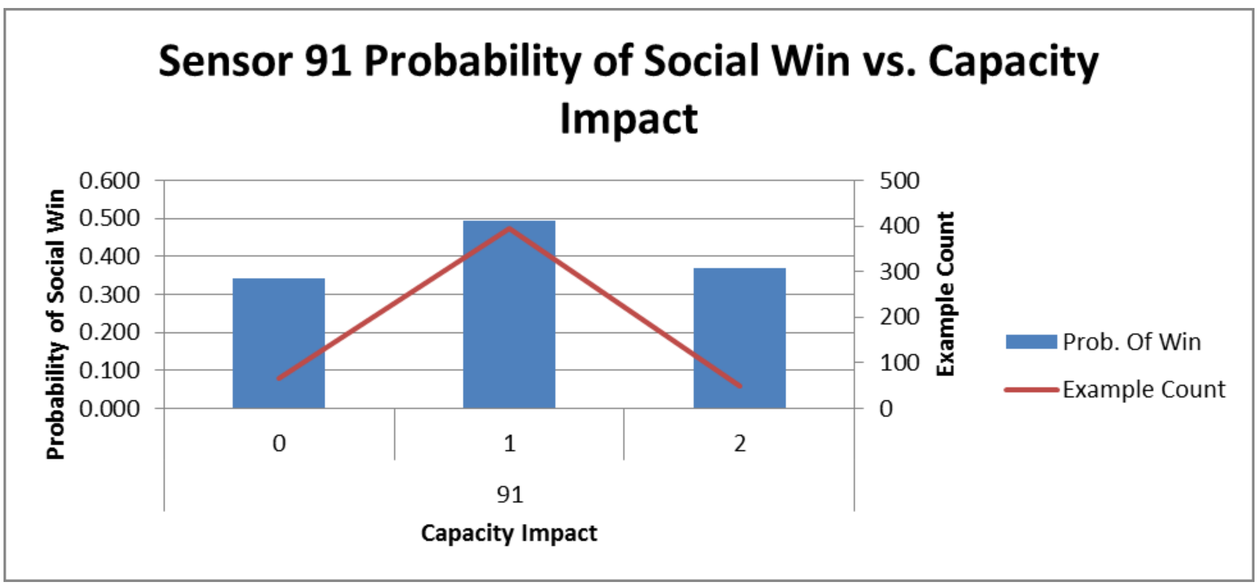
Row Labels	Prob. Of Win	Example Count
1	0.800	235
2	0.603	242
3	0.591	235
4	0.482	193
5	0.440	225
6	0.539	243
7	0.639	249



Row Labels	Prob. Of Win	Example Count
0	0.630	54
1	0.800	5
2	0.667	3
3	0.500	2
4	0.568	44
5	0.564	78
6	0.687	83
7	0.589	73
8	0.456	68
9	0.408	71
10	0.643	84
11	0.662	77
12	0.714	84
13	0.607	84
14	0.617	81
15	0.763	80
16	0.671	82
17	0.712	73
18	0.747	83
19	0.440	84
20	0.452	84
21	0.474	78
22	0.458	83
23	0.464	84

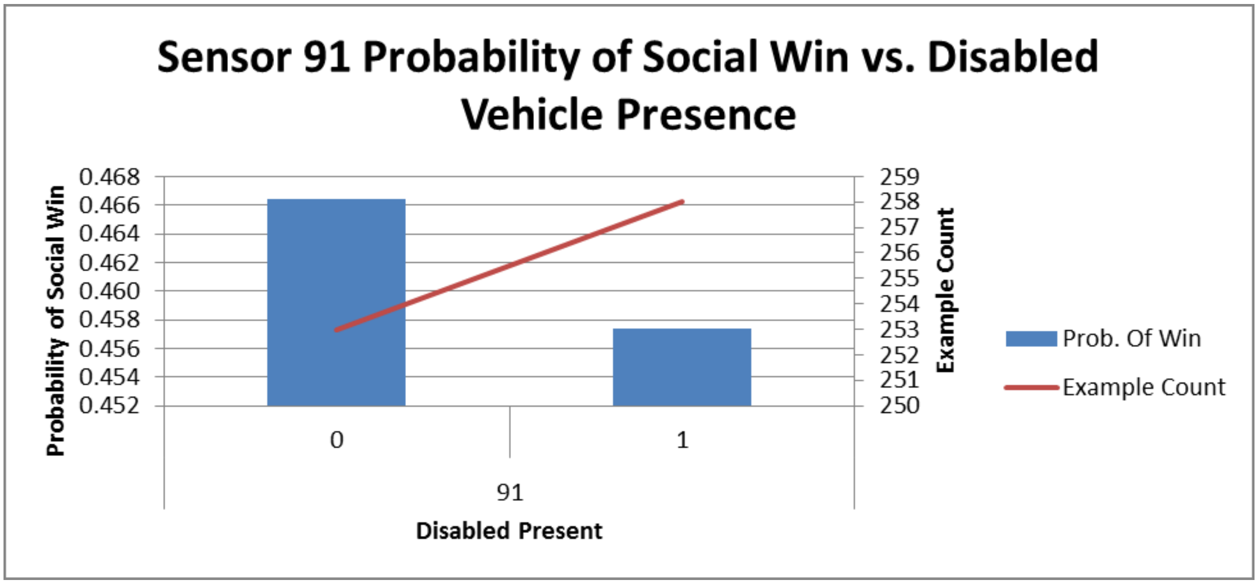


Row Labels	Prob. Of Win	Example Count
0	0.65	1111
1	0.46	511

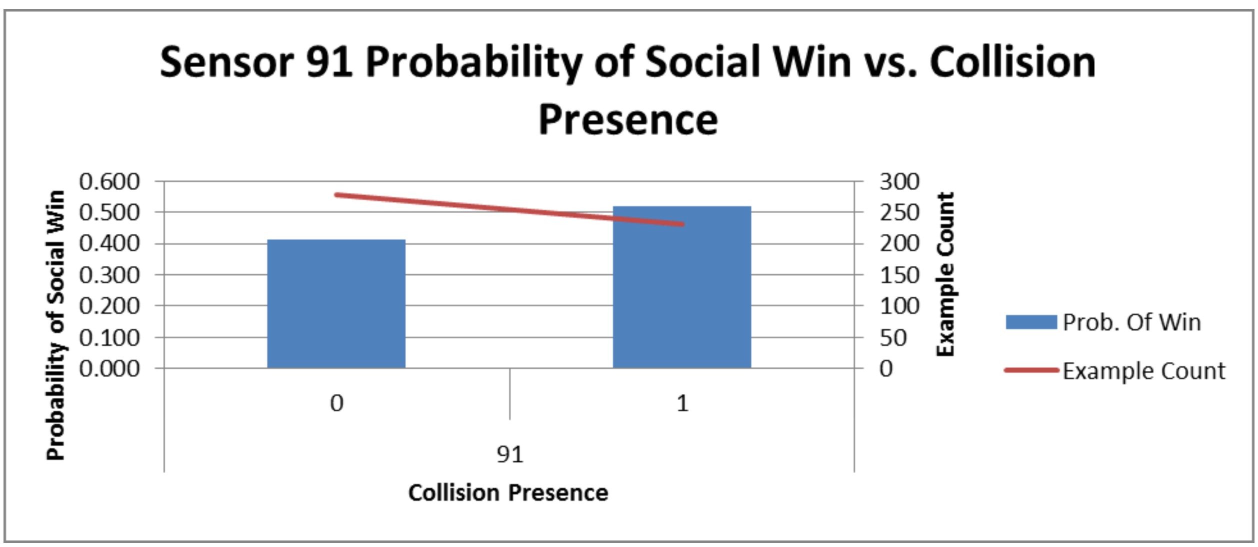


Row Labels	Prob. Of Win	Example Count
0	0.343	67
1	0.494	395

2	0.367	49

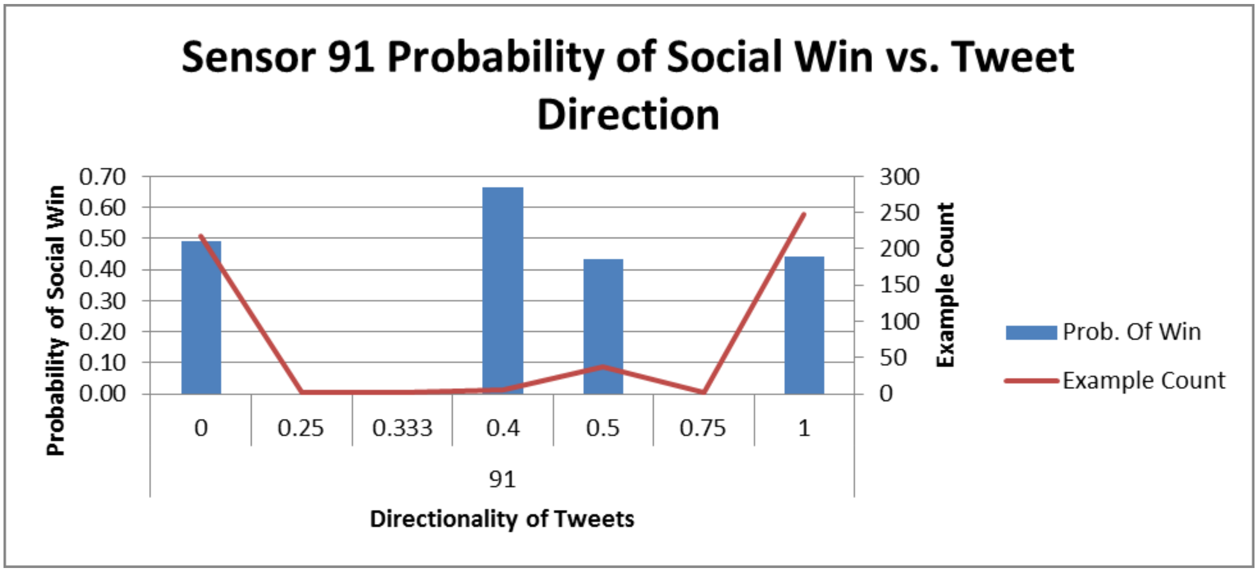


Row Labels	Prob. Of Win	Example Count
0	0.466	253
1	0.457	258

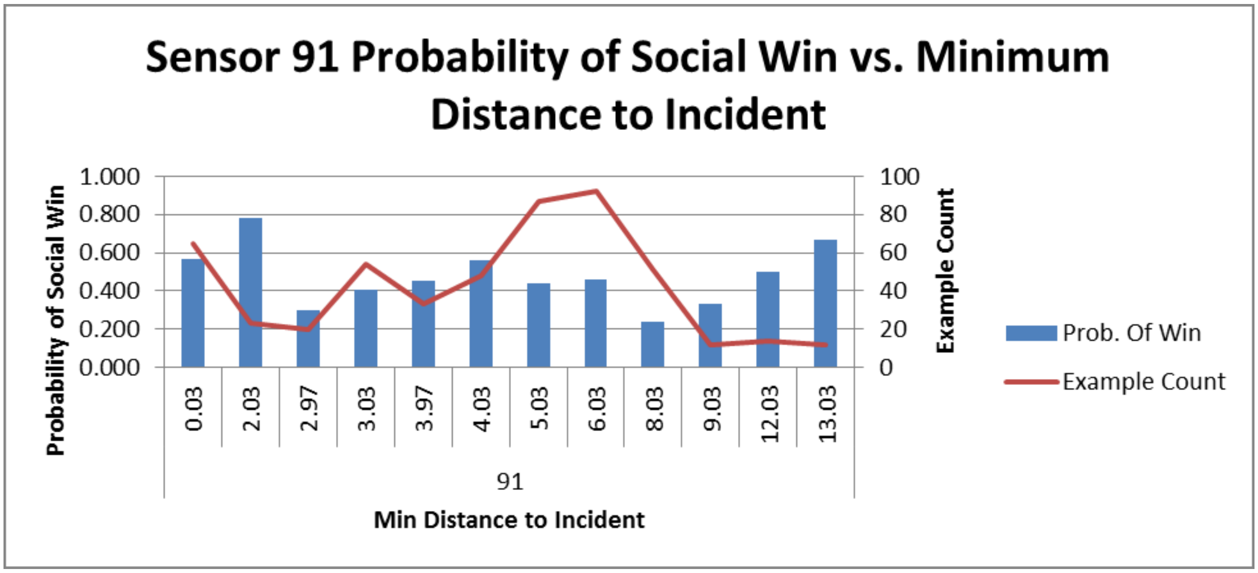


Row Labels	Prob. Of Win	Example Count

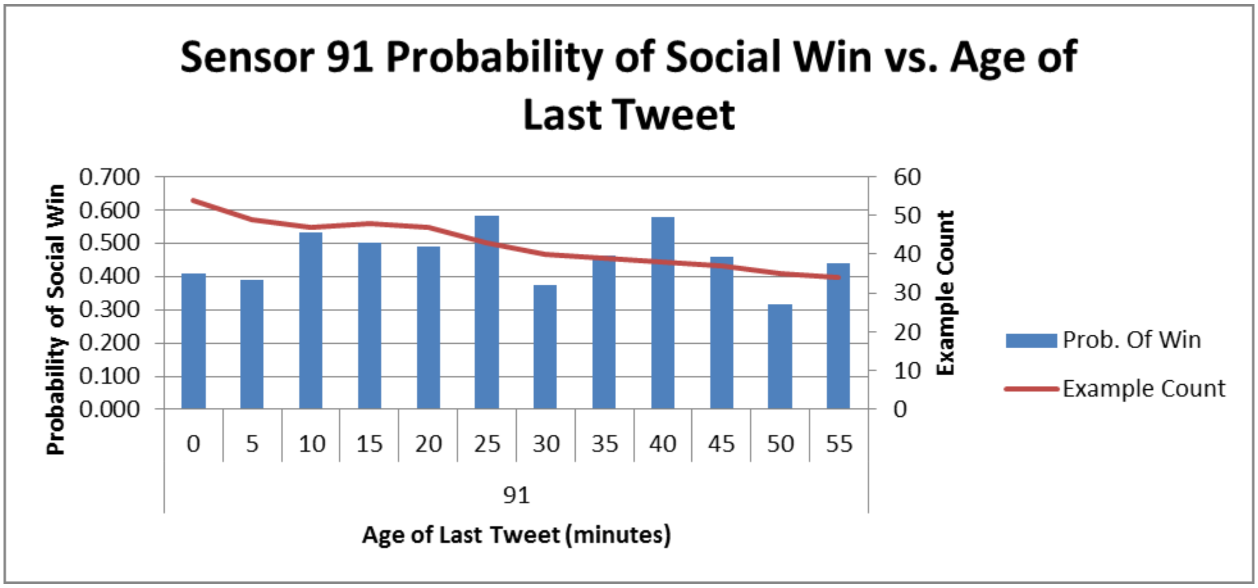
0	0.412	279
1	0.522	232



Row Labels	Prob. Of Win	Example Count
0	0.49	217
0.25	0.00	1
0.333	0.00	2
0.4	0.67	6
0.5	0.43	37
0.75	0.00	1
1	0.44	247

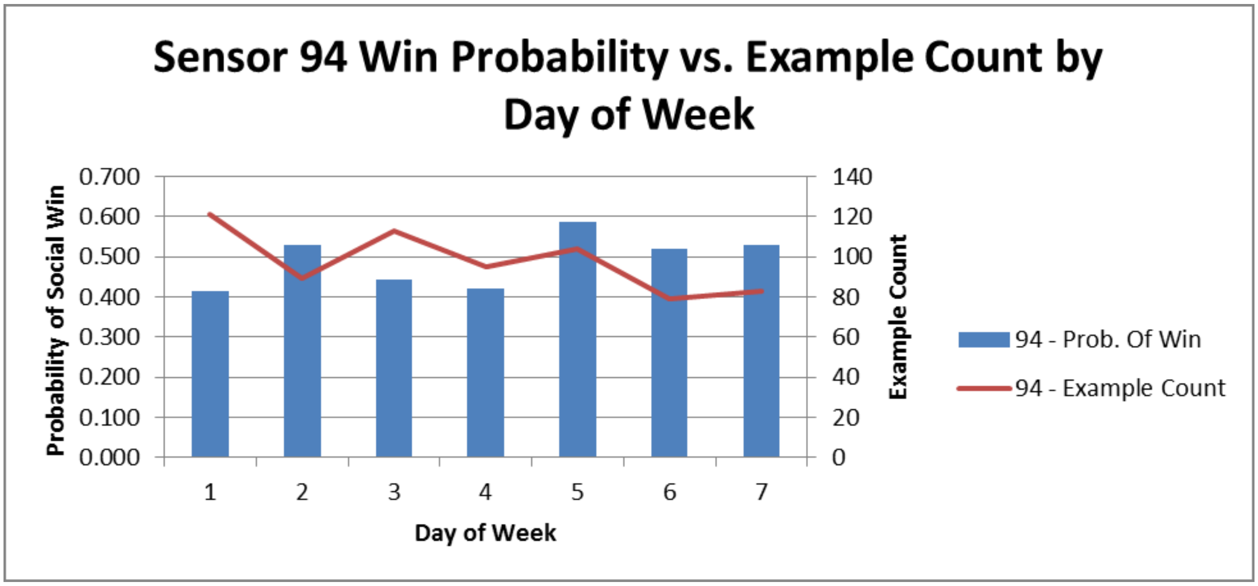


Row Labels	Prob. Of Win	Example Count
0.03	0.569	65
2.03	0.783	23
2.97	0.300	20
3.03	0.407	54
3.97	0.455	33
4.03	0.563	48
5.03	0.437	87
6.03	0.457	92
8.03	0.235	51
9.03	0.333	12
12.03	0.500	14
13.03	0.667	12

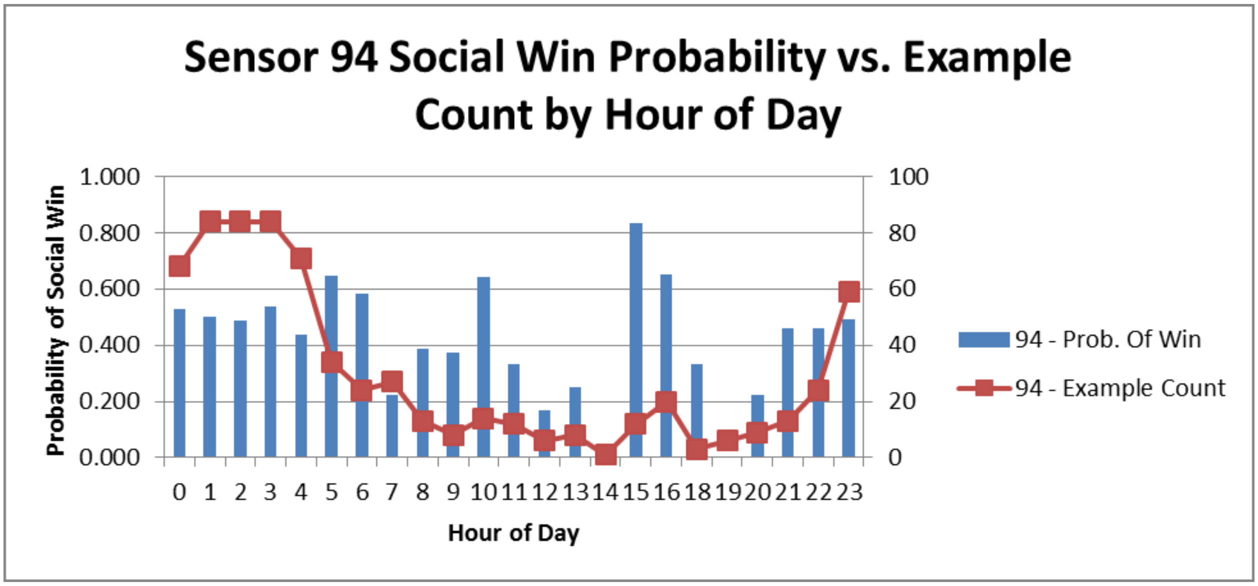


Row Labels	Prob. Of Win	Example Count
0	0.407	54
5	0.388	49
10	0.532	47
15	0.500	48
20	0.489	47
25	0.581	43
30	0.375	40
35	0.462	39
40	0.579	38
45	0.459	37
50	0.314	35
55	0.441	34

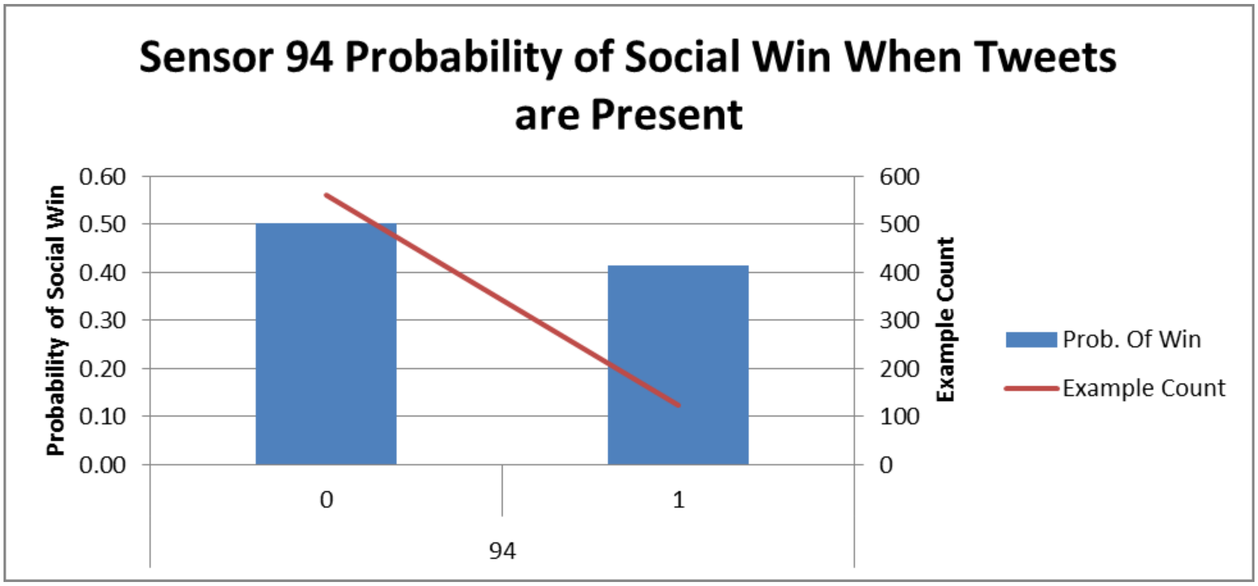
Sensor 94



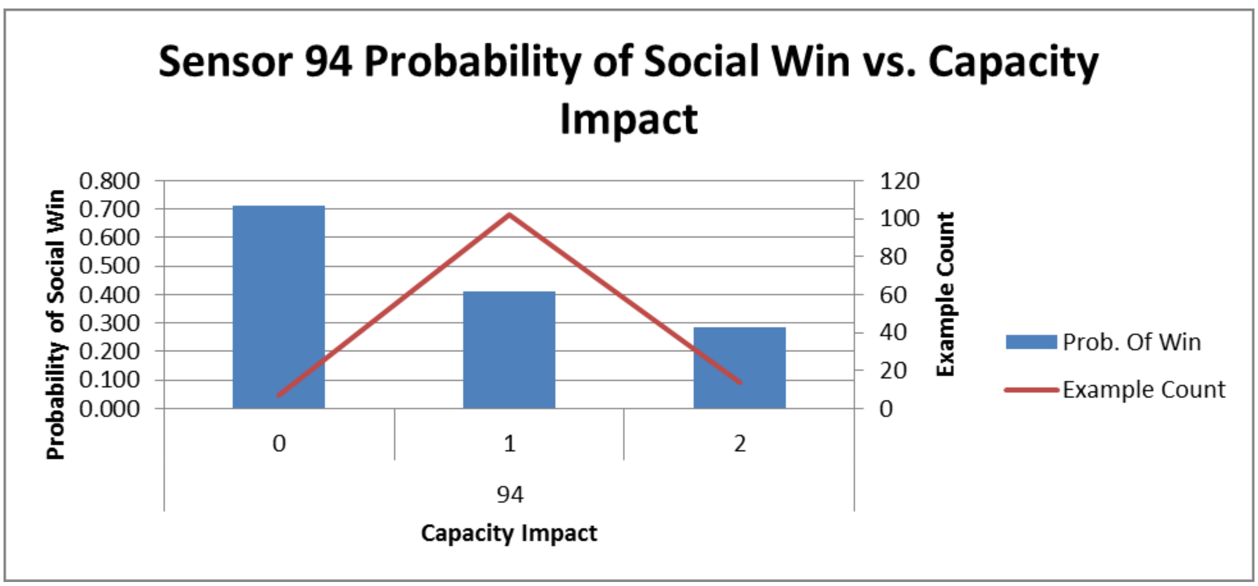
Row Labels	Prob. Of Win	Example Count
1	0.413	121
2	0.528	89
3	0.442	113
4	0.421	95
5	0.587	104
6	0.519	79
7	0.530	83



Row Labels	Prob. Of Win	Example Count
0	0.529	68
1	0.500	84
2	0.488	84
3	0.536	84
4	0.437	71
5	0.647	34
6	0.583	24
7	0.222	27
8	0.385	13
9	0.375	8
10	0.643	14
11	0.333	12
12	0.167	6
13	0.250	8
14	0.000	1
15	0.833	12
16	0.650	20
18	0.333	3
19	0.000	6
20	0.222	9
21	0.462	13
22	0.458	24
23	0.492	59

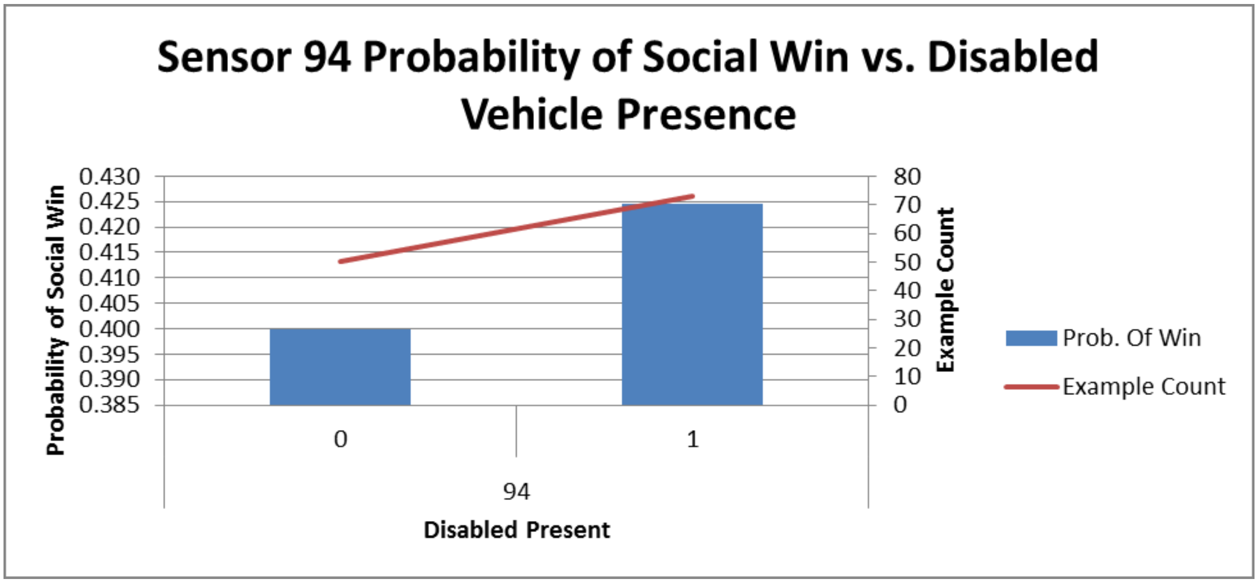


Row Labels	Prob. Of Win	Example Count
0	0.50	561
1	0.41	123

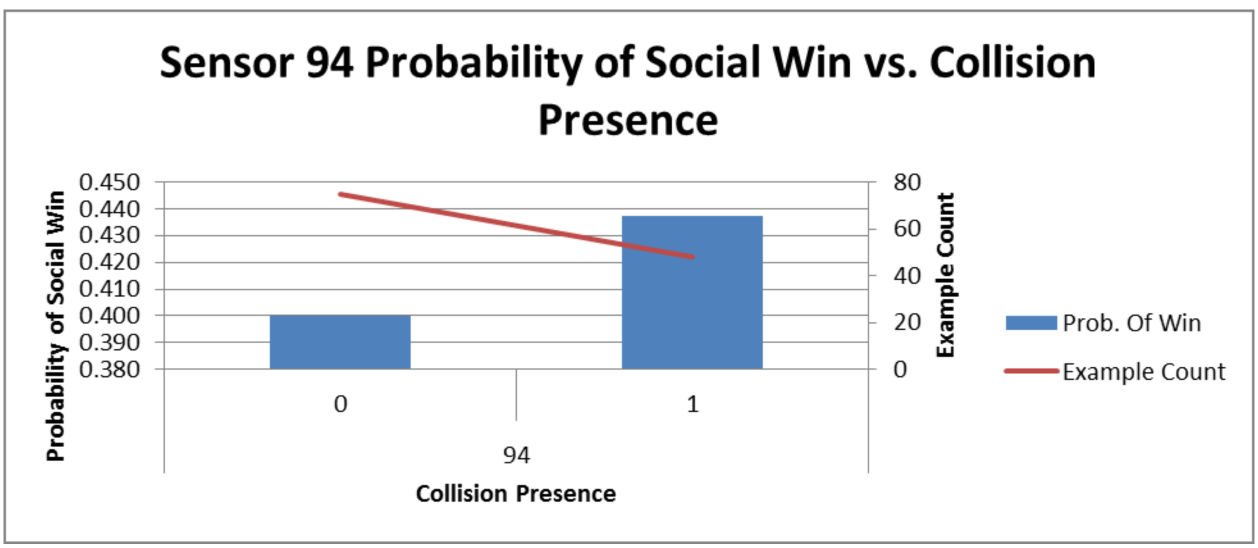


Row Labels	Prob. Of Win	Example Count
0	0.714	7

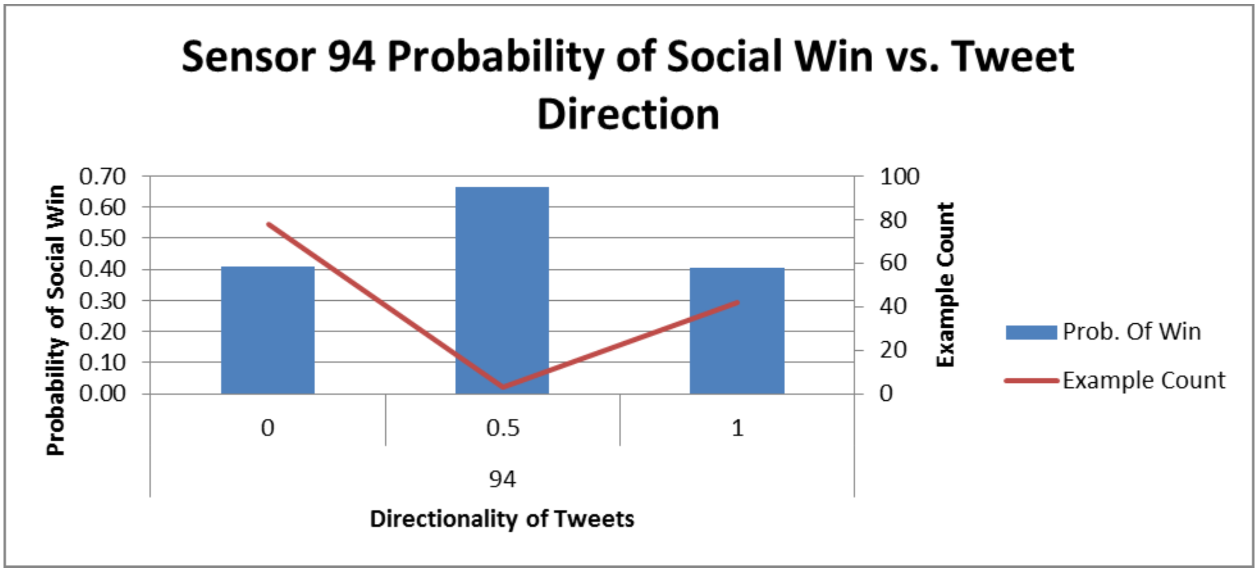
1	0.412	102
2	0.286	14



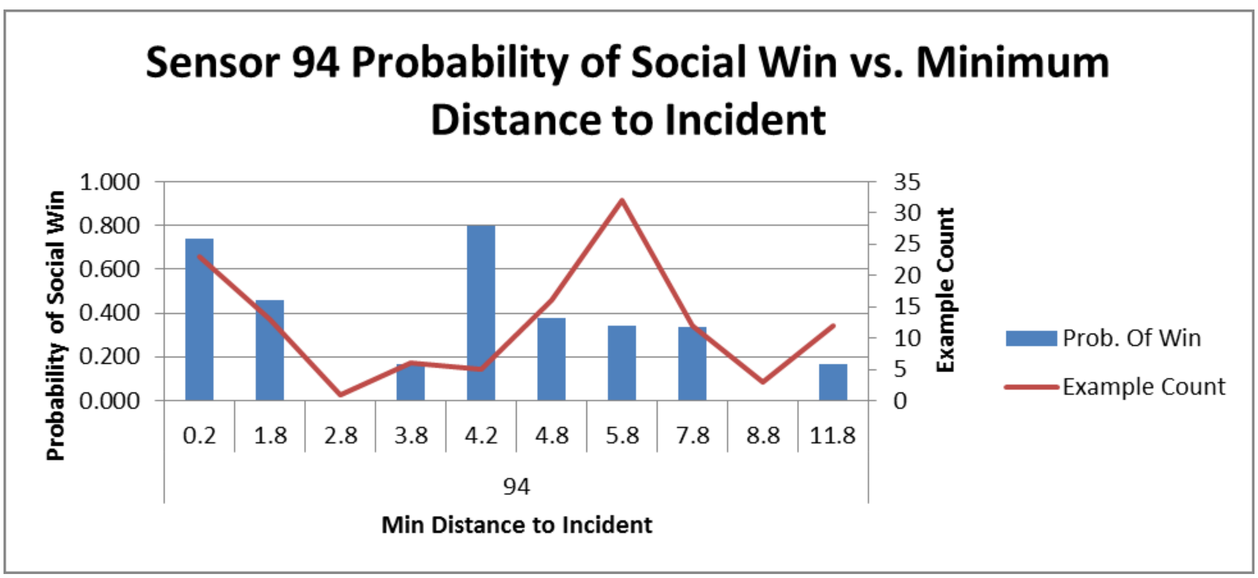
Row Labels	Prob. Of Win	Example Count
0	0.400	50
1	0.425	73



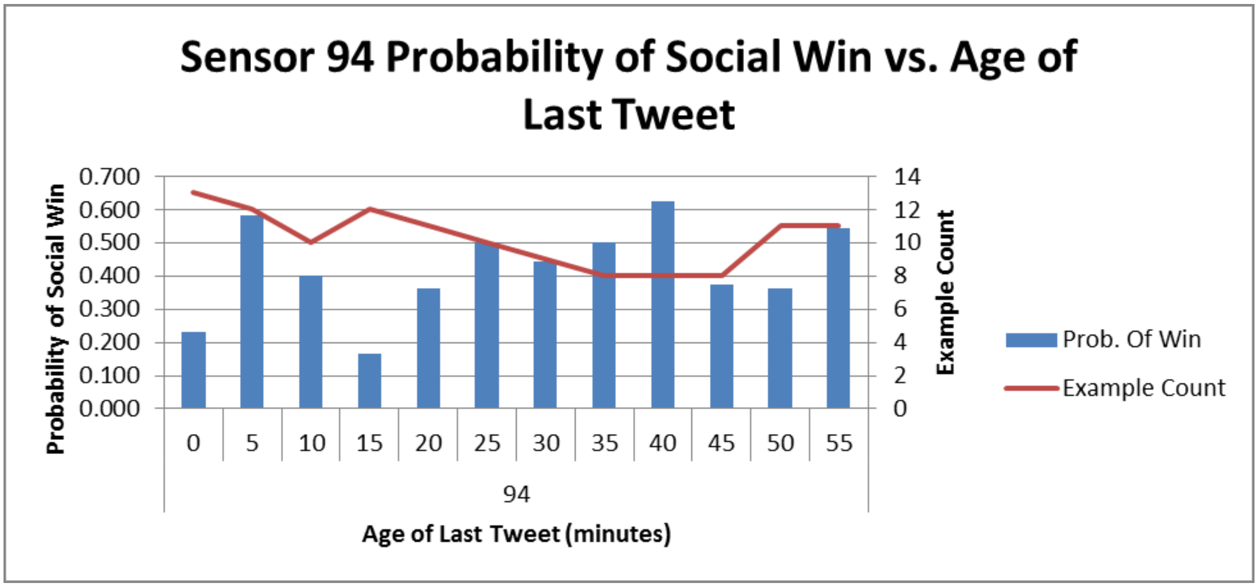
Row Labels	Prob. Of Win	Example Count
0	0.400	75
1	0.438	48



Row Labels	Prob. Of Win	Example Count
0	0.41	78
0.5	0.67	3
1	0.40	42



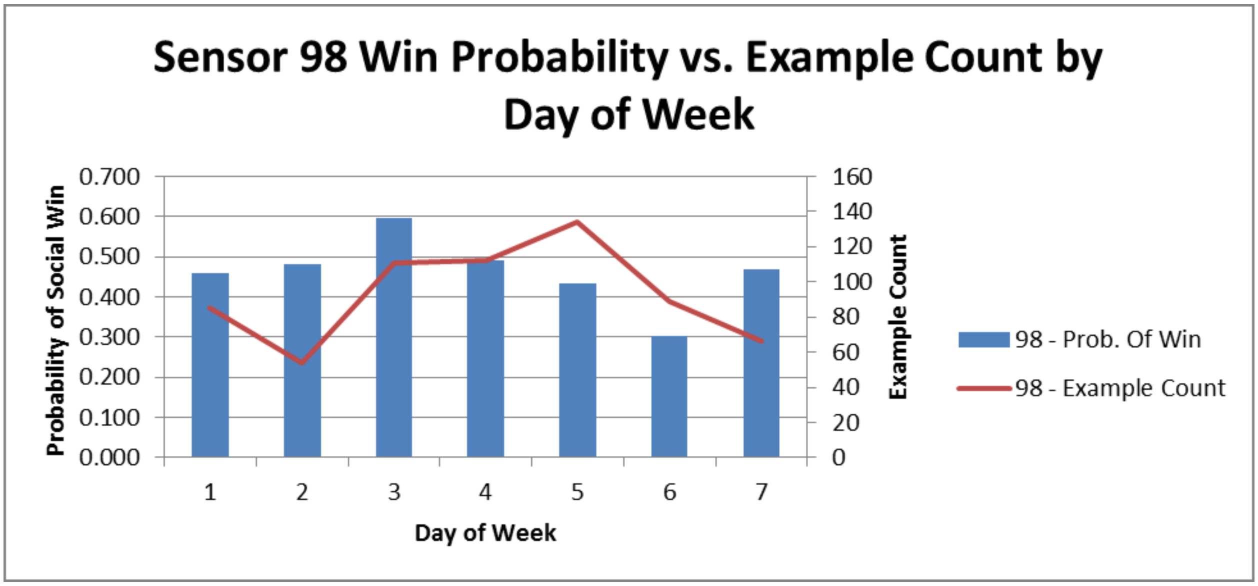
Row Labels	Prob. Of Win	Example Count
0.2	0.739	23
1.8	0.462	13
2.8	0.000	1
3.8	0.167	6
4.2	0.800	5
4.8	0.375	16
5.8	0.344	32
7.8	0.333	12
8.8	0.000	3
11.8	0.167	12



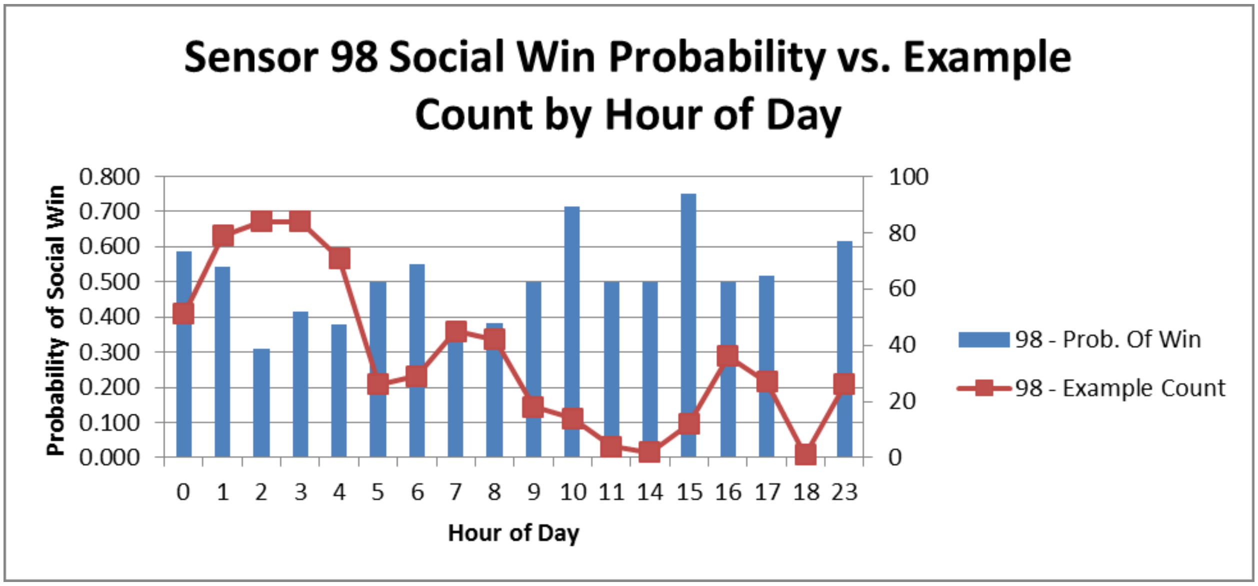
Row Labels	Prob. Of Win	Example Count
0	0.231	13
5	0.583	12
10	0.400	10
15	0.167	12
20	0.364	11
25	0.500	10
30	0.444	9
35	0.500	8
40	0.625	8

45	0.375	8
50	0.364	11
55	0.545	11

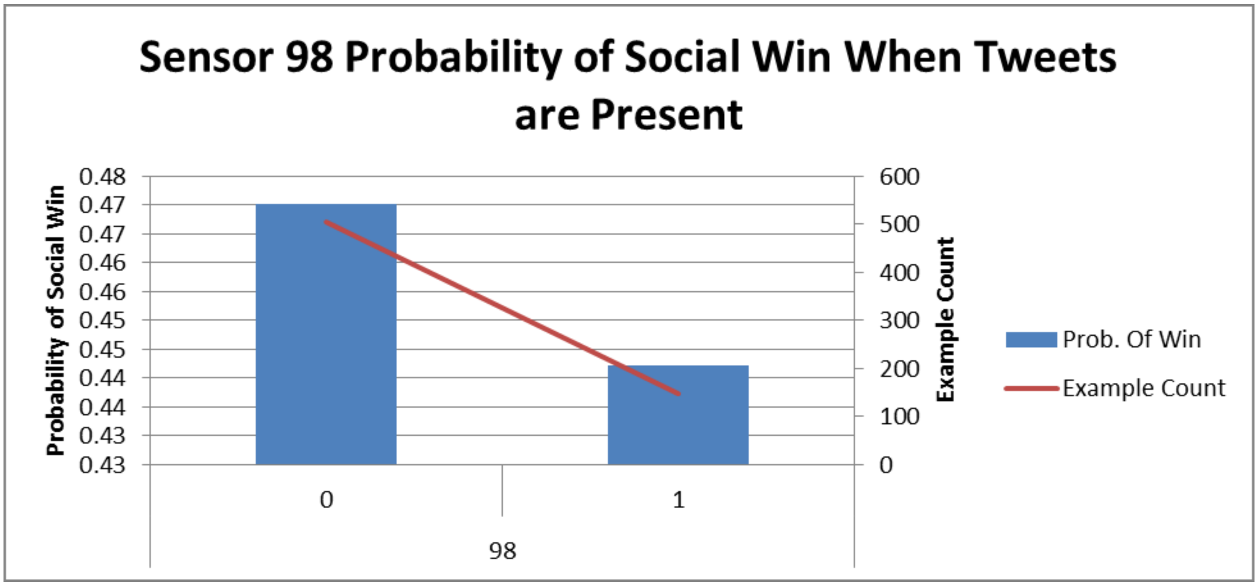
Sensor 98



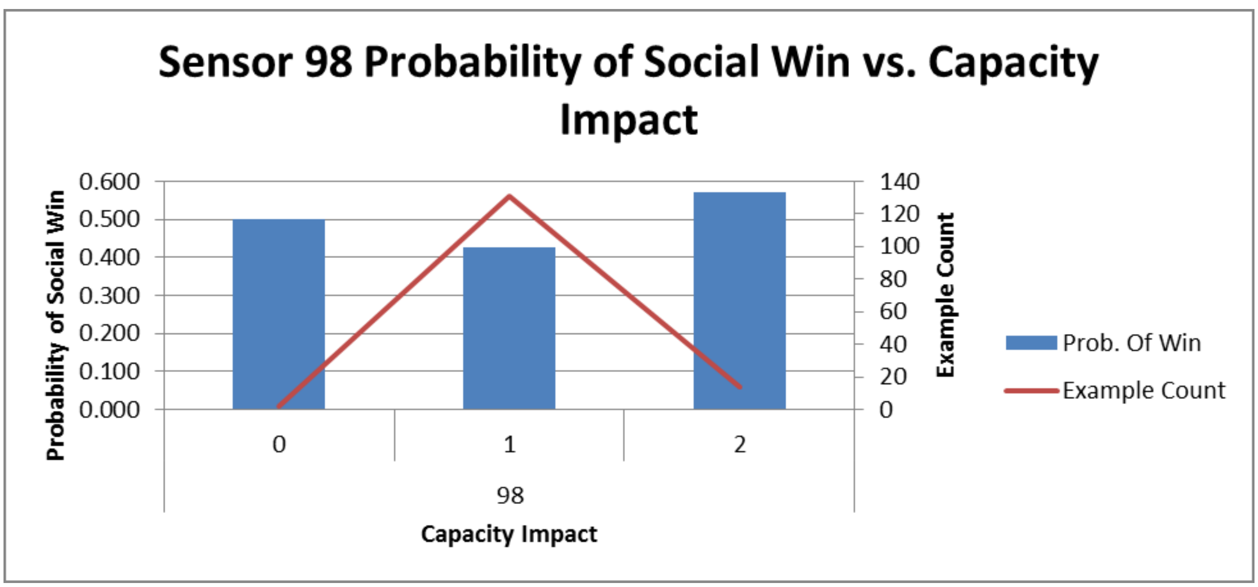
Row Labels	Prob. Of Win	Example Count
1	0.459	85
2	0.481	54
3	0.595	111
4	0.491	112
5	0.433	134
6	0.303	89
7	0.470	66



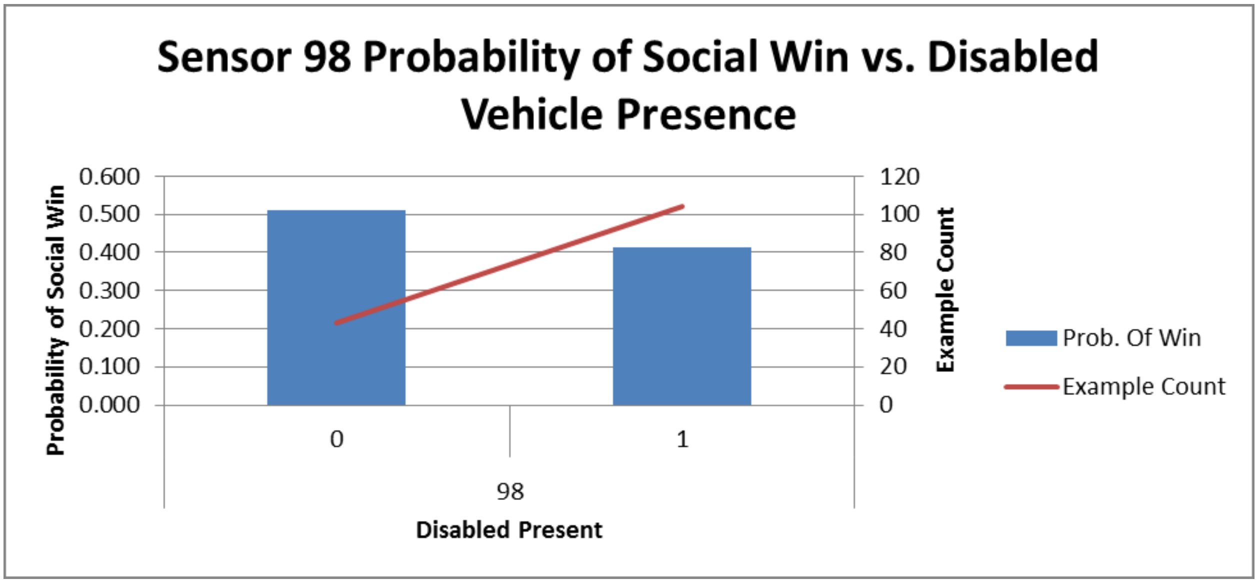
Row Labels	Prob. Of Win	Example Count
0	0.588	51
1	0.544	79
2	0.310	84
3	0.417	84
4	0.380	71
5	0.500	26
6	0.552	29
7	0.378	45
8	0.381	42
9	0.500	18
10	0.714	14
11	0.500	4
14	0.500	2
15	0.750	12
16	0.500	36
17	0.519	27
18	0.000	1
23	0.615	26



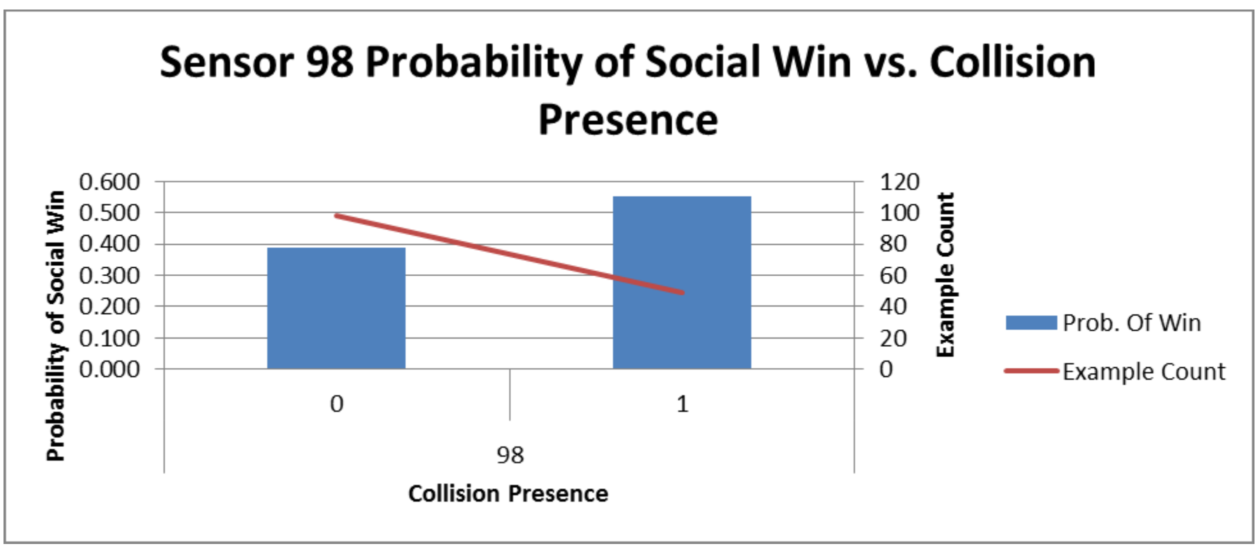
Row Labels	Prob. Of Win	Example Count
0	0.47	504
1	0.44	147



Row Labels	Prob. Of Win	Example Count
0	0.500	2
1	0.427	131
2	0.571	14

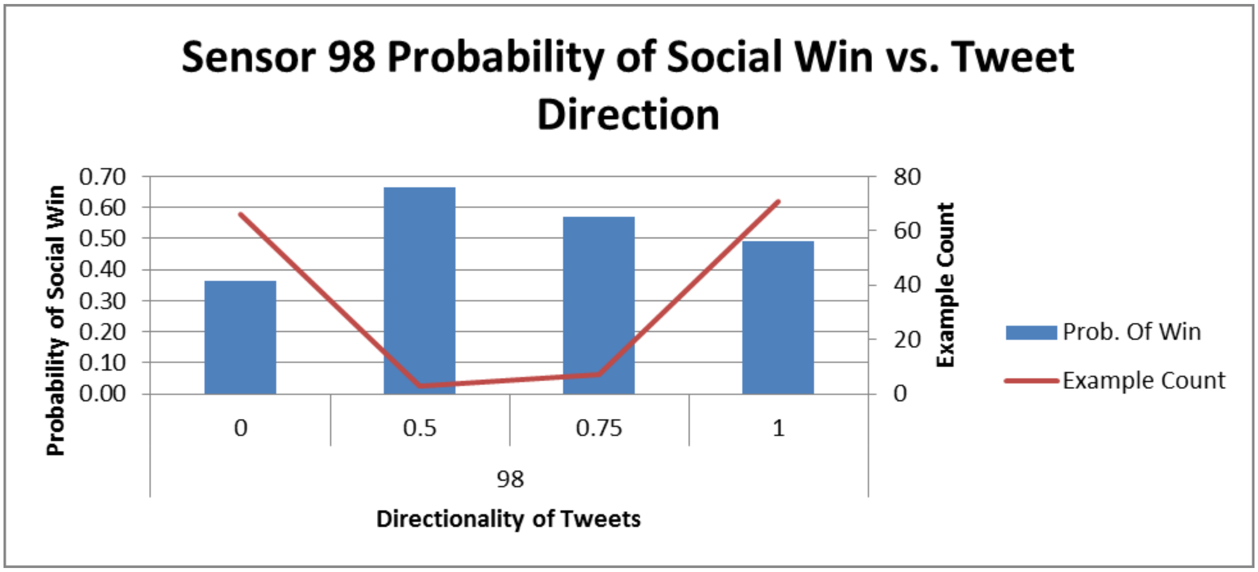


Row Labels	Prob. Of Win	Example Count
0	0.512	43
1	0.413	104

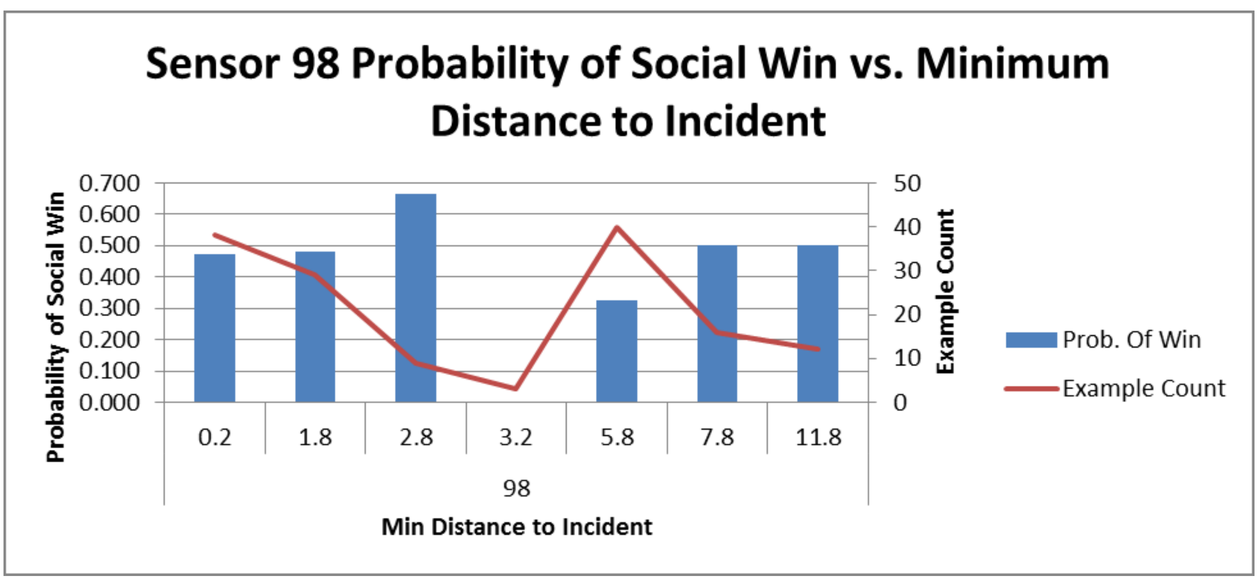


Row Labels	Prob. Of Win	Example Count
0	0.388	98

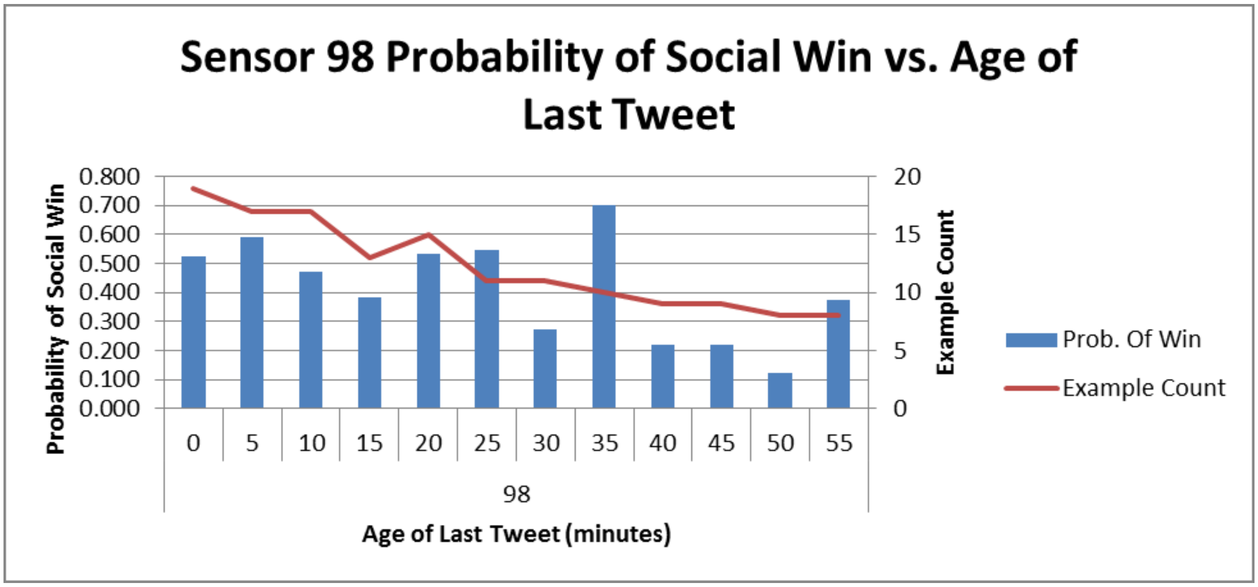
1	0.551	49



Row Labels	Prob. Of Win	Example Count
0	0.36	66
0.5	0.67	3
0.75	0.57	7
1	0.49	71

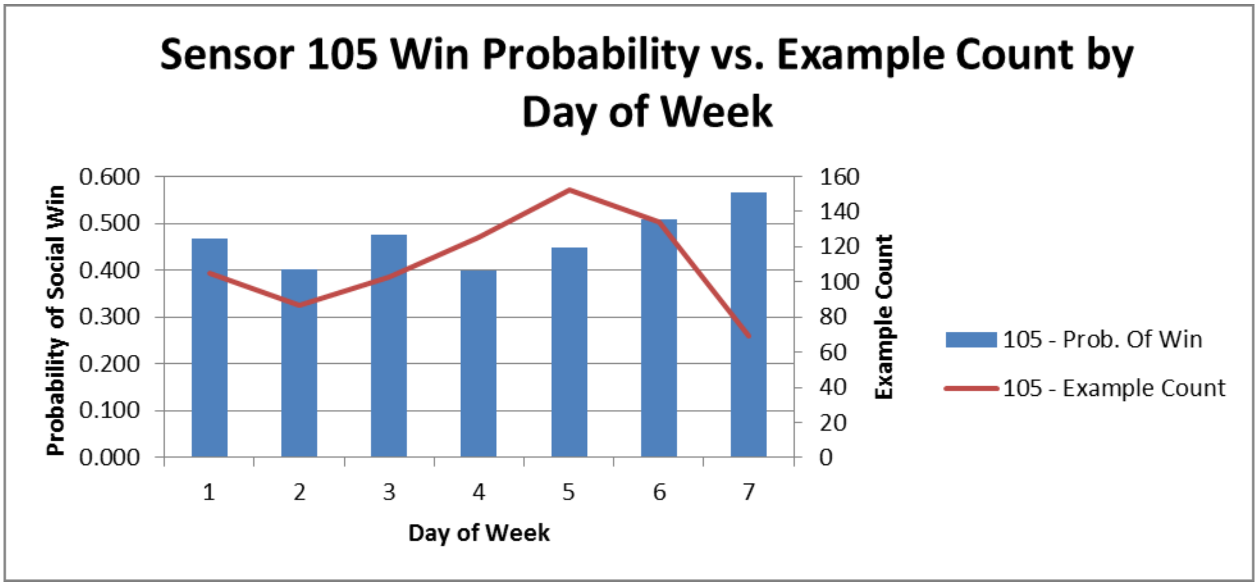


Row Labels	Prob. Of Win	Example Count
0.2	0.474	38
1.8	0.483	29
2.8	0.667	9
3.2	0.000	3
5.8	0.325	40
7.8	0.500	16
11.8	0.500	12

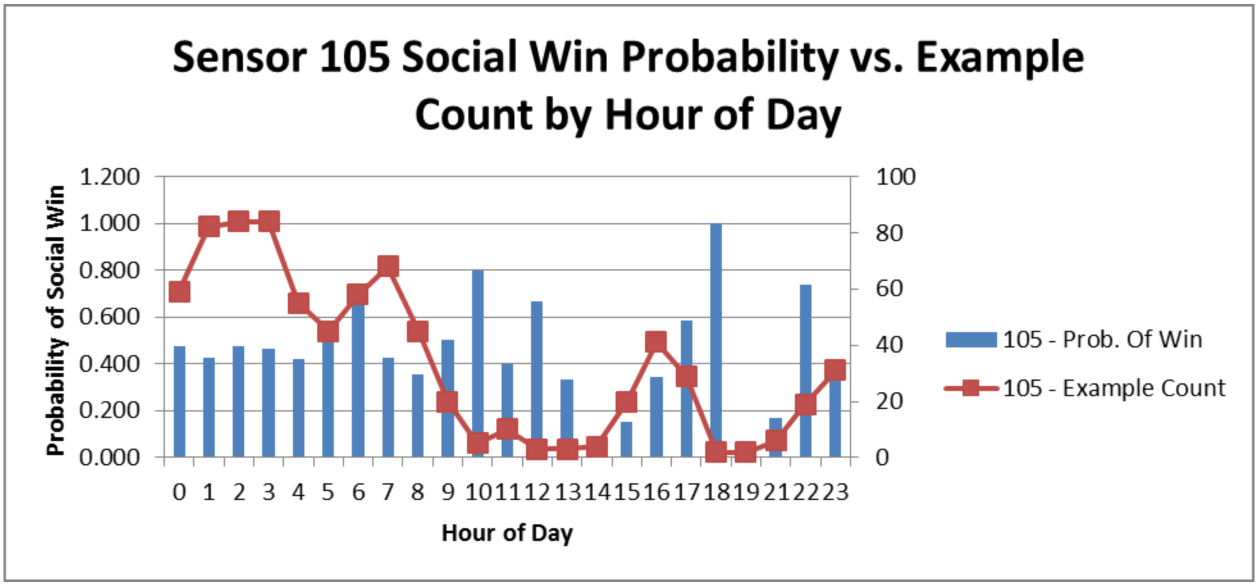


Row Labels	Prob. Of Win	Example Count
0	0.526	19
5	0.588	17
10	0.471	17
15	0.385	13
20	0.533	15
25	0.545	11
30	0.273	11
35	0.700	10
40	0.222	9
45	0.222	9
50	0.125	8
55	0.375	8

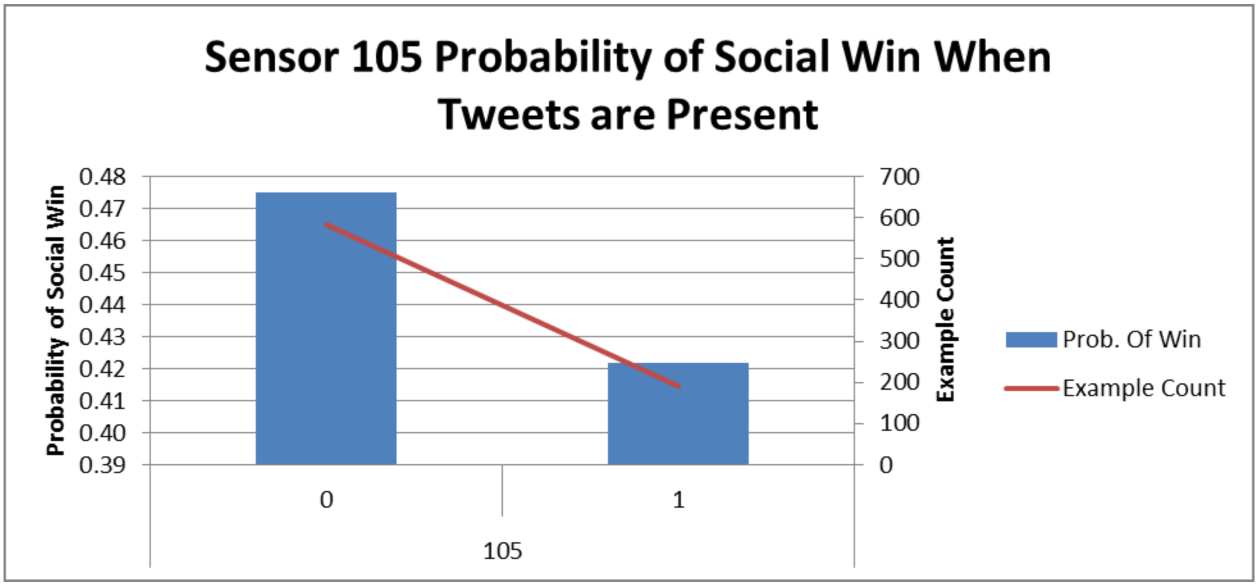
Sensor 105



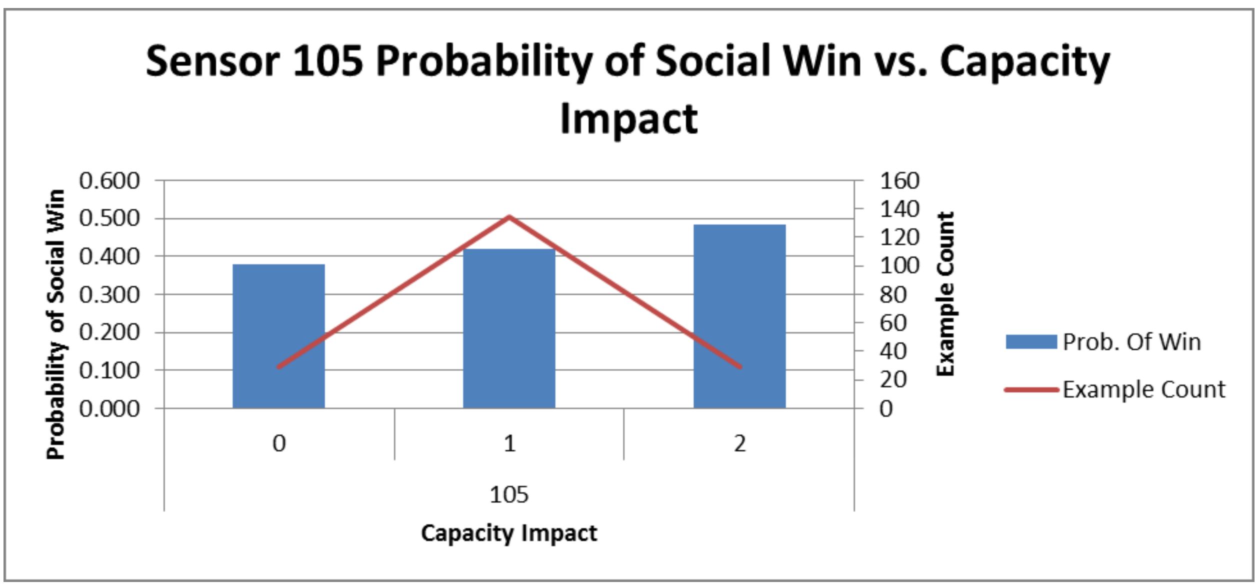
Row Labels	Prob. Of Win	Example Count
1	0.467	105
2	0.402	87
3	0.476	103
4	0.400	125
5	0.447	152
6	0.507	134
7	0.565	69



Row Labels	Prob. Of Win	Example Count
0	0.475	59
1	0.427	82
2	0.476	84
3	0.464	84
4	0.418	55
5	0.533	45
6	0.672	58
7	0.426	68
8	0.356	45
9	0.500	20
10	0.800	5
11	0.400	10
12	0.667	3
13	0.333	3
14	0.000	4
15	0.150	20
16	0.341	41
17	0.586	29
18	1.000	2
19	0.000	2
21	0.167	6
22	0.737	19
23	0.419	31

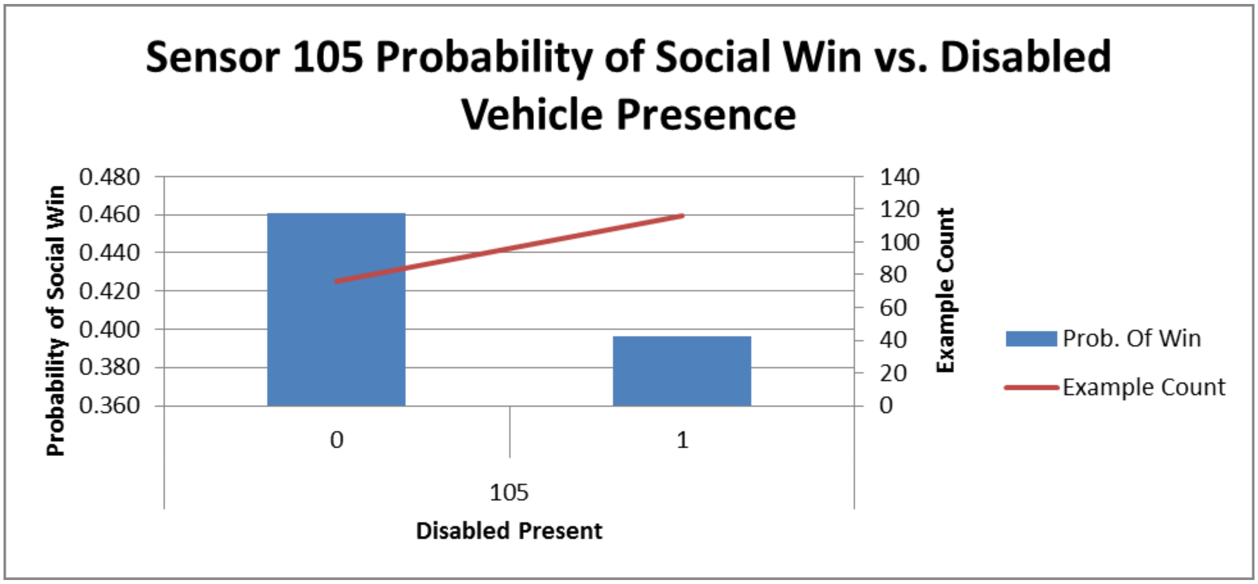


Row Labels	Prob. Of Win	Example Count
0	0.48	583
1	0.42	192

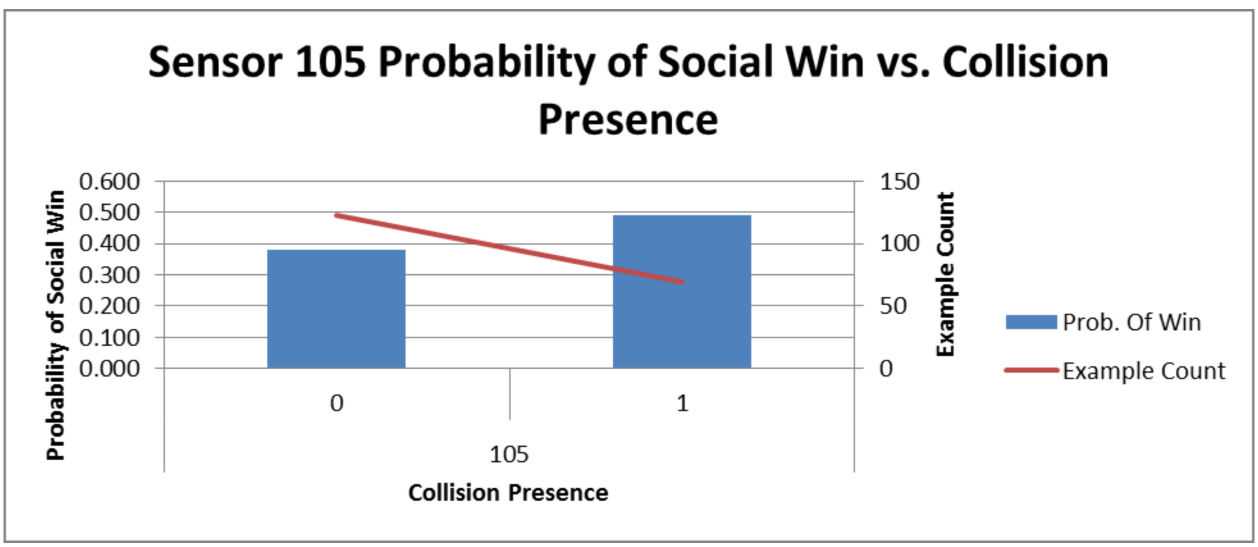


Row Labels	Prob. Of Win	Example Count
0	0.379	29
1	0.418	134

2	0.483	29

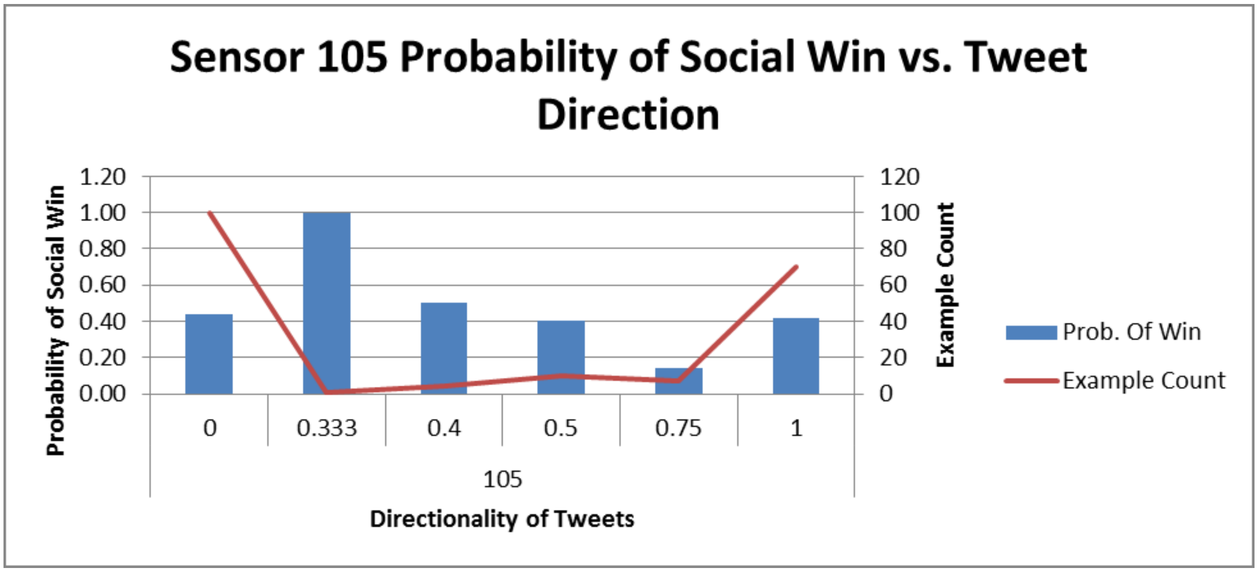


Row Labels	Prob. Of Win	Example Count
0	0.461	76
1	0.397	116

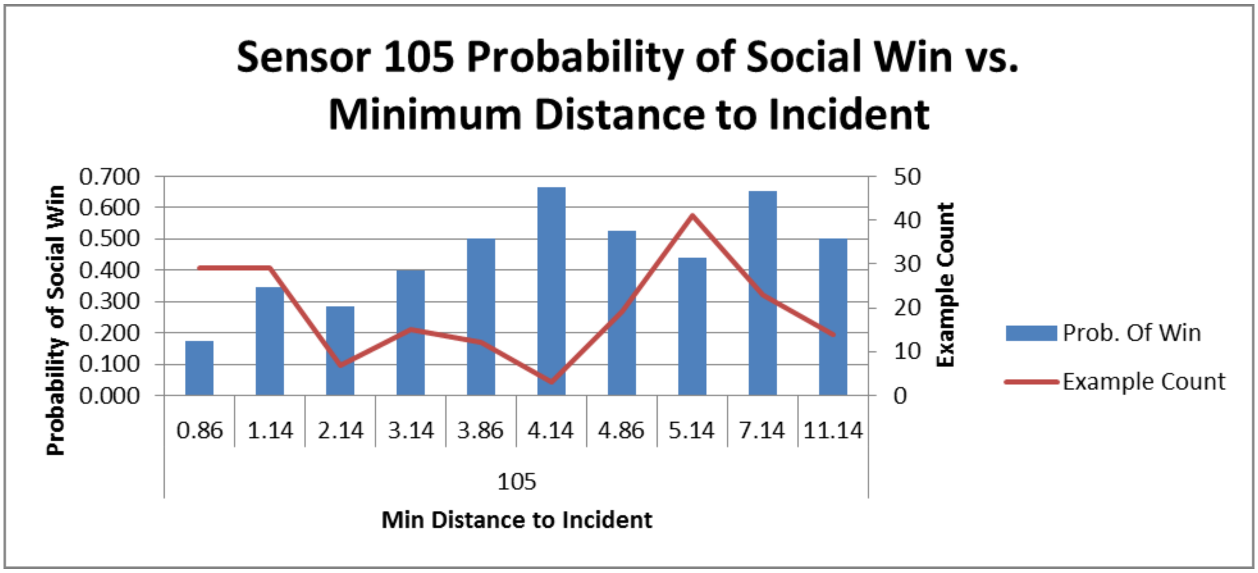


Row Labels	Prob. Of Win	Example Count

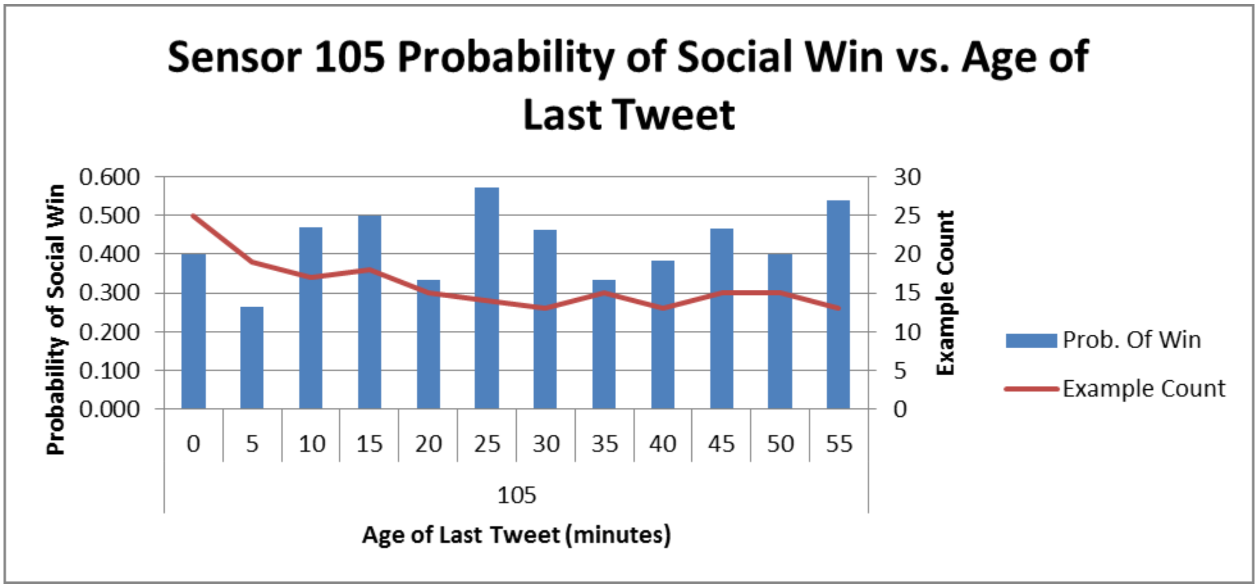
0	0.382	123
1	0.493	69



Row Labels	Prob. Of Win	Example Count
0	0.44	100
0.333	1.00	1
0.4	0.50	4
0.5	0.40	10
0.75	0.14	7
1	0.41	70

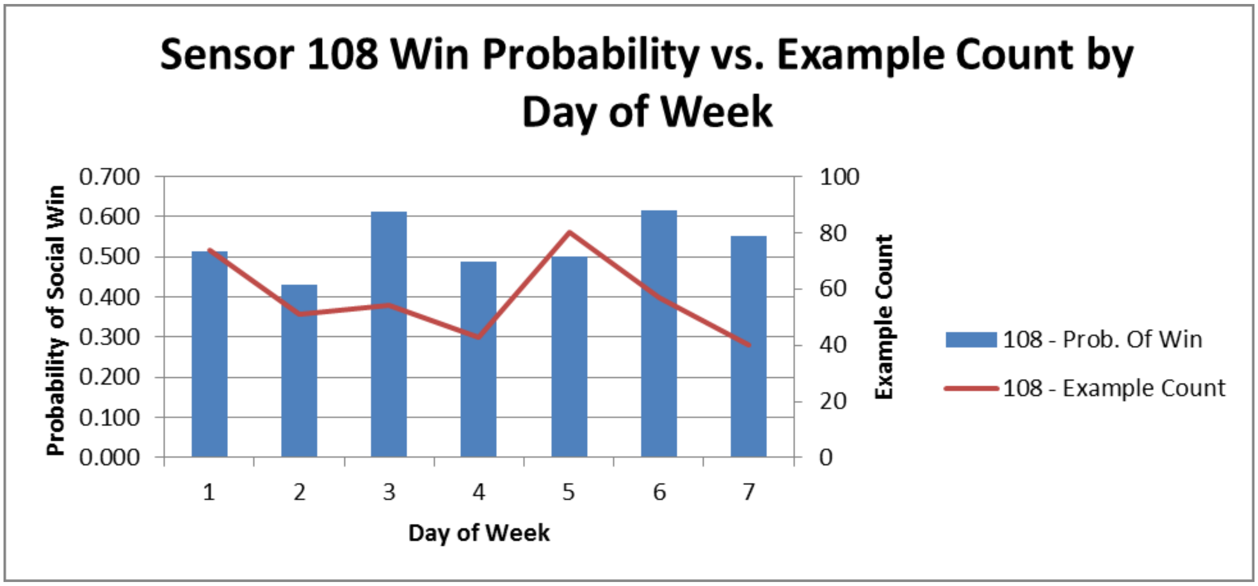


Row Labels	Prob. Of Win	Example Count
0.86	0.172	29
1.14	0.345	29
2.14	0.286	7
3.14	0.400	15
3.86	0.500	12
4.14	0.667	3
4.86	0.526	19
5.14	0.439	41
7.14	0.652	23
11.14	0.500	14

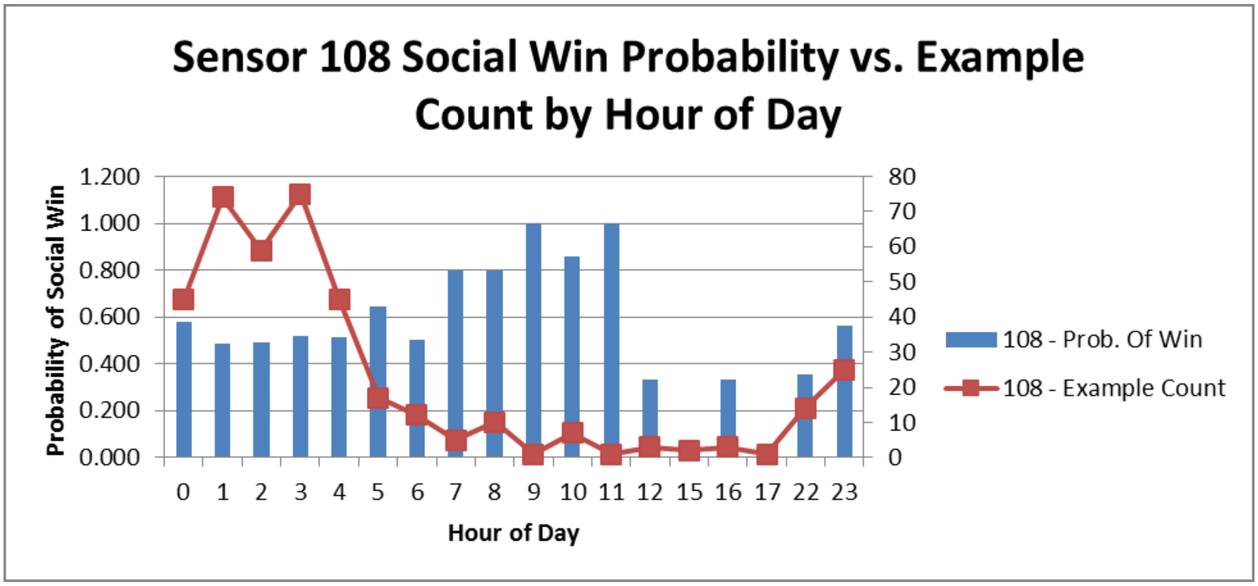


Row Labels	Prob. Of Win	Example Count
0	0.400	25
5	0.263	19
10	0.471	17
15	0.500	18
20	0.333	15
25	0.571	14
30	0.462	13
35	0.333	15
40	0.385	13
45	0.467	15
50	0.400	15
55	0.538	13

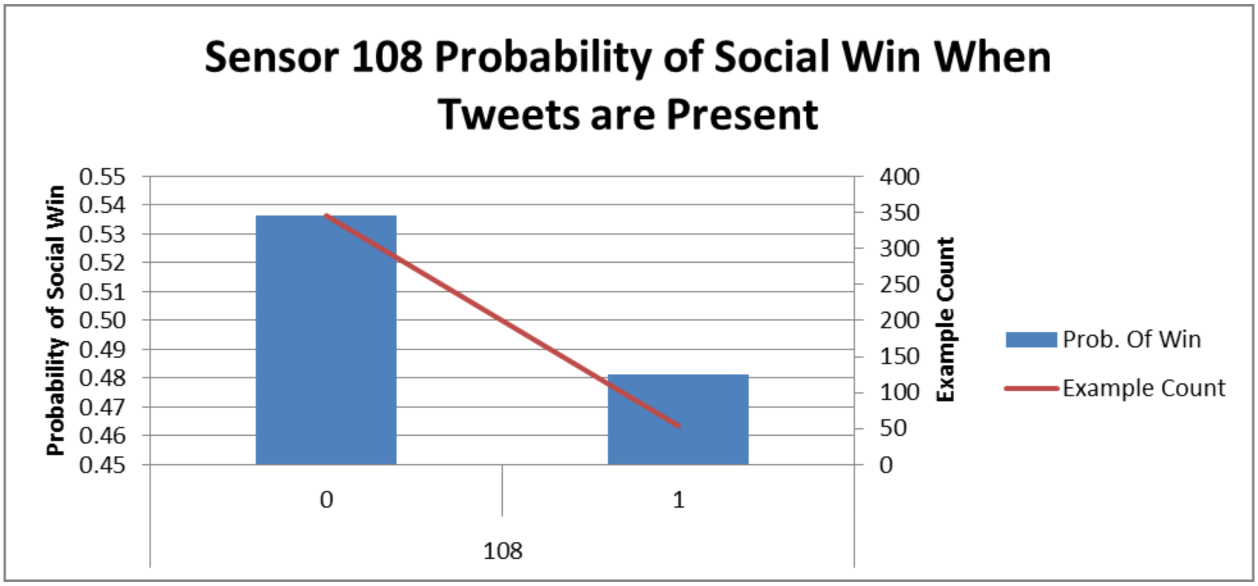
Sensor 108



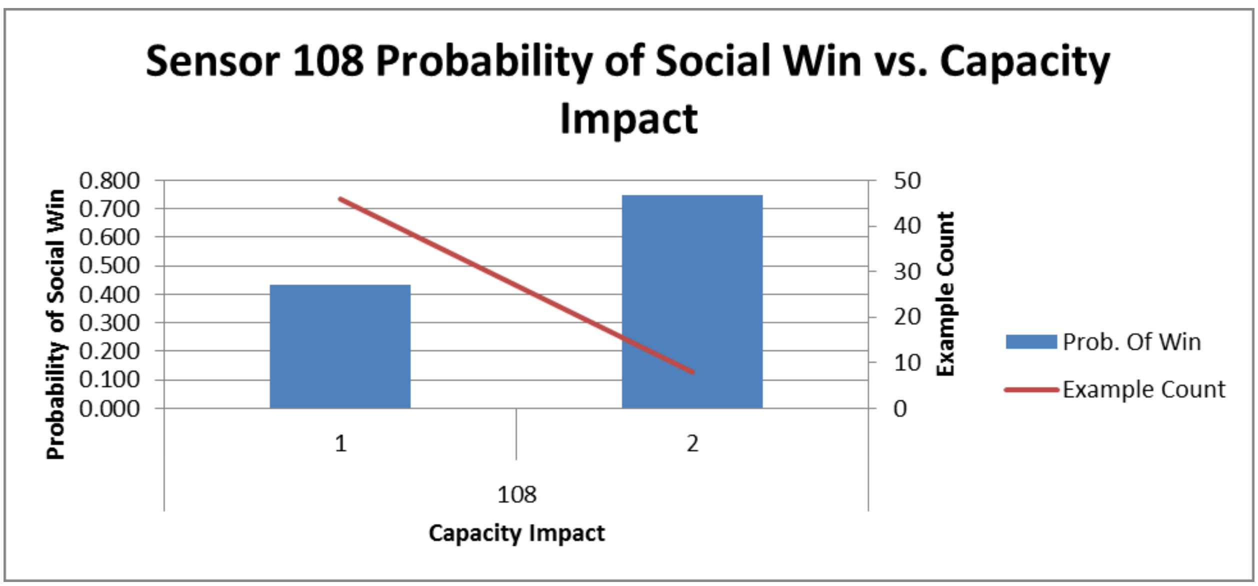
Row Labels	Prob. Of Win	Example Count
1	0.514	74
2	0.431	51
3	0.611	54
4	0.488	43
5	0.500	80
6	0.614	57
7	0.550	40



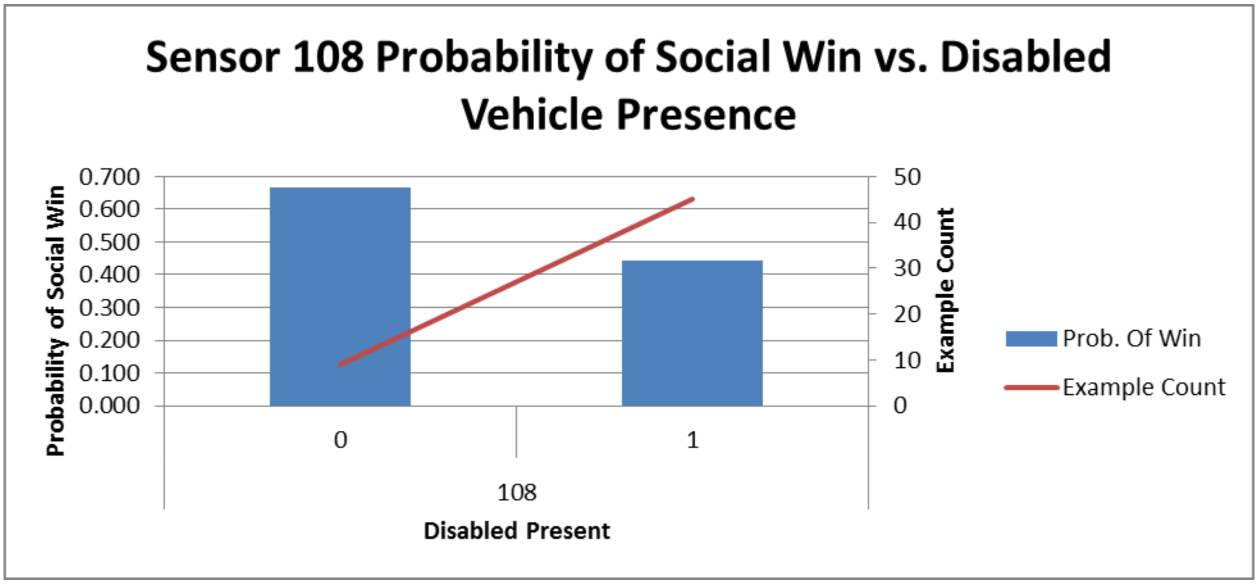
Row Labels	Prob. Of Win	Example Count
0	0.578	45
1	0.486	74
2	0.492	59
3	0.520	75
4	0.511	45
5	0.647	17
6	0.500	12
7	0.800	5
8	0.800	10
9	1.000	1
10	0.857	7
11	1.000	1
12	0.333	3
15	0.000	2
16	0.333	3
17	0.000	1
22	0.357	14
23	0.560	25



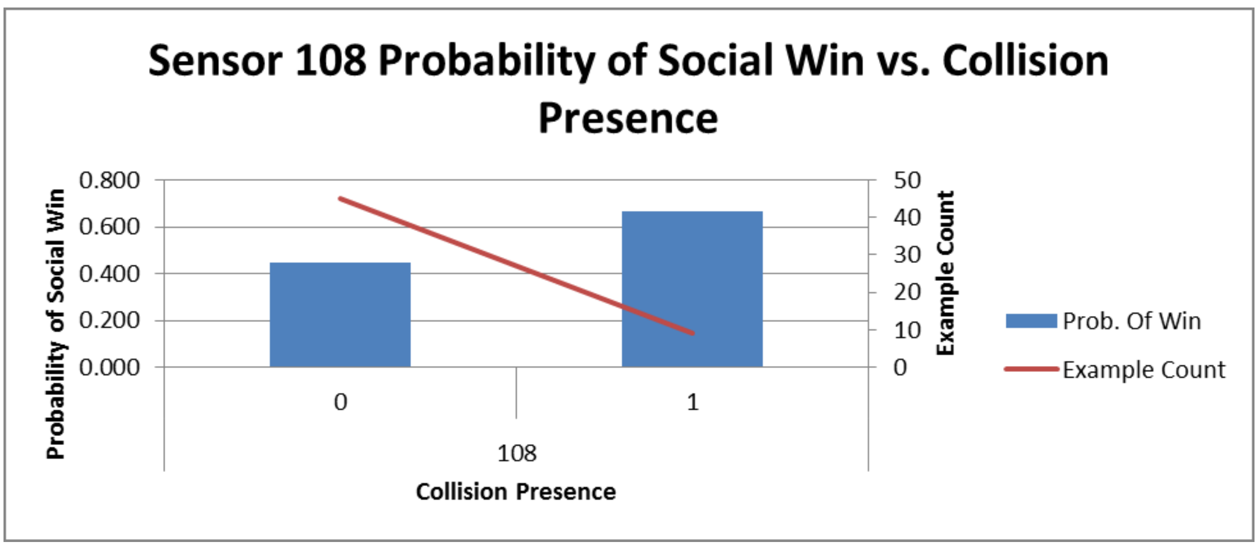
Row Labels	Prob. Of Win	Example Count
0	0.54	345
1	0.48	54



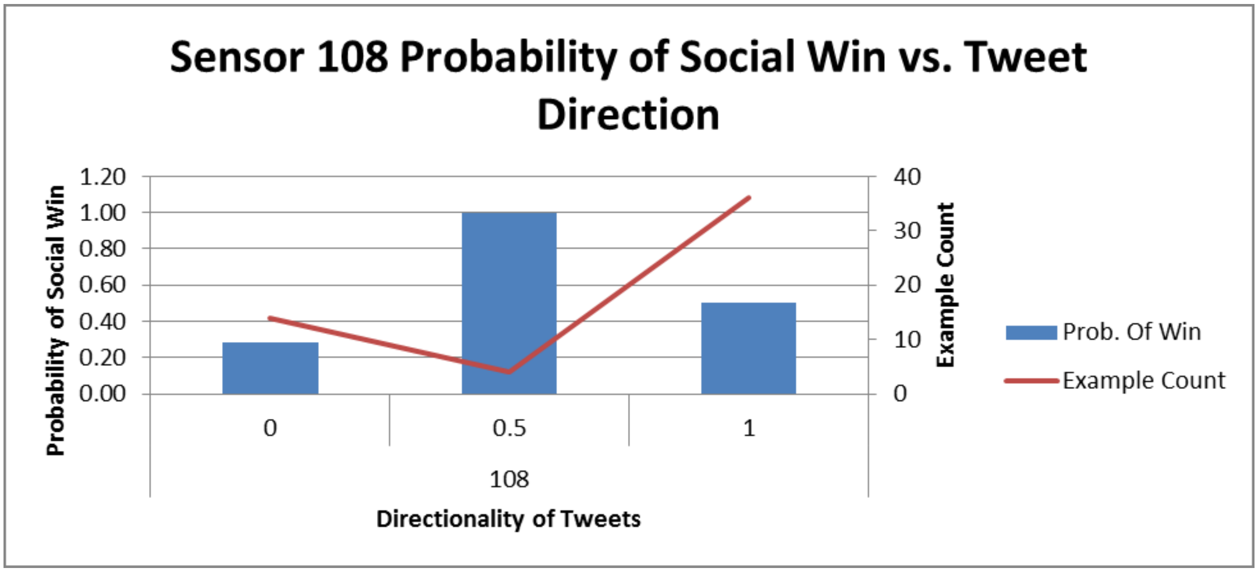
Row Labels	Prob. Of Win	Example Count
1	0.435	46
2	0.750	8



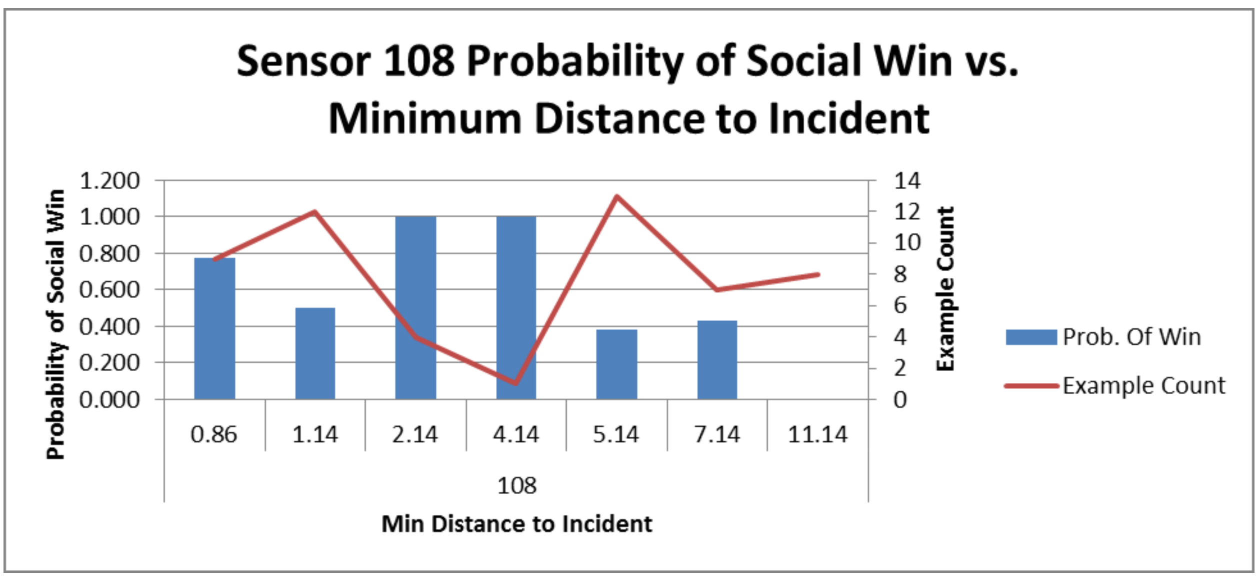
Row Labels	Prob. Of Win	Example Count
0	0.667	9
1	0.444	45



Row Labels	Prob. Of Win	Example Count
0	0.444	45
1	0.667	9

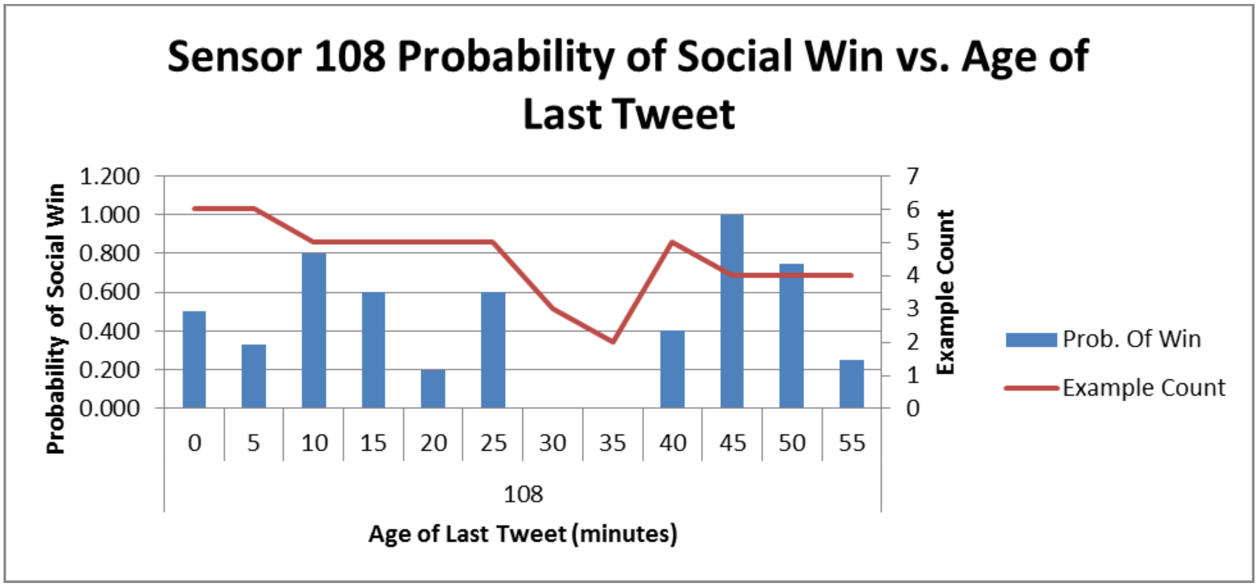


Row Labels	Prob. Of Win	Example Count
0	0.29	14
0.5	1.00	4
1	0.50	36



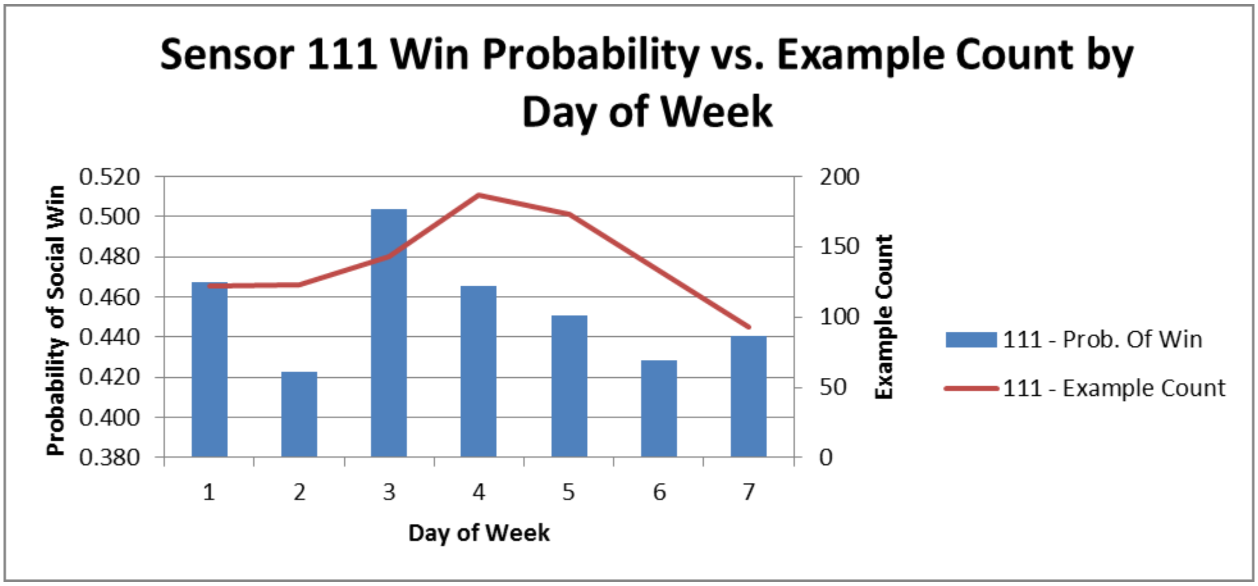
Row Labels	Prob. Of Win	Example Count
0.86	0.778	9
1.14	0.500	12
2.14	1.000	4
4.14	1.000	1

5.14	0.385	13
7.14	0.429	7
11.14	0.000	8

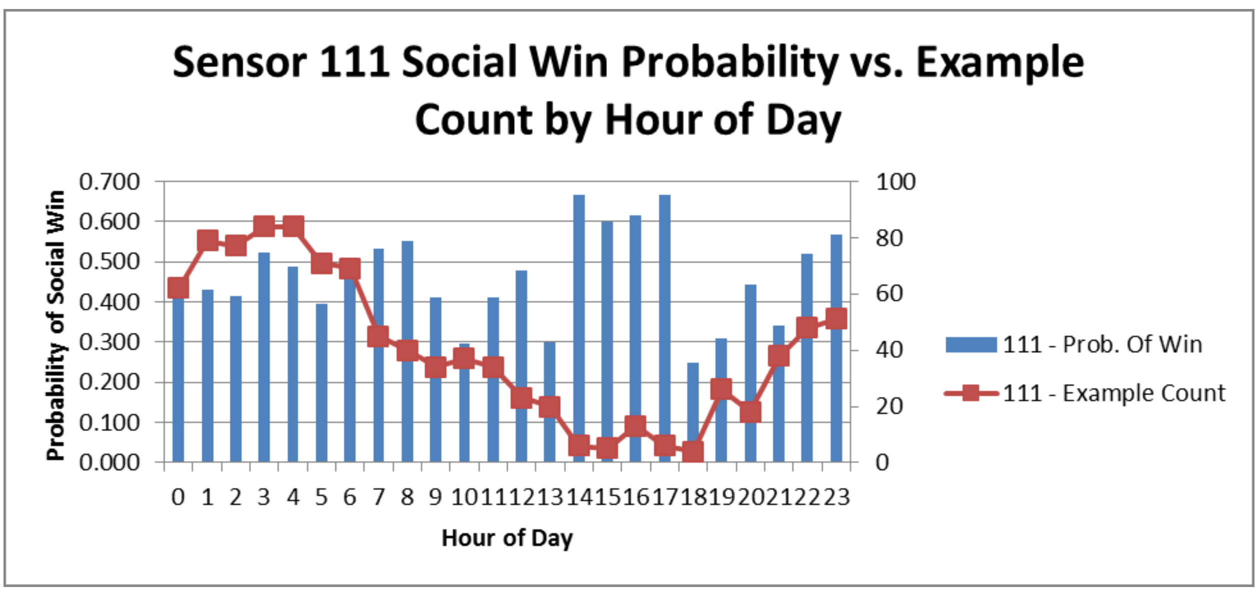


Row Labels	Prob. Of Win	Example Count
0	0.500	6
5	0.333	6
10	0.800	5
15	0.600	5
20	0.200	5
25	0.600	5
30	0.000	3
35	0.000	2
40	0.400	5
45	1.000	4
50	0.750	4
55	0.250	4

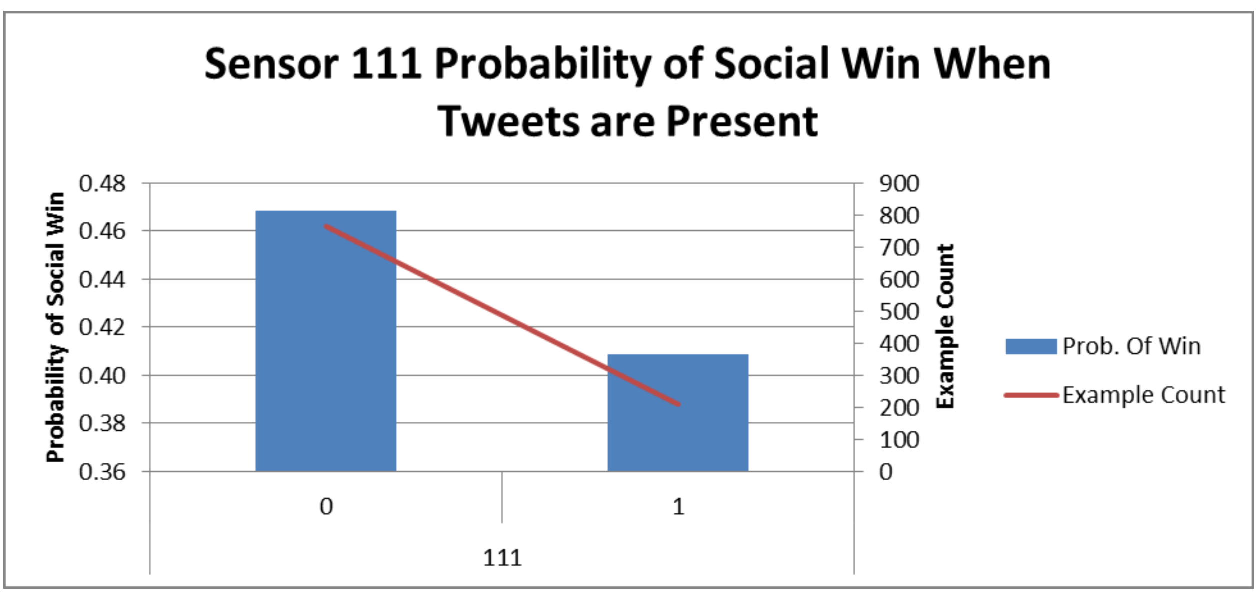
Sensor 111



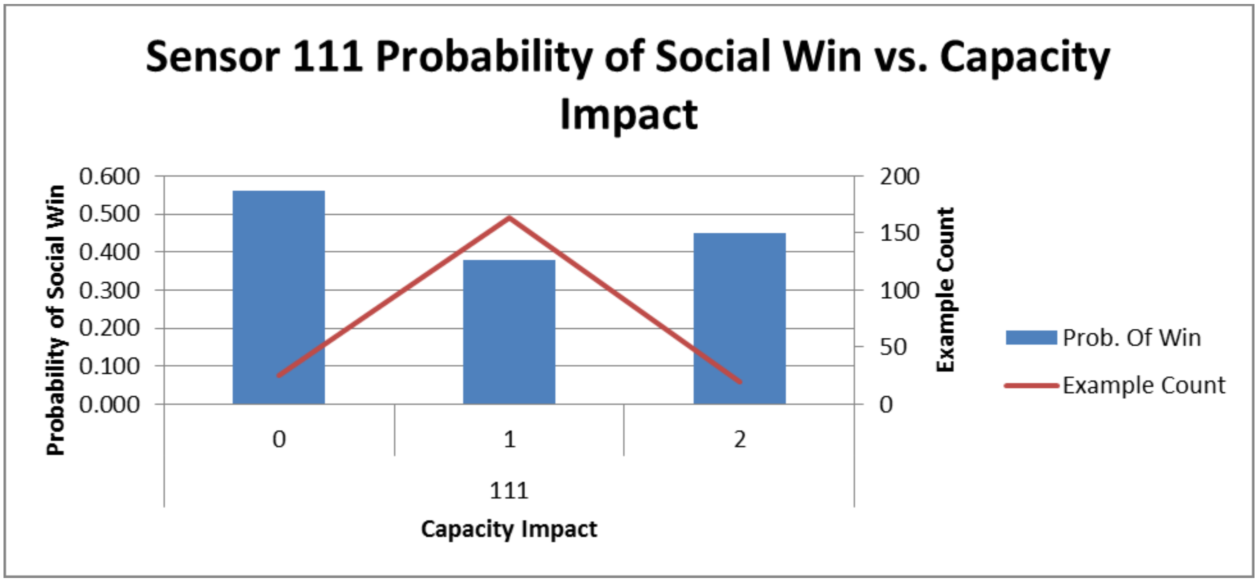
Row Labels	Prob. Of Win	Example Count
1	0.467	122
2	0.423	123
3	0.503	143
4	0.465	187
5	0.451	173
6	0.429	133
7	0.441	93



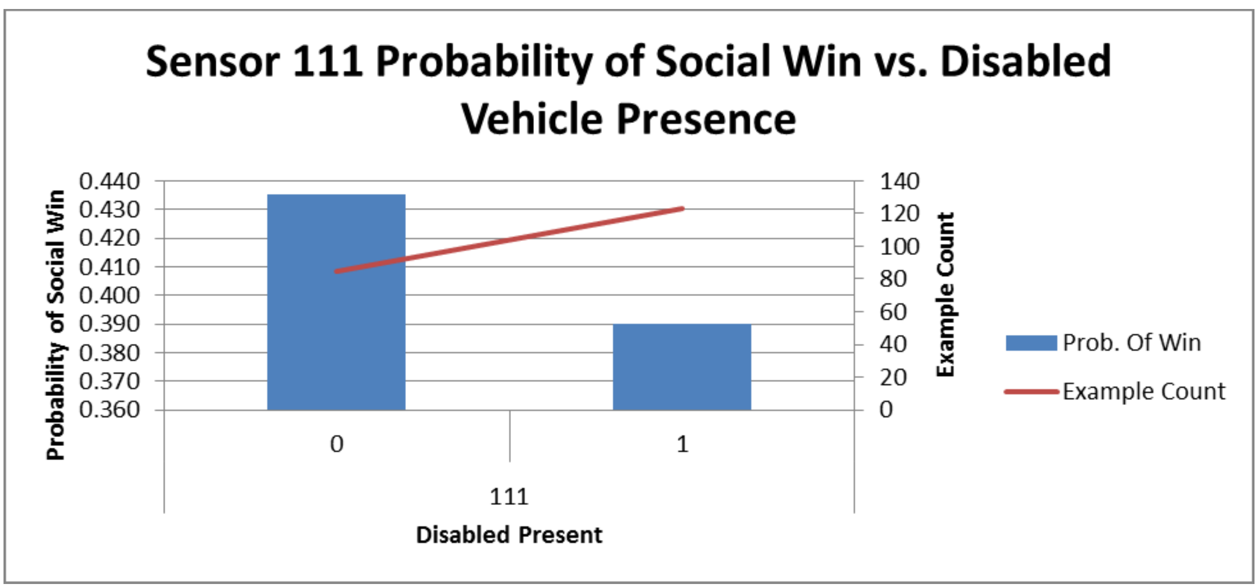
Row Labels	Prob. Of Win	Example Count
0	0.435	62
1	0.430	79
2	0.416	77
3	0.524	84
4	0.488	84
5	0.394	71
6	0.478	69
7	0.533	45
8	0.550	40
9	0.412	34
10	0.297	37
11	0.412	34
12	0.478	23
13	0.300	20
14	0.667	6
15	0.600	5
16	0.615	13
17	0.667	6
18	0.250	4
19	0.308	26
20	0.444	18
21	0.342	38
22	0.521	48
23	0.569	51



Row Labels	Prob. Of Win	Example Count
0	0.47	766
1	0.41	208

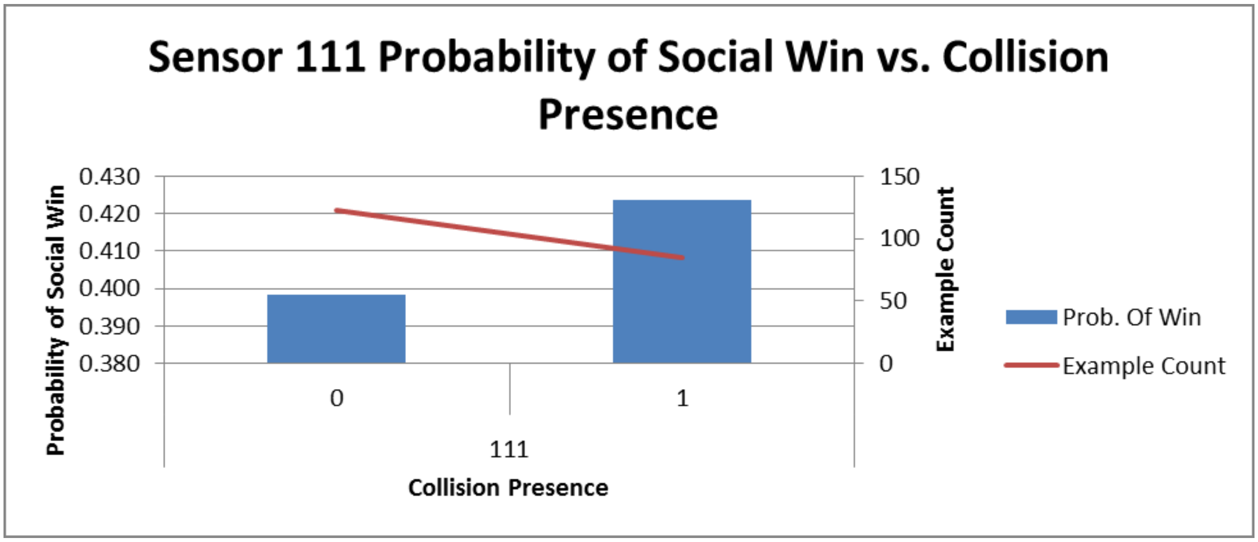


Row Labels	Prob. Of Win	Example Count
0	0.560	25
1	0.380	163
2	0.450	20

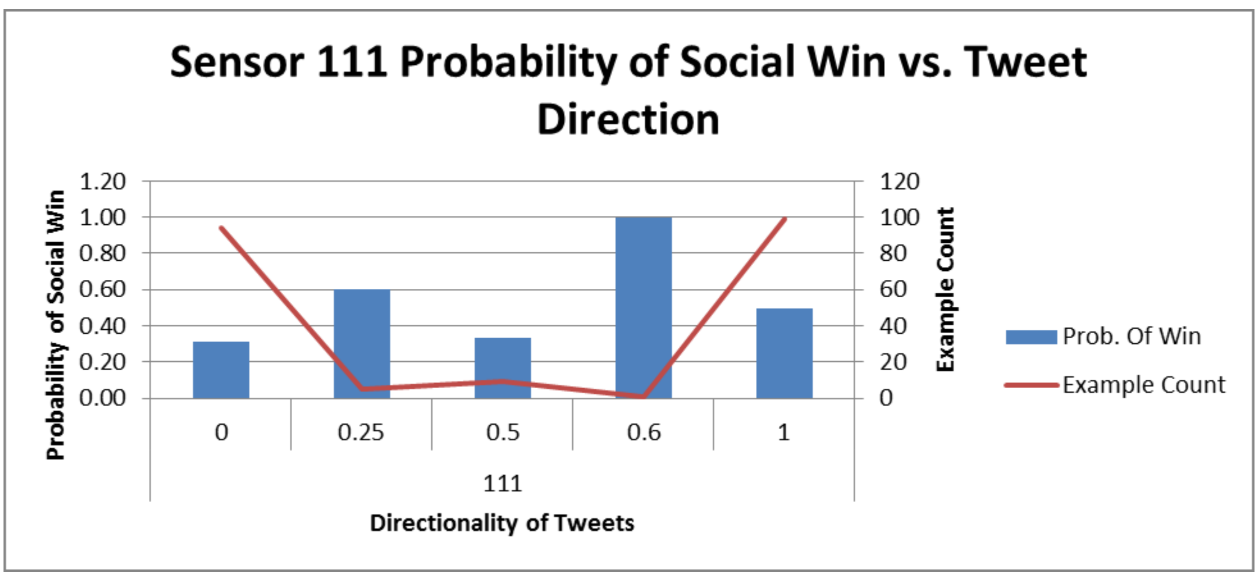


Row Labels	Prob. Of Win	Example Count

0	0.435	85
1	0.390	123

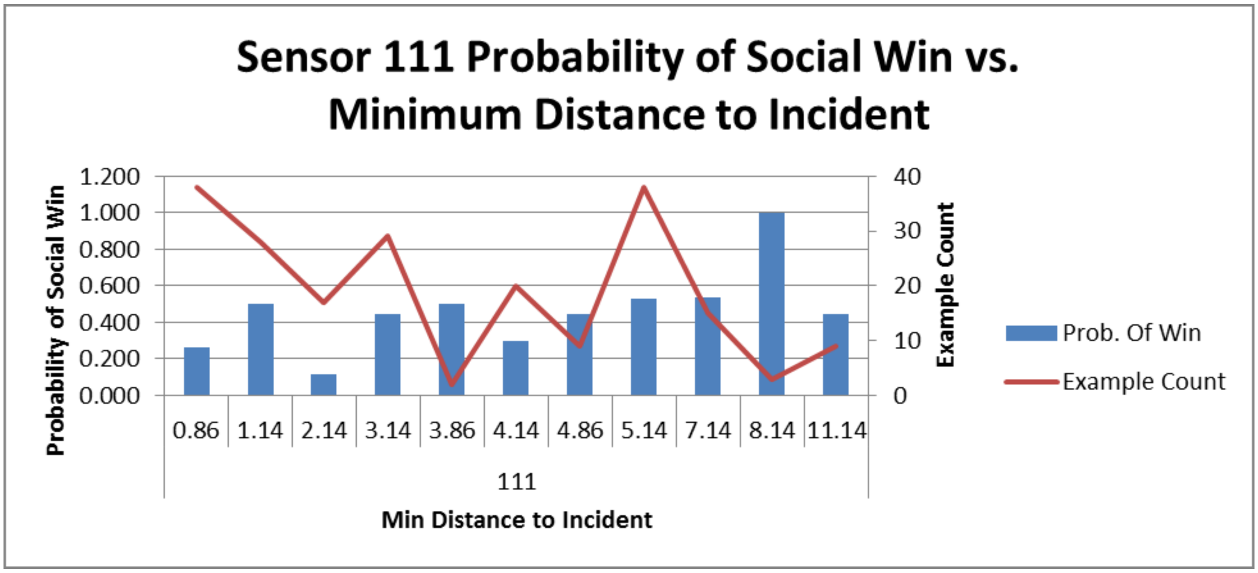


Row Labels	Prob. Of Win	Example Count
0	0.398	123
1	0.424	85

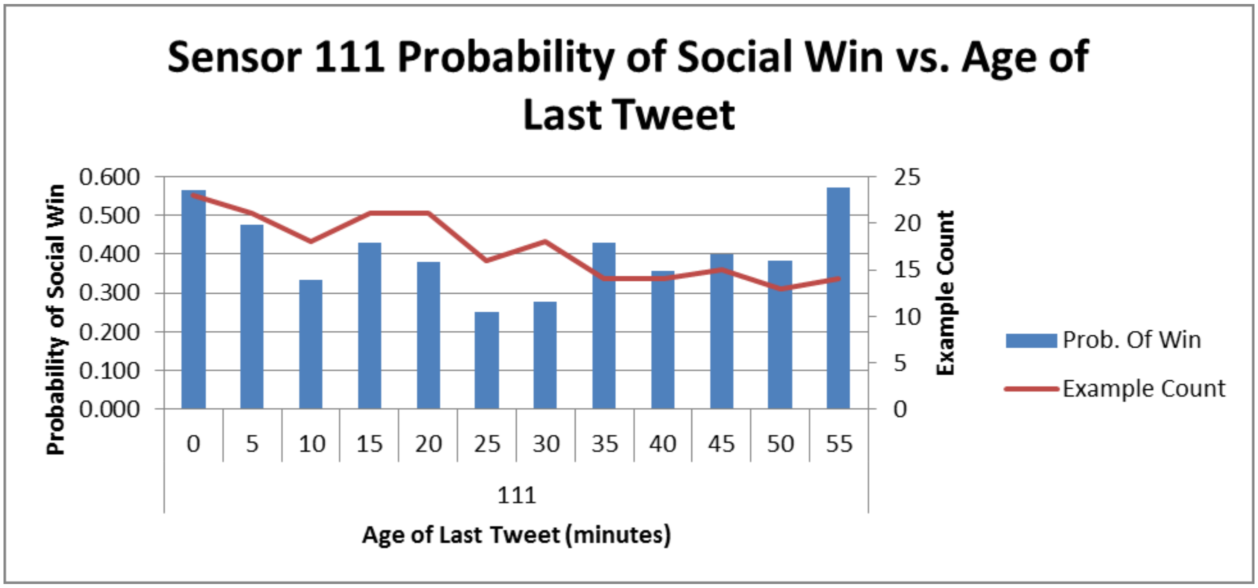


Row Labels	Prob. Of Win	Example Count
0	0.31	94
0.25	0.60	5
0.5	0.33	9

0.6	1.00	1
1	0.49	99

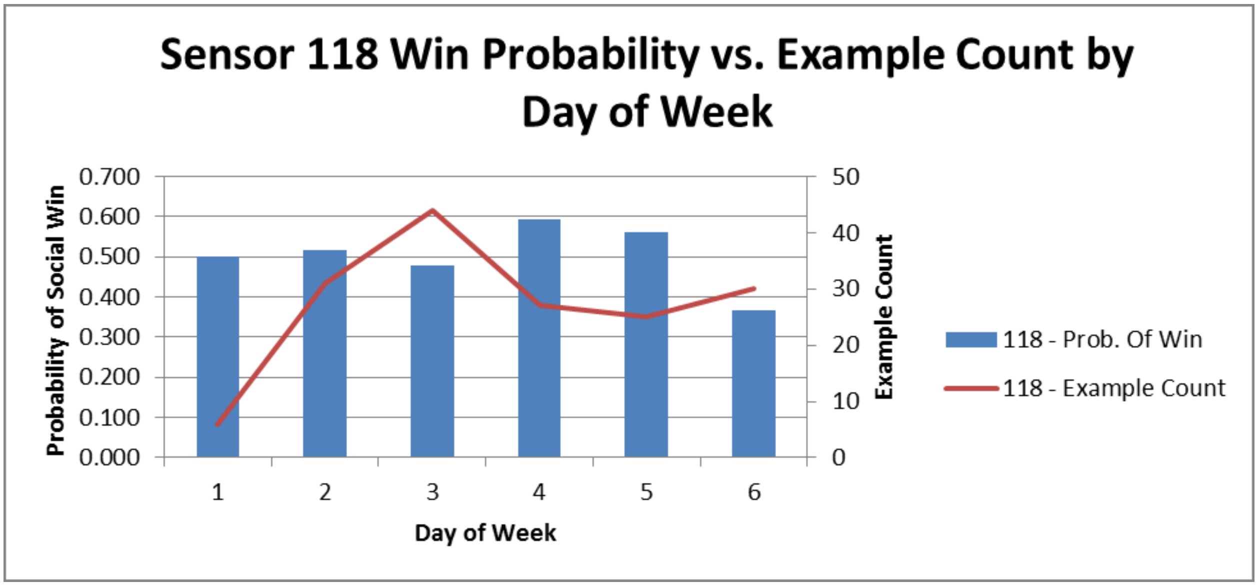


Row Labels	Prob. Of Win	Example Count
0.86	0.263	38
1.14	0.500	28
2.14	0.118	17
3.14	0.448	29
3.86	0.500	2
4.14	0.300	20
4.86	0.444	9
5.14	0.526	38
7.14	0.533	15
8.14	1.000	3
11.14	0.444	9

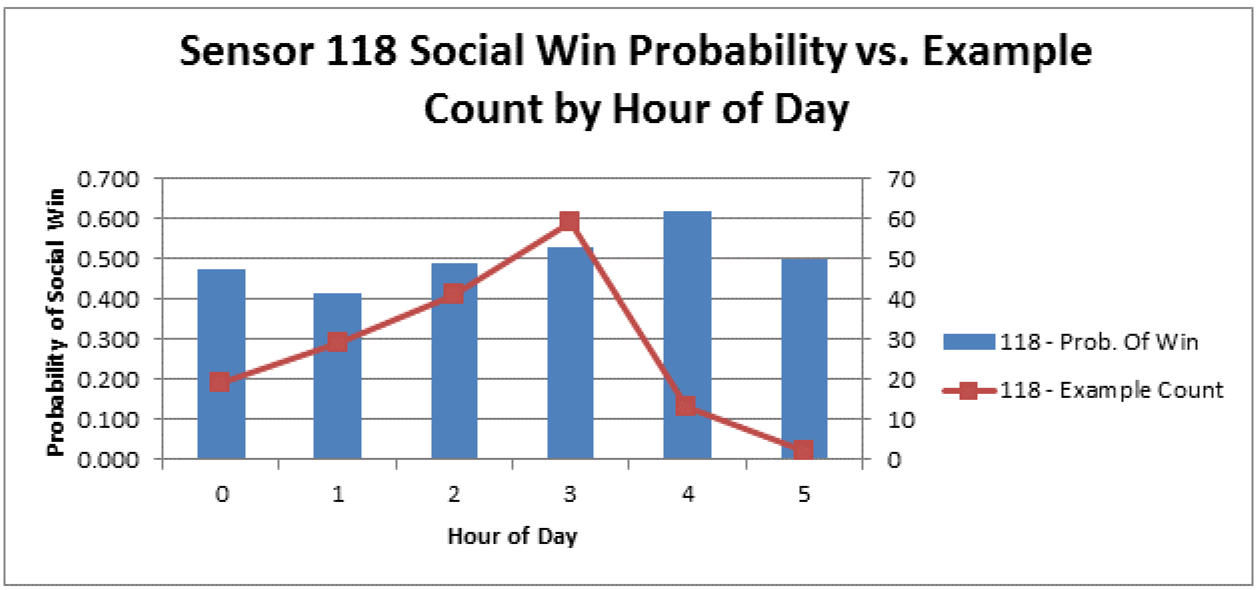


Row Labels	Prob. Of Win	Example Count
0	0.565	23
5	0.476	21
10	0.333	18
15	0.429	21
20	0.381	21
25	0.250	16
30	0.278	18
35	0.429	14
40	0.357	14
45	0.400	15
50	0.385	13
55	0.571	14

Sensor 118

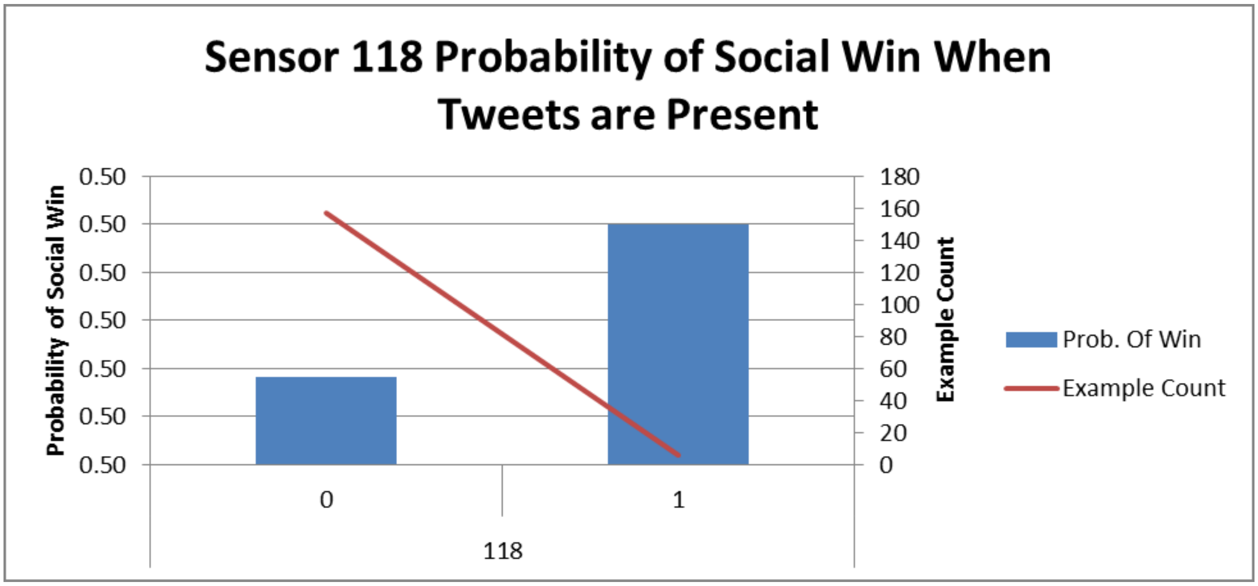


Row Labels	Prob. Of Win	Example Count
1	0.500	6
2	0.516	31
3	0.477	44
4	0.593	27
5	0.560	25
6	0.367	30

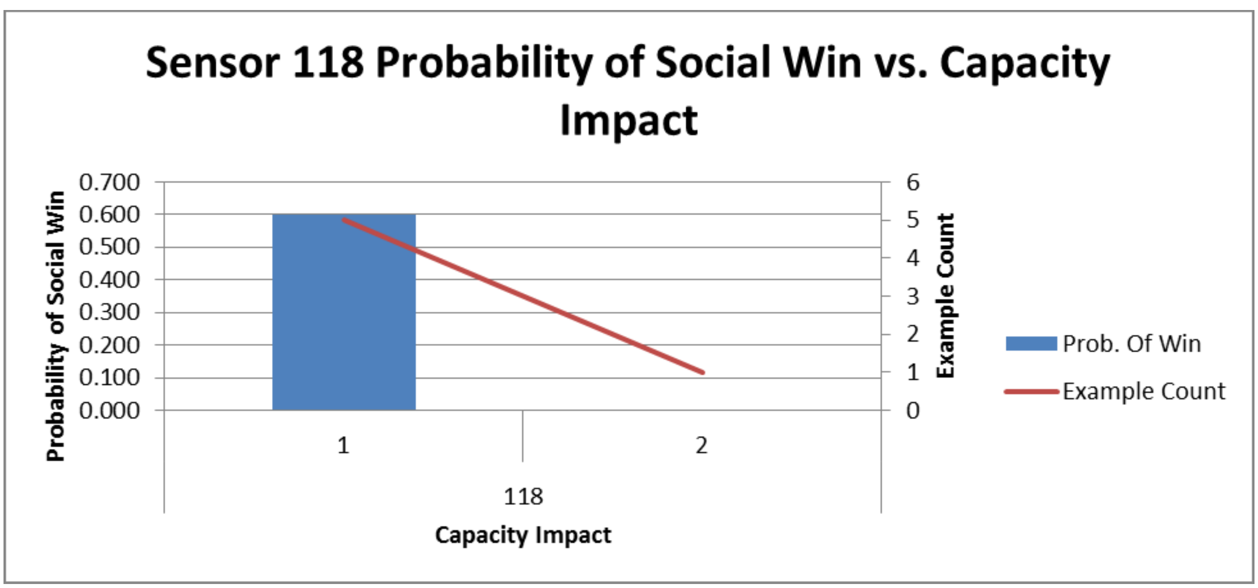


Row Labels	Prob. Of Win	Example Count

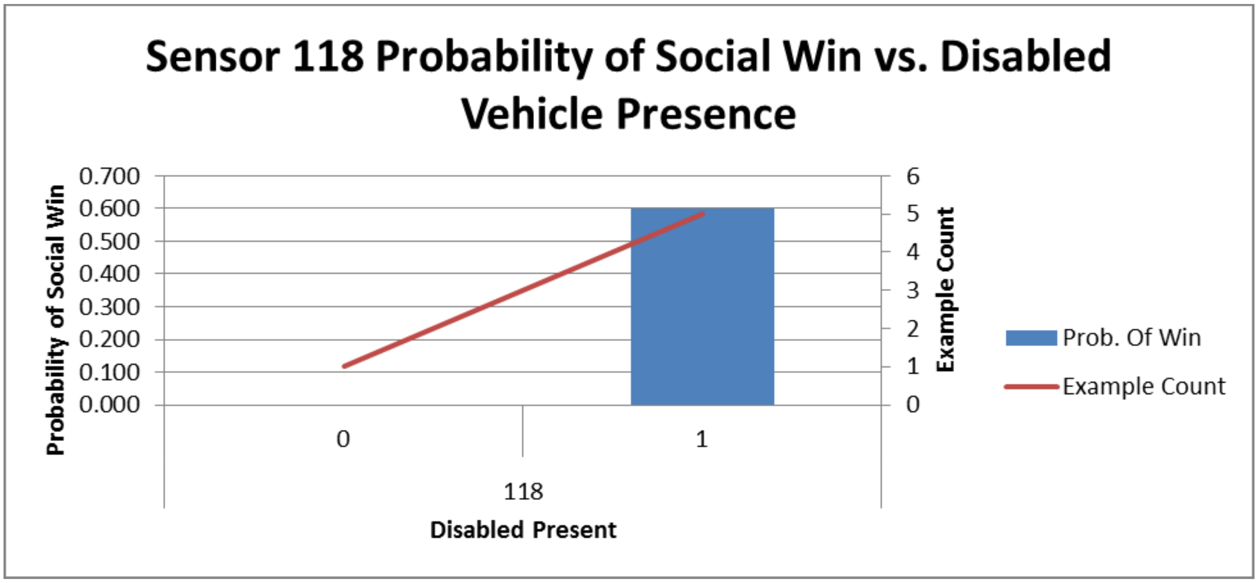
0	0.474	19
1	0.414	29
2	0.488	41
3	0.525	59
4	0.615	13
5	0.500	2



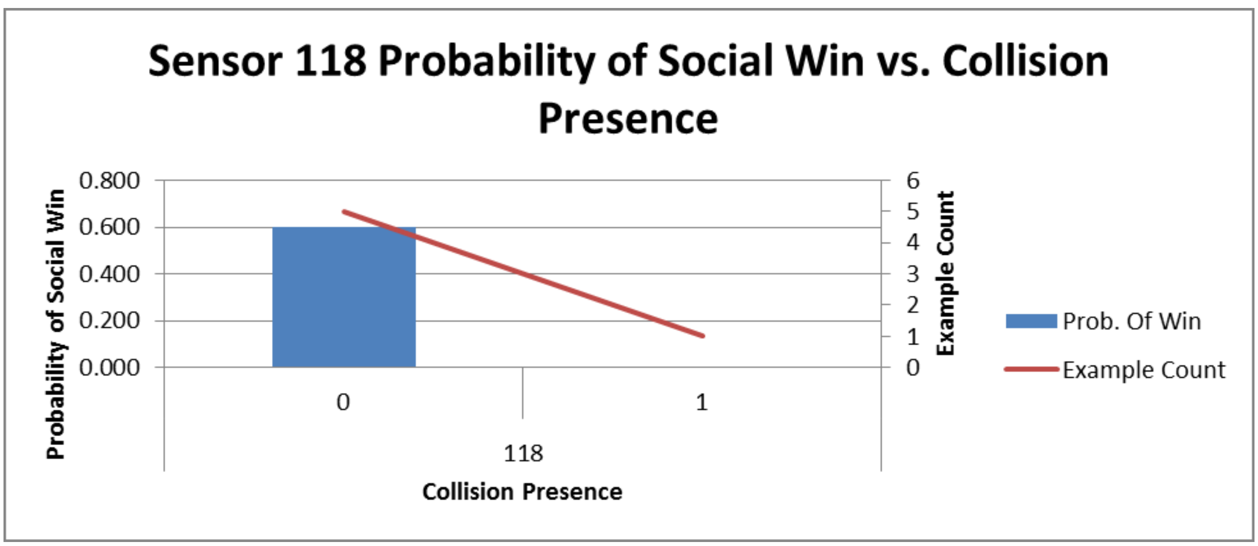
Row Labels	Prob. Of Win	Example Count
0	0.50	157
1	0.50	6



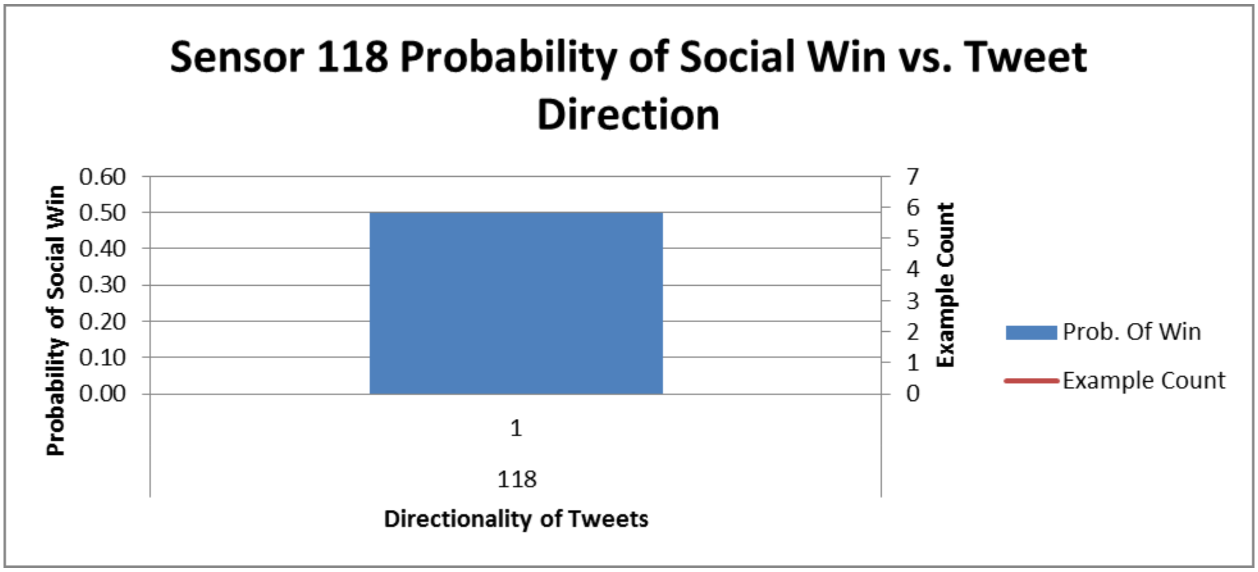
Row Labels	Prob. Of Win	Example Count
1	0.600	5
2	0.000	1



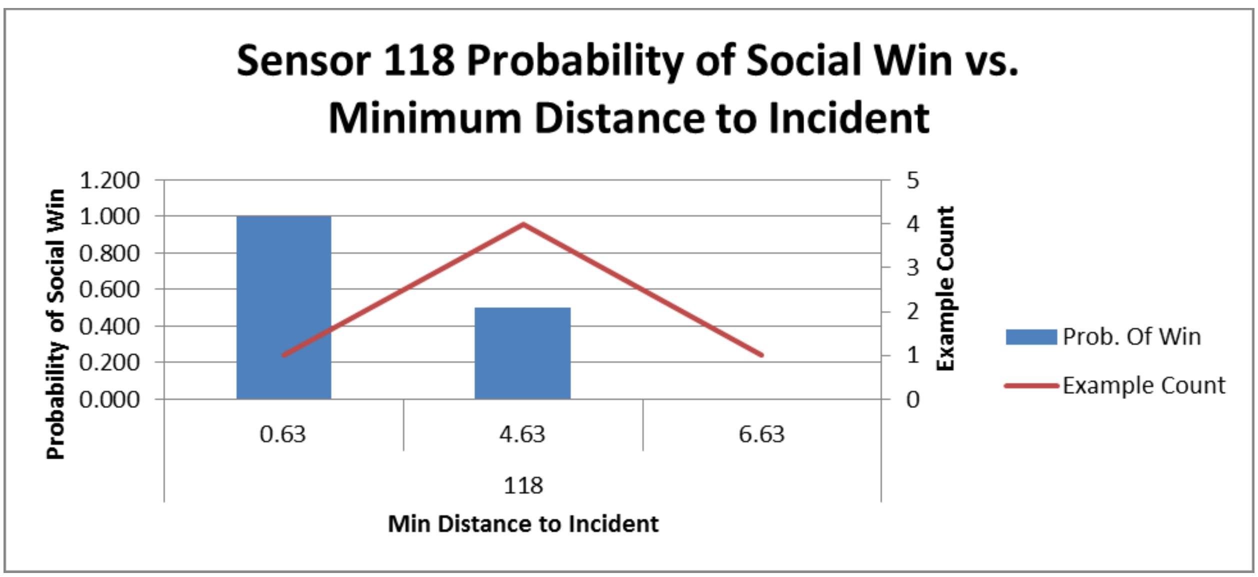
Row Labels	Prob. Of Win	Example Count
0	0.000	1
1	0.600	5



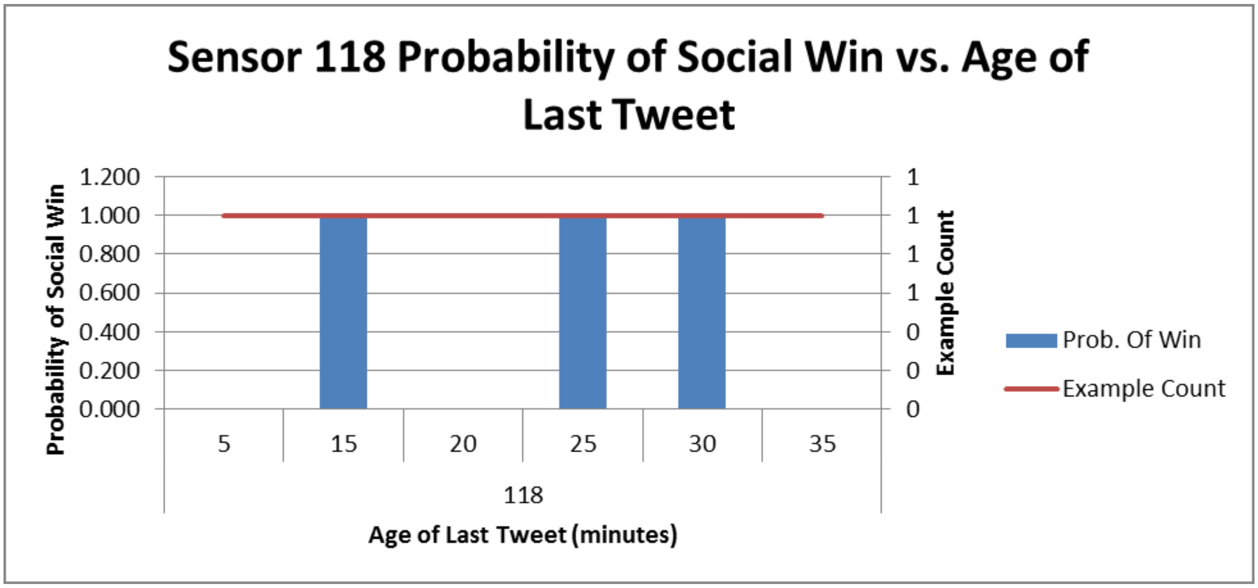
Row Labels	Prob. Of Win	Example Count
0	0.600	5
1	0.000	1



Row Labels	Prob. Of Win	Example Count
1	0.50	6

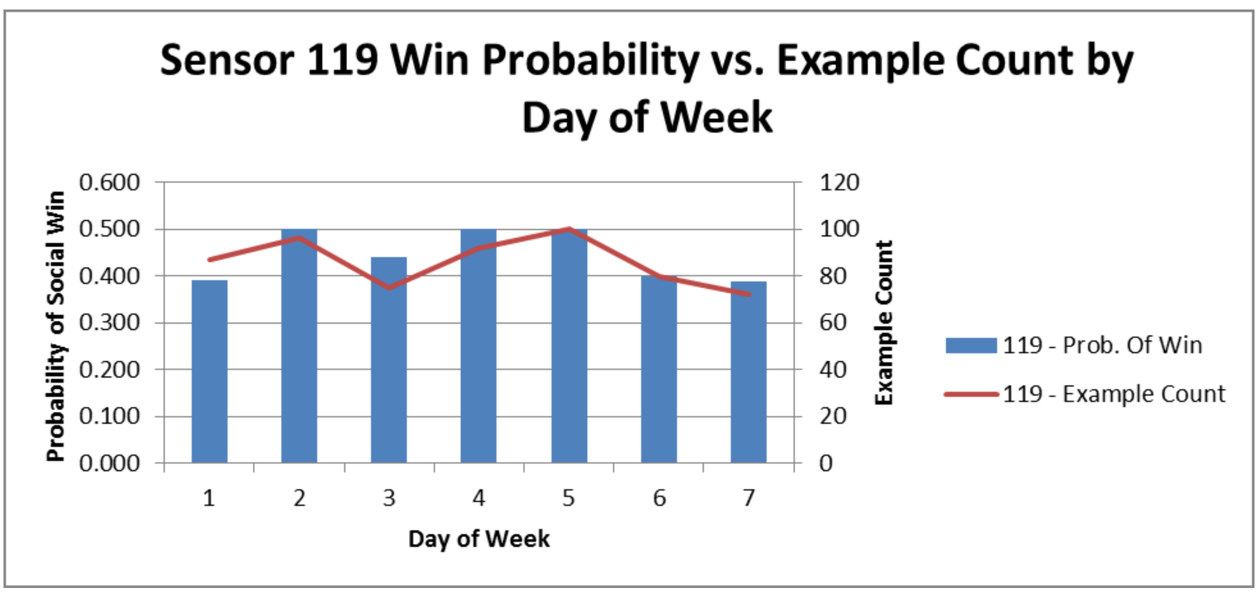


Row Labels	Prob. Of Win	Example Count
0.63	1.000	1
4.63	0.500	4
6.63	0.000	1

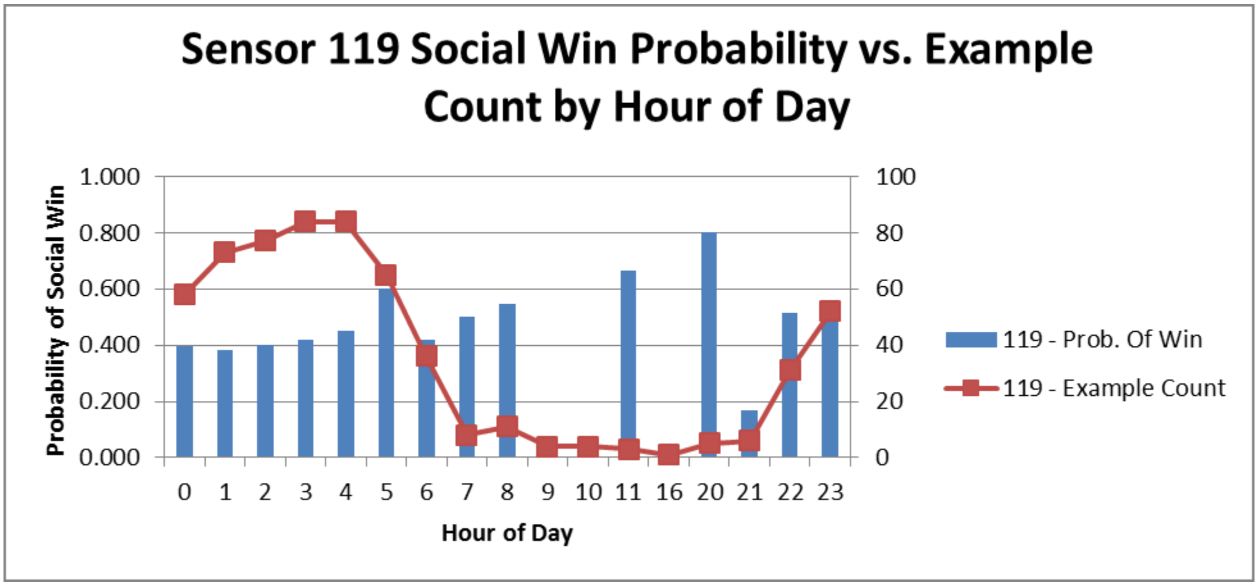


Row Labels	Prob. Of Win	Example Count
5	0.000	1
15	1.000	1
20	0.000	1
25	1.000	1
30	1.000	1
35	0.000	1

Sensor 119

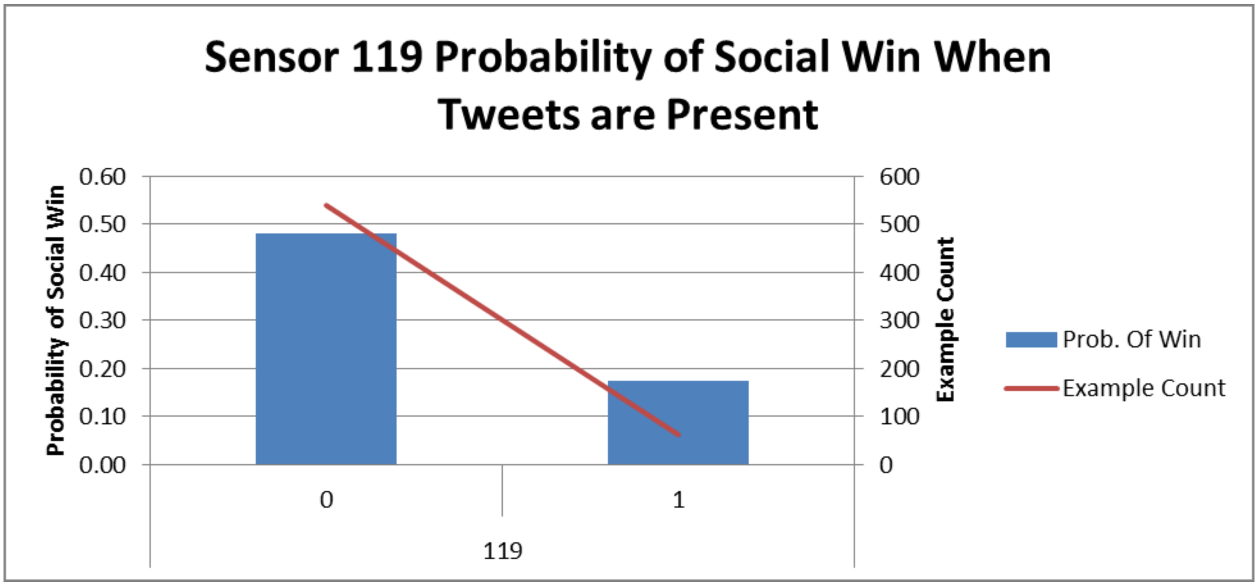


Row Labels	Prob. Of Win	Example Count
1	0.391	87
2	0.500	96
3	0.440	75
4	0.500	92
5	0.500	100
6	0.400	80
7	0.389	72

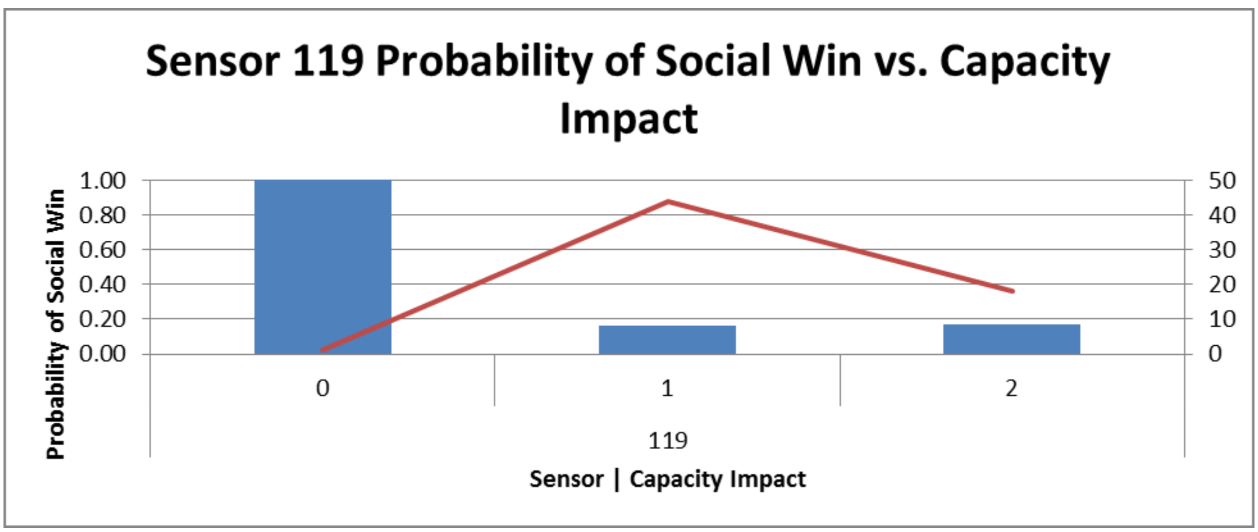


Row Labels	Prob. Of Win	Example Count
0	0.397	58
1	0.384	73
2	0.403	77
3	0.417	84
4	0.452	84
5	0.600	65
6	0.417	36
7	0.500	8
8	0.545	11
9	0.000	4
10	0.000	4
11	0.667	3
16	0.000	1
20	0.800	5
21	0.167	6
22	0.516	31

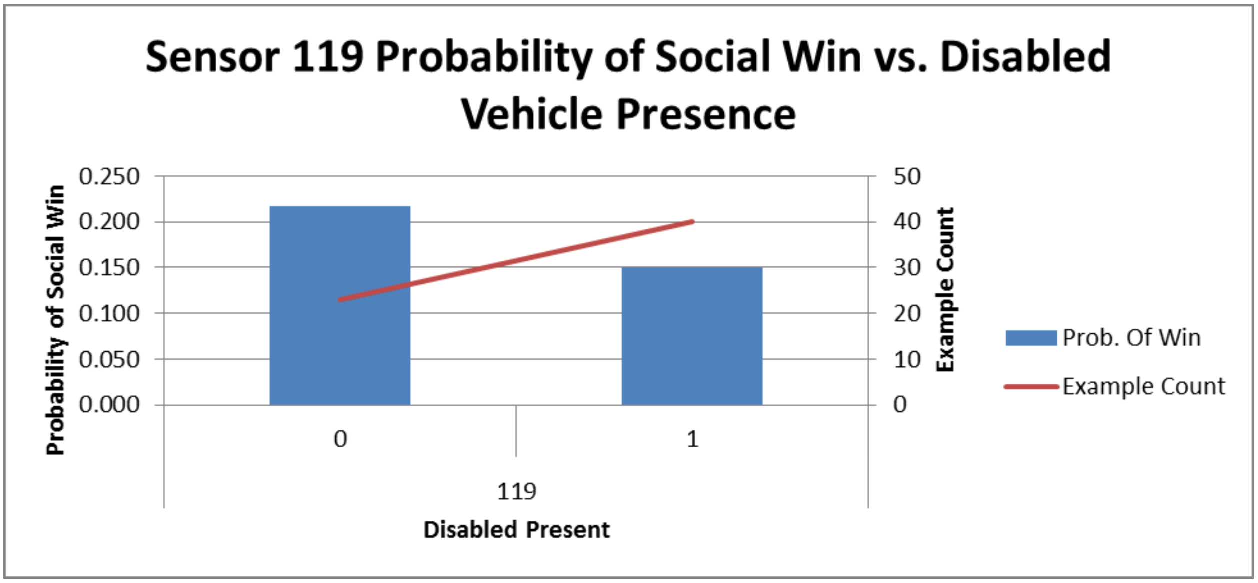
23	0.558	52



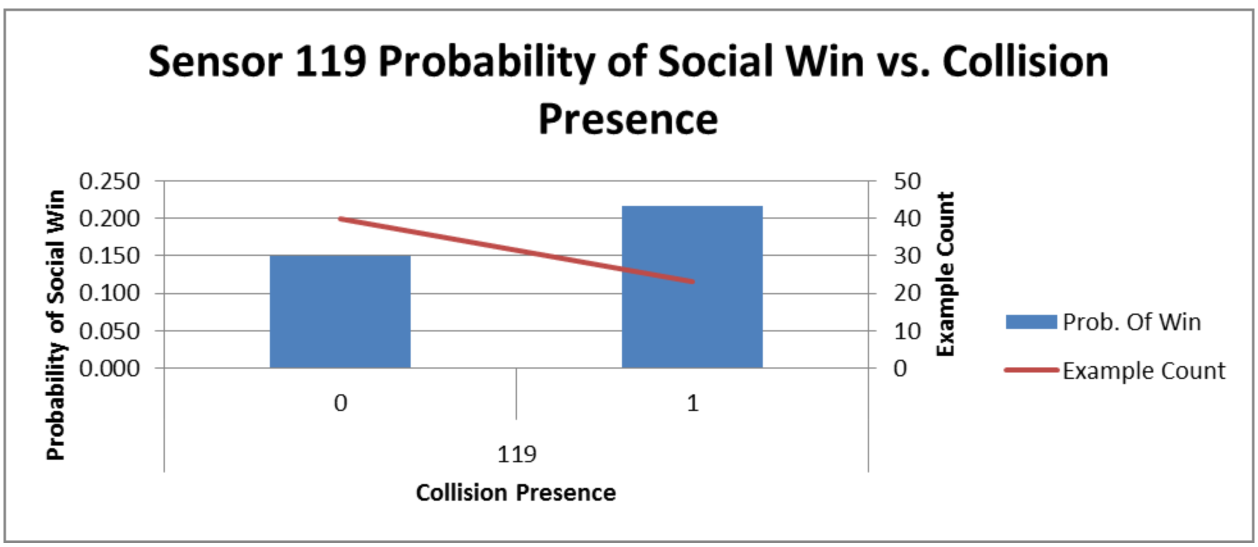
Row Labels	Prob. Of Win	Example Count
0	0.48	539
1	0.17	63



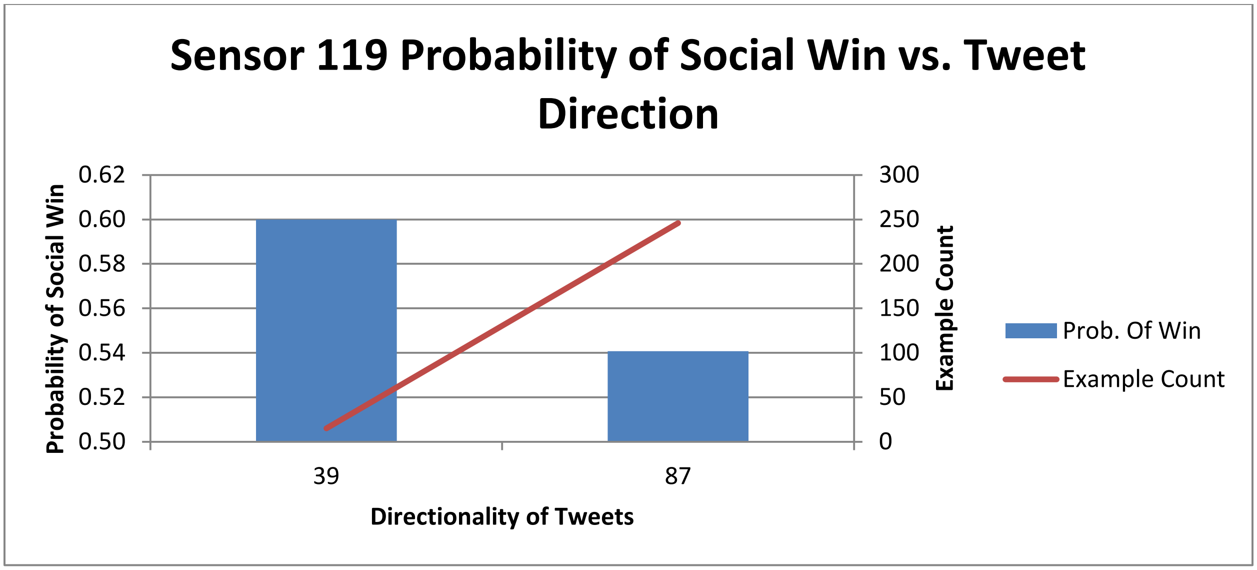
Row Labels	Prob. Of Win	Example Count
0	1.000	1
1	0.159	44
2	0.167	18



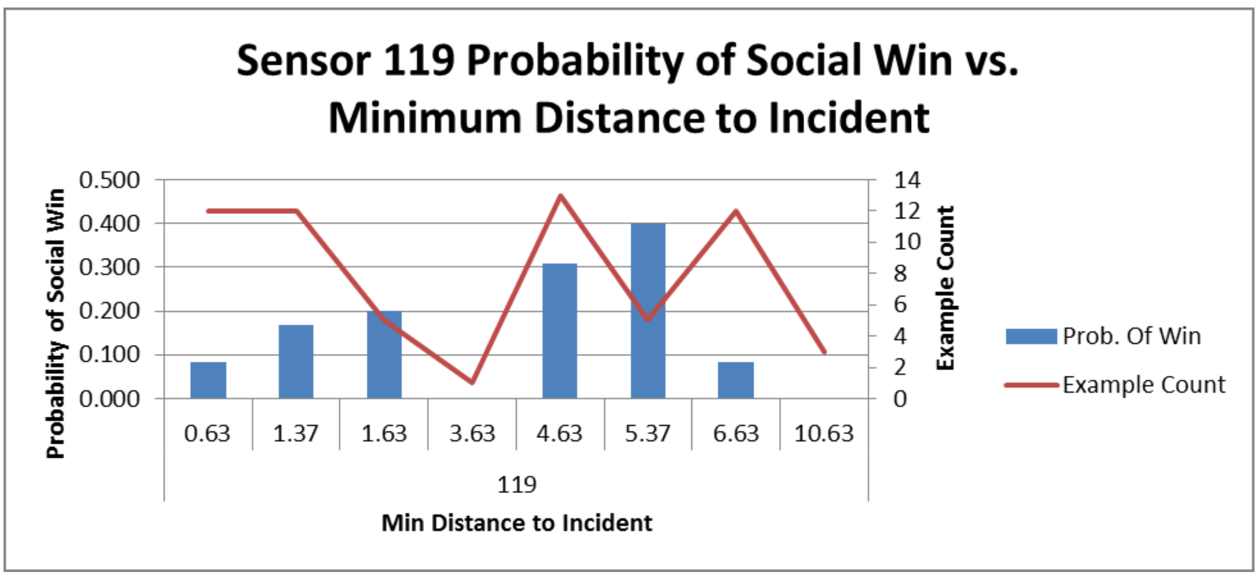
Row Labels	Prob. Of Win	Example Count
0	0.217	23
1	0.150	40



Row Labels	Prob. Of Win	Example Count
0	0.150	40
1	0.217	23

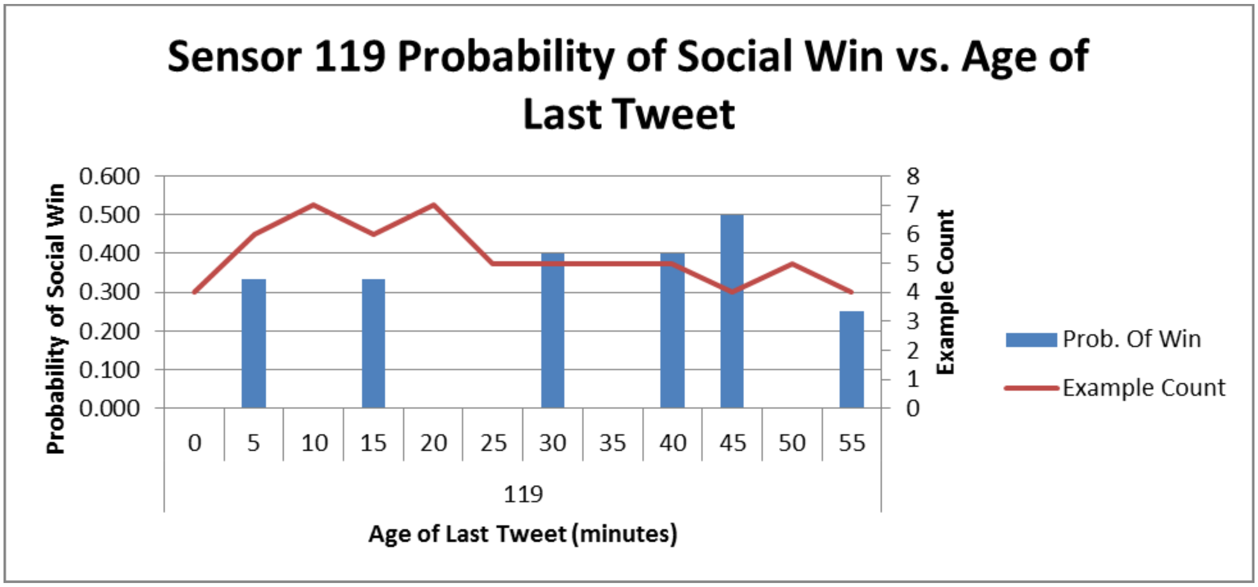


Row Labels	Prob. Of Win	Example Count
0	0.00	5
0.5	0.25	4
1	0.19	54



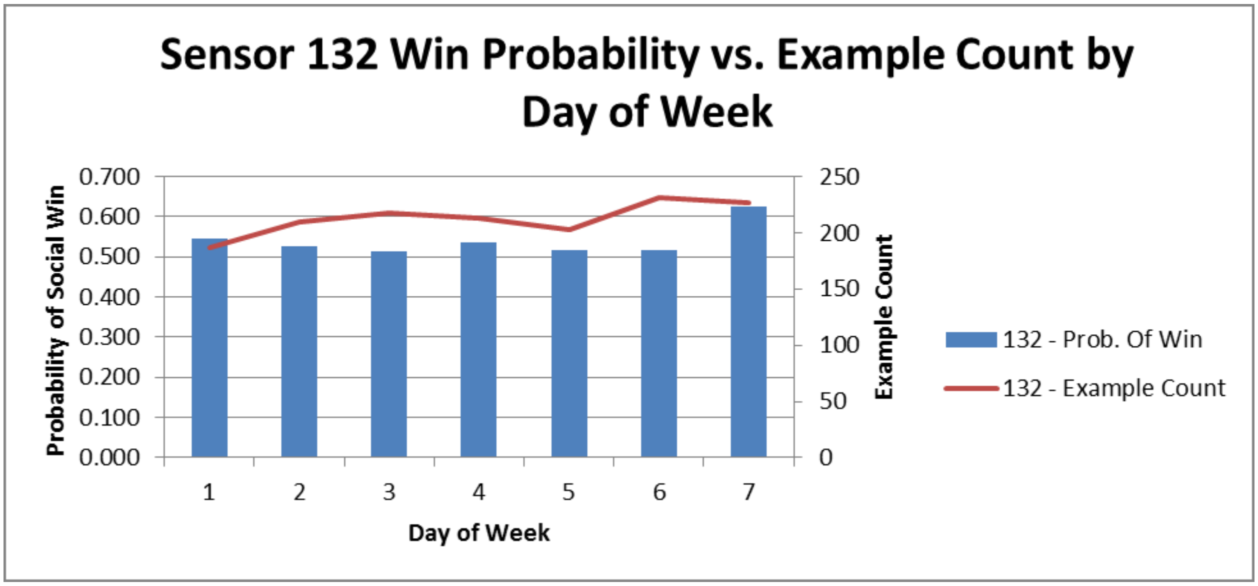
Row Labels	Prob. Of Win	Example Count
0.63	0.083	12
1.37	0.167	12
1.63	0.200	5
3.63	0.000	1
4.63	0.308	13

5.37	0.400	5
6.63	0.083	12
10.63	0.000	3

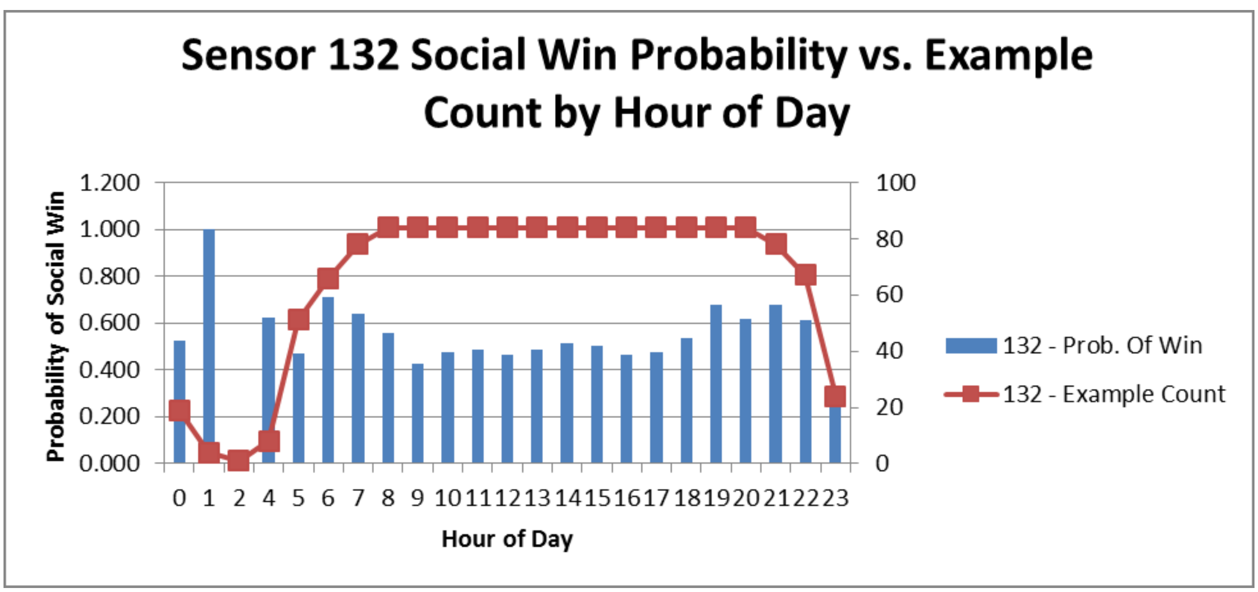


Row Labels	Prob. Of Win	Example Count
0	0.000	4
5	0.333	6
10	0.000	7
15	0.333	6
20	0.000	7
25	0.000	5
30	0.400	5
35	0.000	5
40	0.400	5
45	0.500	4
50	0.000	5
55	0.250	4

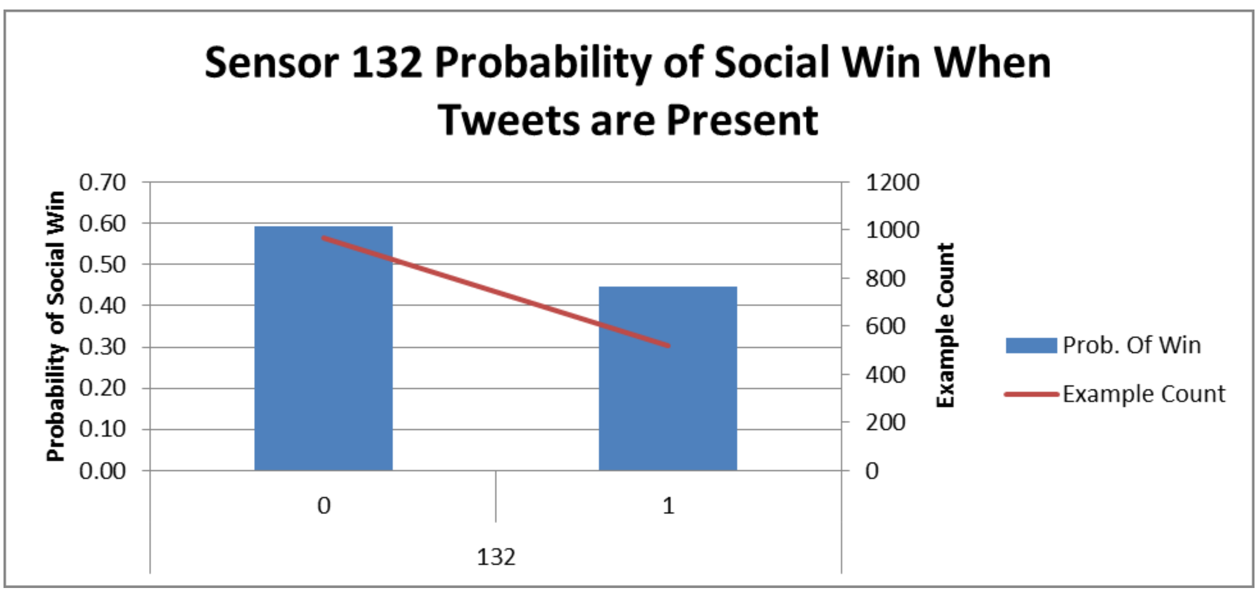
Sensor 132



Row Labels	Prob. Of Win	Example Count
1	0.545	187
2	0.526	209
3	0.514	218
4	0.535	213
5	0.517	203
6	0.515	231
7	0.626	227

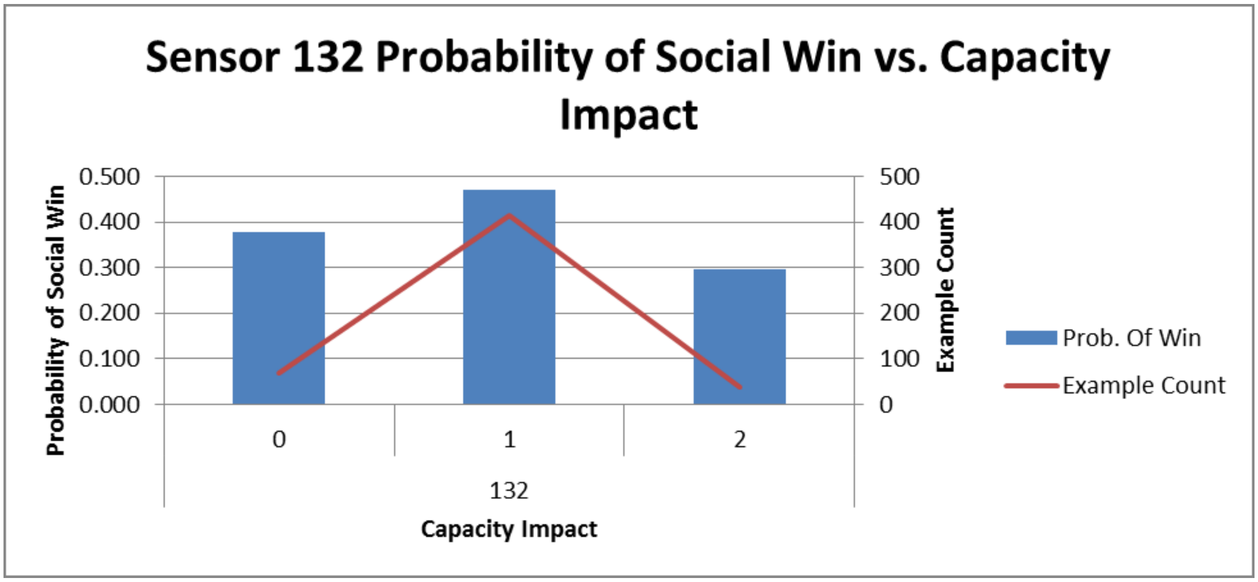


Row Labels	Prob. Of Win	Example Count
0	0.526	19
1	1.000	4
2	0.000	1
4	0.625	8
5	0.471	51
6	0.712	66
7	0.641	78
8	0.560	84
9	0.429	84
10	0.476	84
11	0.488	84
12	0.464	84
13	0.488	84
14	0.512	84
15	0.500	84
16	0.464	84
17	0.476	84
18	0.536	84
19	0.679	84
20	0.619	84
21	0.679	78
22	0.612	67
23	0.333	24

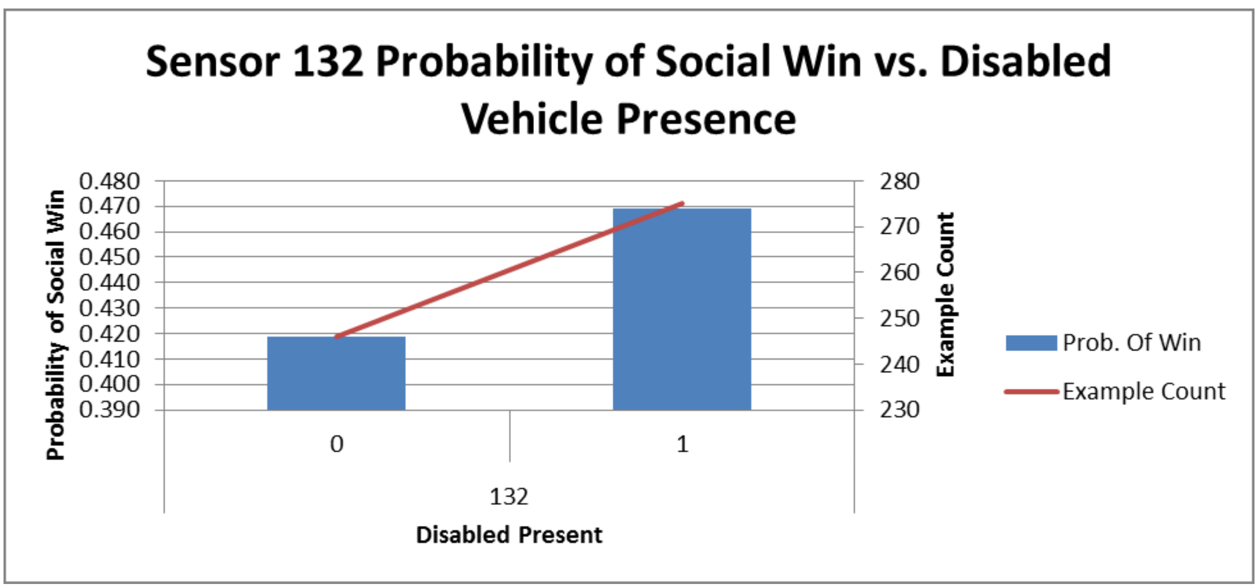


Row Labels	Prob. Of Win	Example Count

0	0.59	967
1	0.45	521

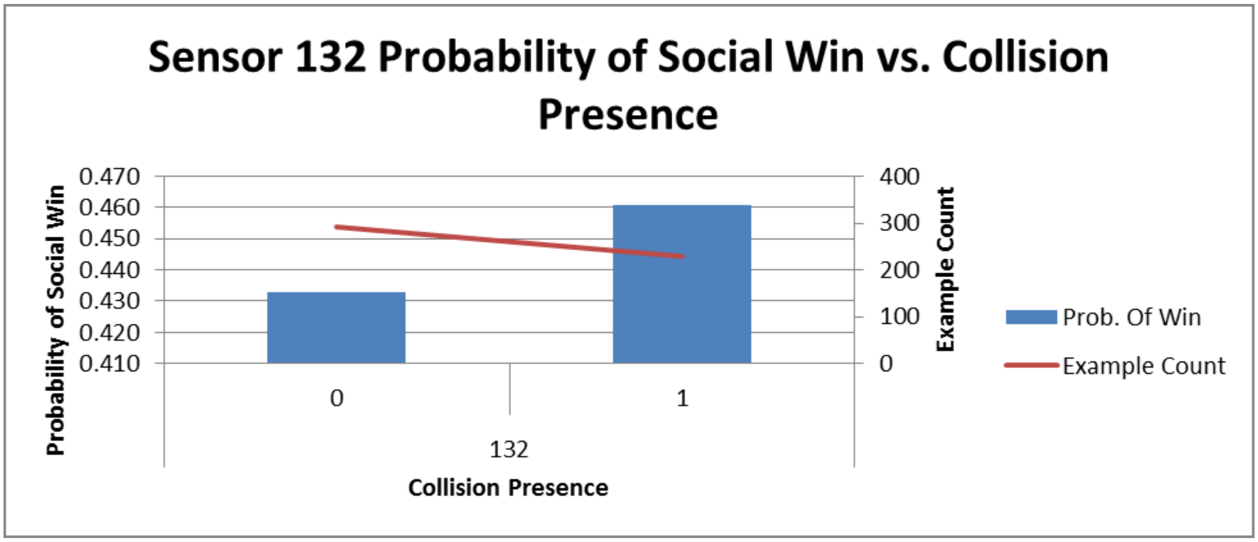


Row Labels	Prob. Of Win	Example Count
0	0.377	69
1	0.470	415
2	0.297	37

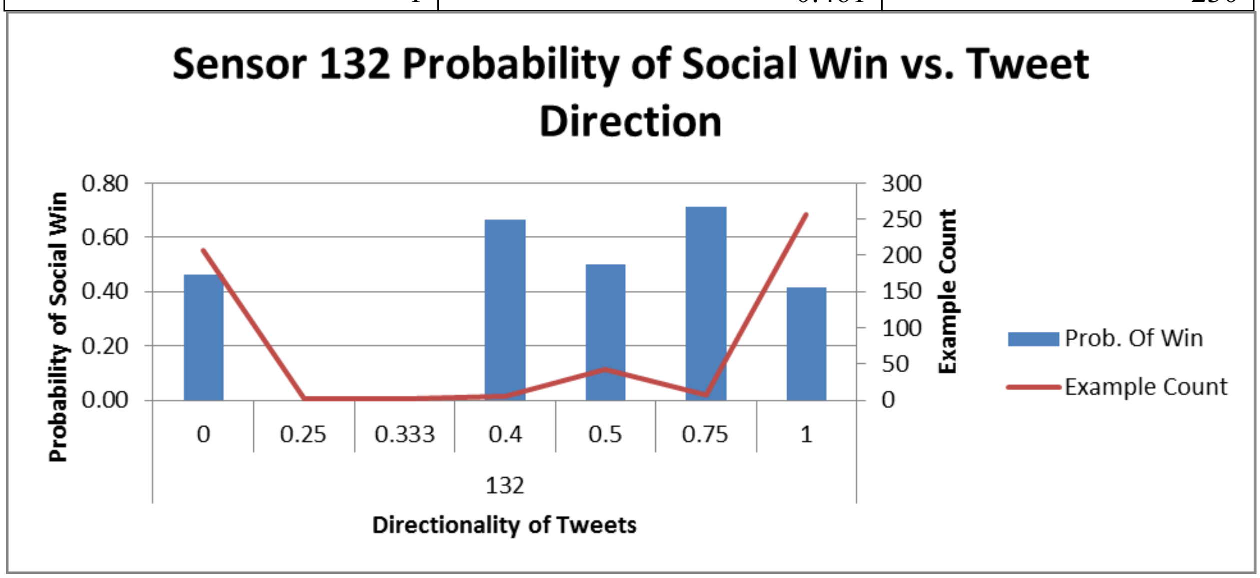


Row Labels	Prob. Of Win	Example Count
0	0.419	246

1	0.469	275

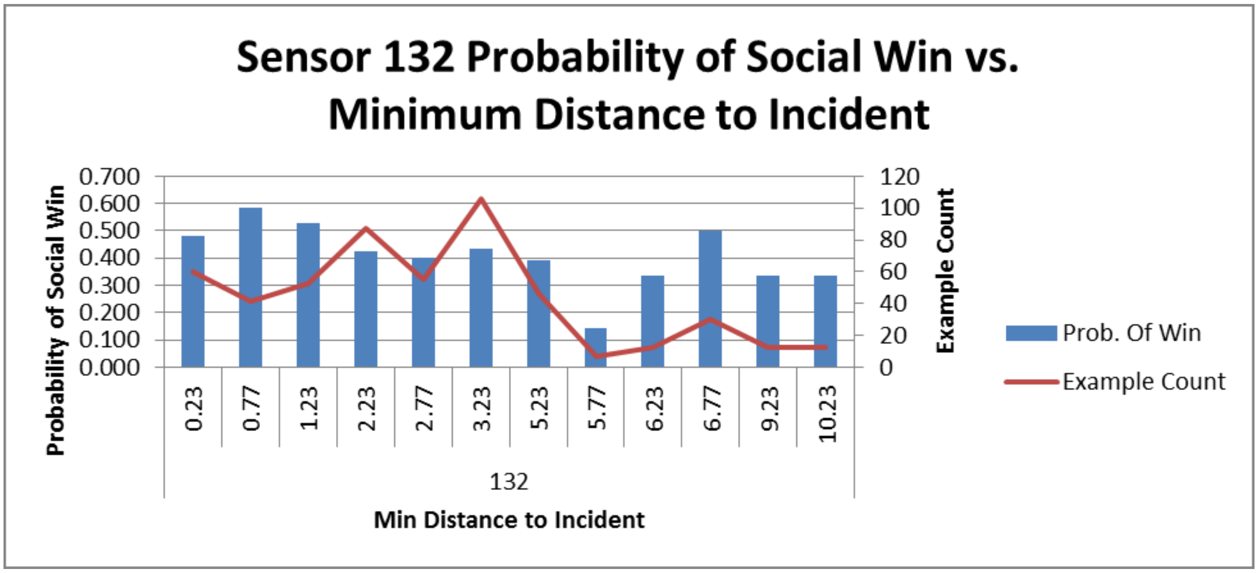


Row Labels	Prob. Of Win	Example Count
0	0.433	291
1	0.461	230

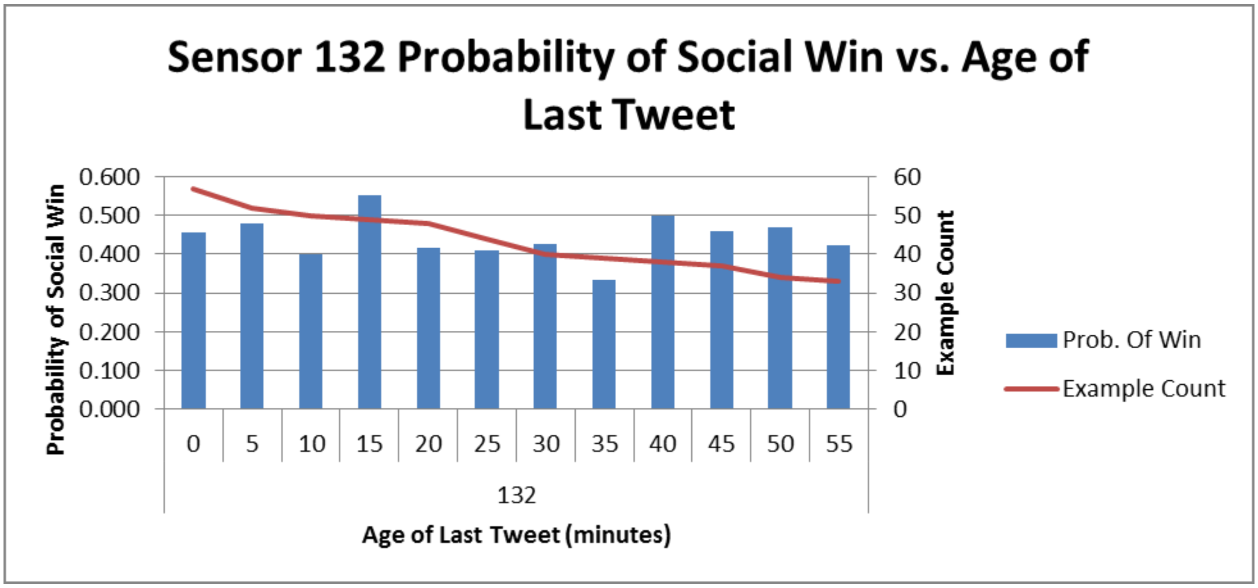


Row Labels	Prob. Of Win	Example Count
0	0.46	207
0.25	0.00	1
0.333	0.00	2
0.4	0.67	6
0.5	0.50	42
0.75	0.71	7

1	0.41	256

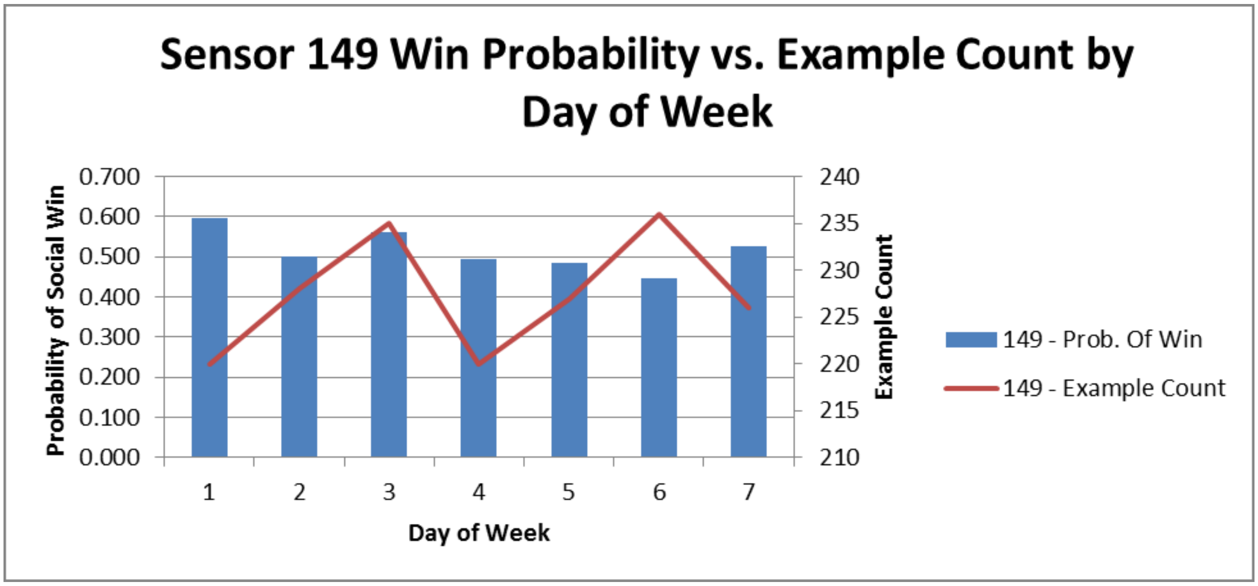


Row Labels	Prob. Of Win	Example Count
0.23	0.483	60
0.77	0.585	41
1.23	0.528	53
2.23	0.425	87
2.77	0.400	55
3.23	0.434	106
5.23	0.391	46
5.77	0.143	7
6.23	0.333	12
6.77	0.500	30
9.23	0.333	12
10.23	0.333	12

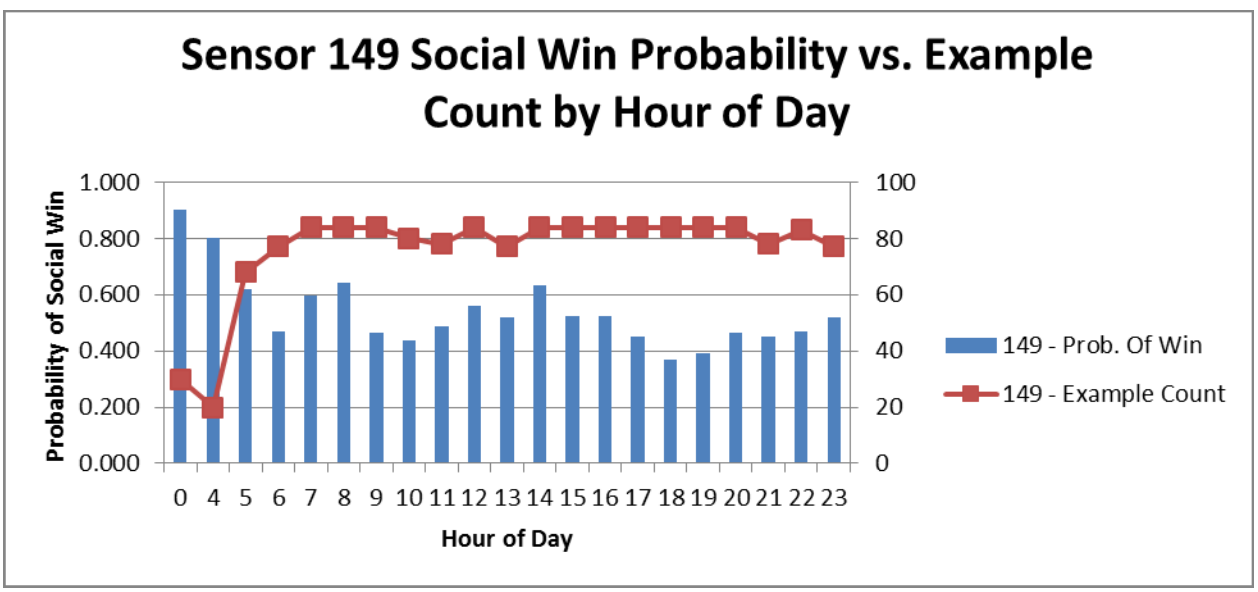


Row Labels	Prob. Of Win	Example Count
0	0.456	57
5	0.481	52
10	0.400	50
15	0.551	49
20	0.417	48
25	0.409	44
30	0.425	40
35	0.333	39
40	0.500	38
45	0.459	37
50	0.471	34
55	0.424	33

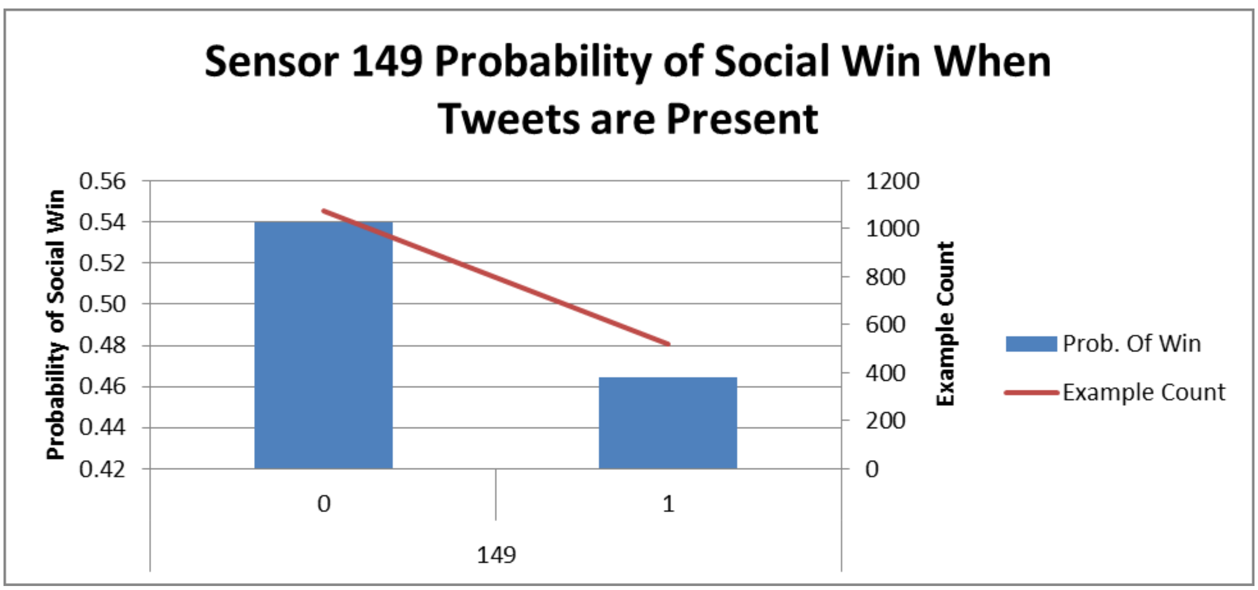
Sensor 149



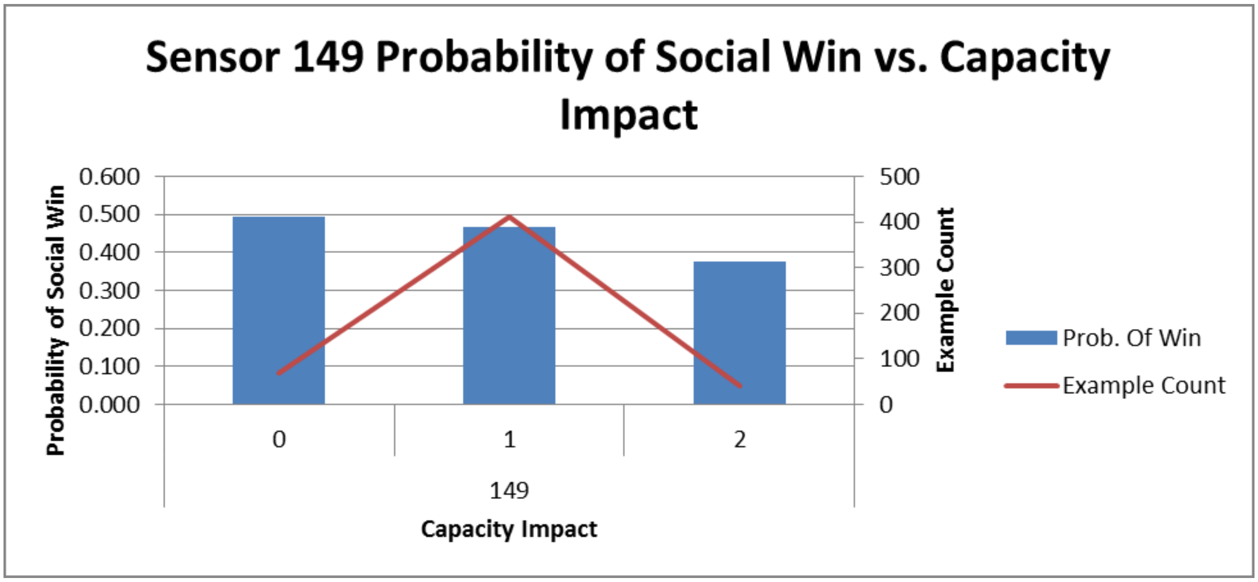
Row Labels	Prob. Of Win	Example Count
1	0.595	220
2	0.500	228
3	0.562	235
4	0.495	220
5	0.485	227
6	0.445	236
7	0.527	226



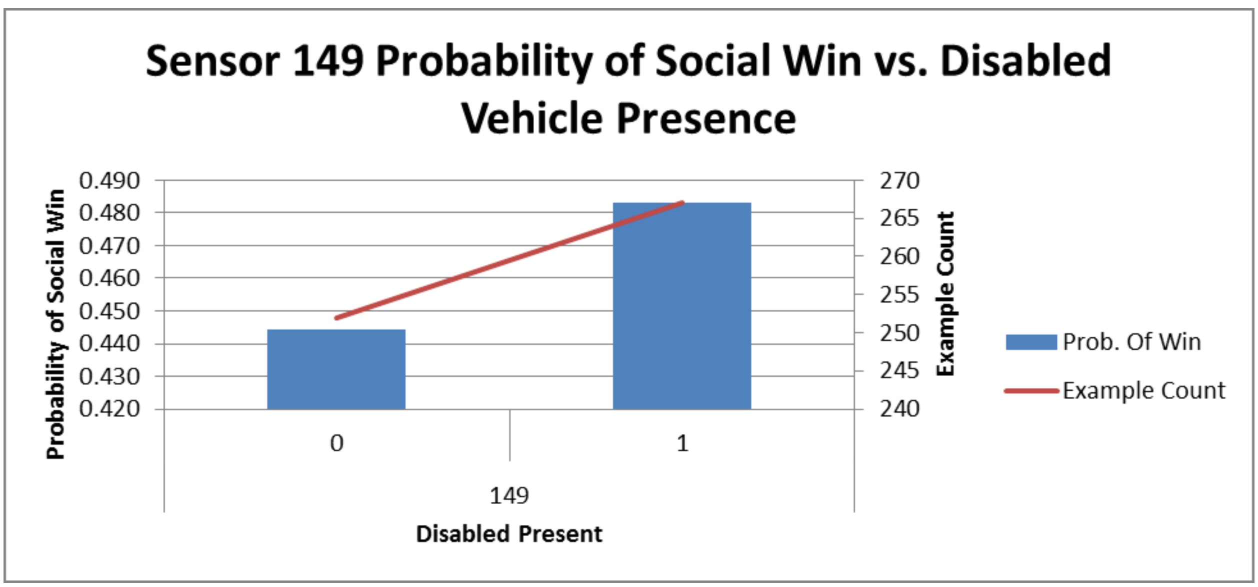
Row Labels	Prob. Of Win	Example Count
0	0.900	30
4	0.800	20
5	0.618	68
6	0.468	77
7	0.595	84
8	0.643	84
9	0.464	84
10	0.438	80
11	0.487	78
12	0.560	84
13	0.519	77
14	0.631	84
15	0.524	84
16	0.524	84
17	0.452	84
18	0.369	84
19	0.393	84
20	0.464	84
21	0.449	78
22	0.470	83
23	0.519	77



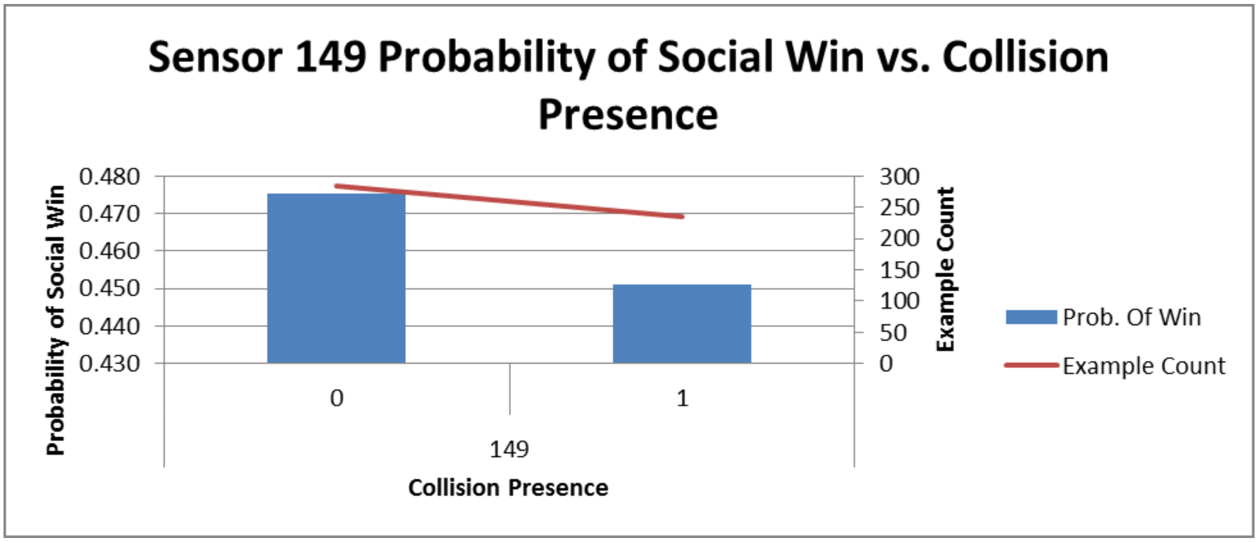
Row Labels	Prob. Of Win	Example Count
0	0.54	1073
1	0.46	519



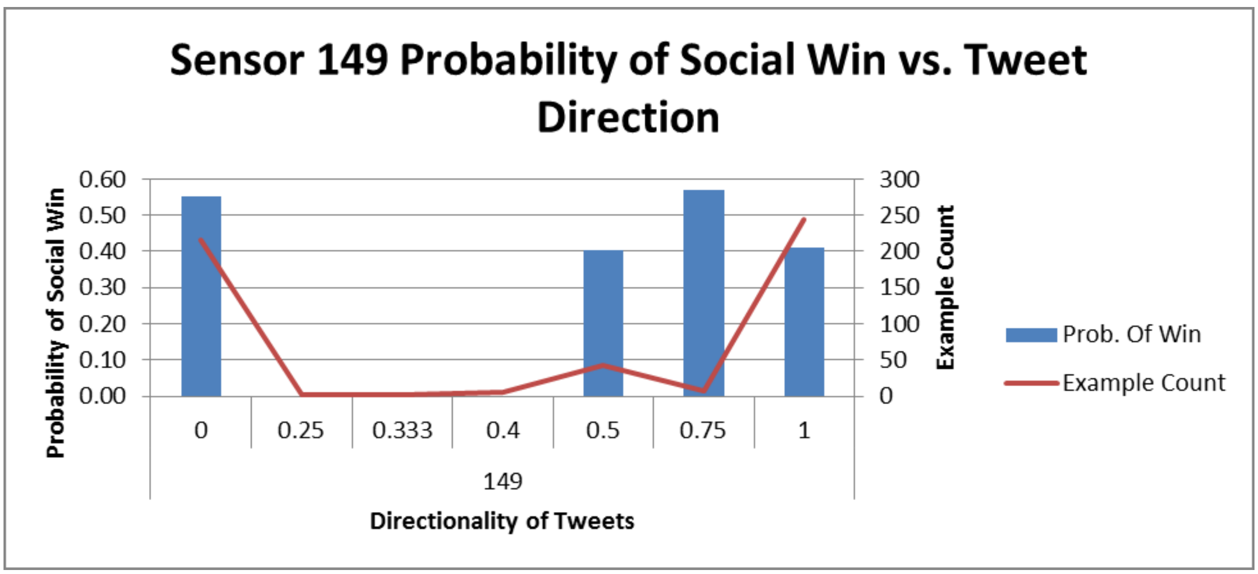
Row Labels	Prob. Of Win	Example Count
0	0.493	69
1	0.468	410
2	0.375	40



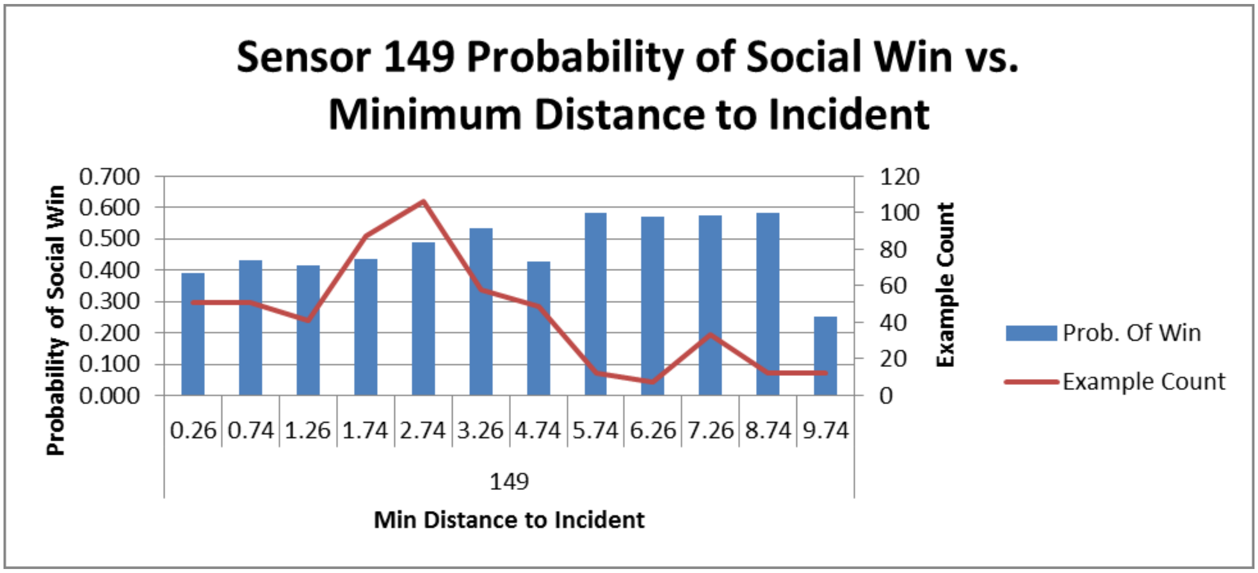
Row Labels	Prob. Of Win	Example Count
0	0.444	252
1	0.483	267



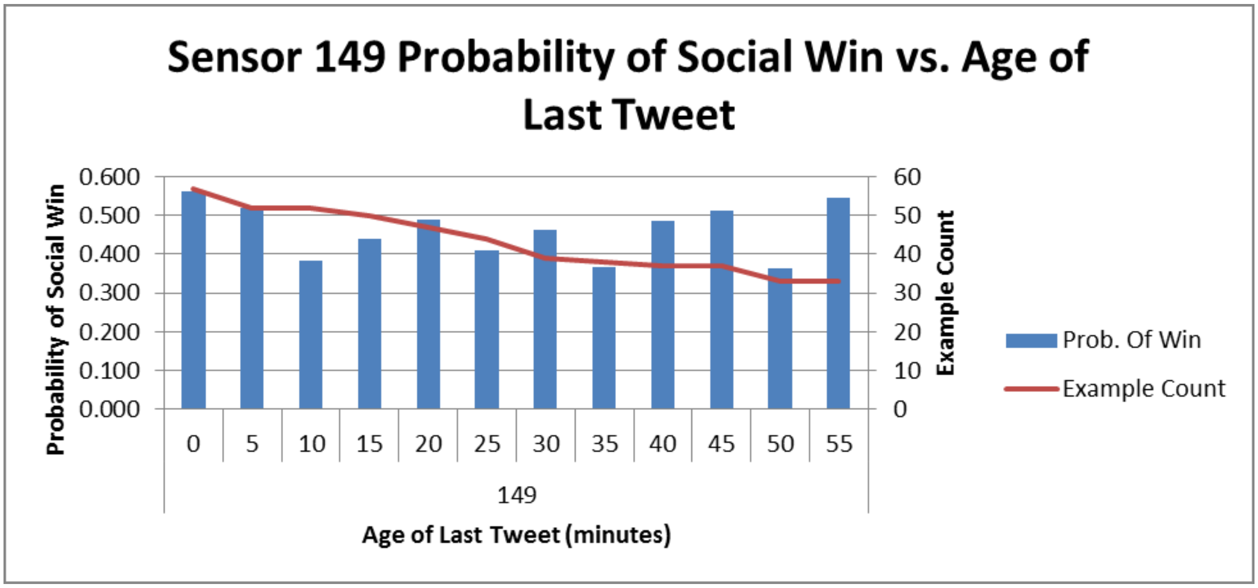
Row Labels	Prob. Of Win	Example Count
0	0.475	284
1	0.451	235



Row Labels	Prob. Of Win	Example Count
0	0.55	216
0.25	0.00	1
0.333	0.00	2
0.4	0.00	6
0.5	0.40	42
0.75	0.57	7
1	0.41	245

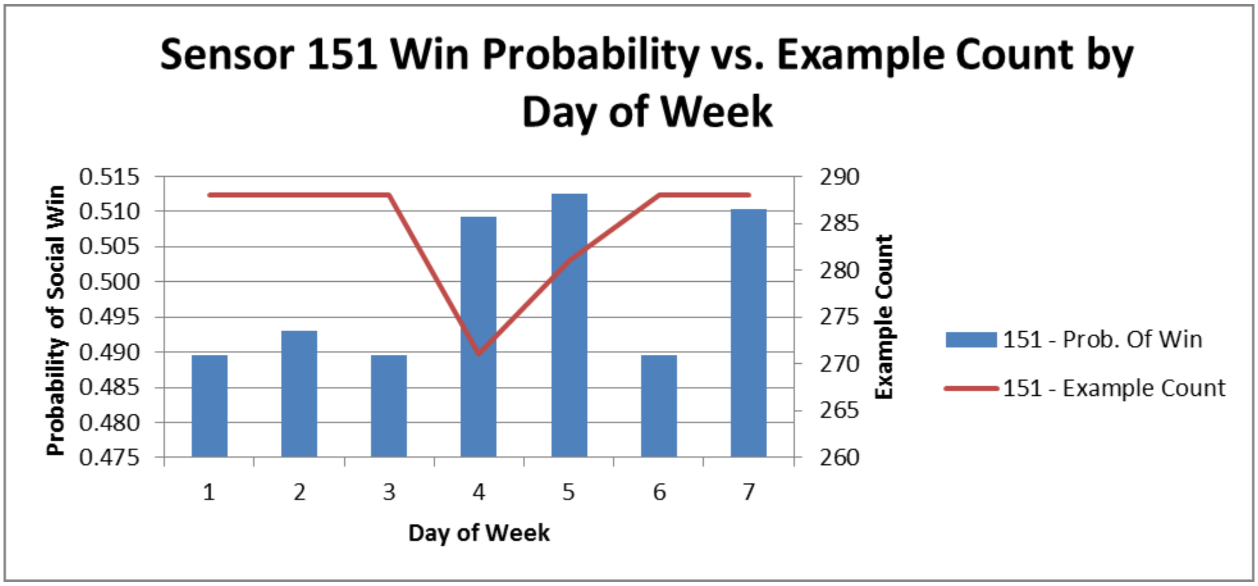


Row Labels	Prob. Of Win	Example Count
0.26	0.392	51
0.74	0.431	51
1.26	0.415	41
1.74	0.437	87
2.74	0.491	106
3.26	0.534	58
4.74	0.429	49
5.74	0.583	12
6.26	0.571	7
7.26	0.576	33
8.74	0.583	12
9.74	0.250	12

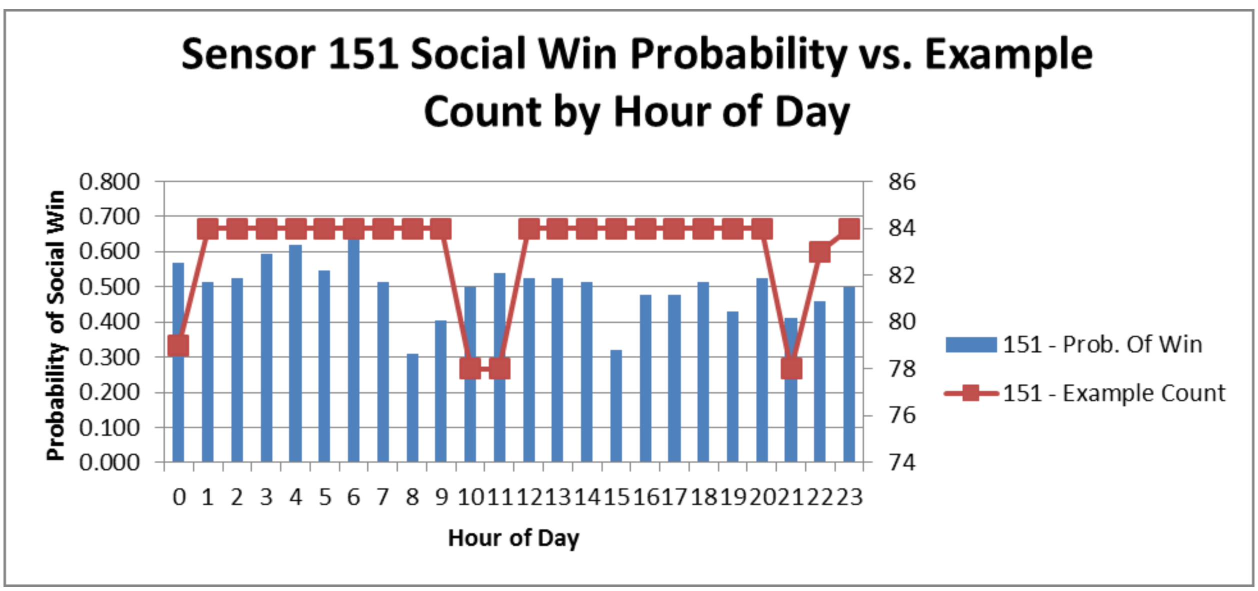


Row Labels	Prob. Of Win	Example Count
0	0.561	57
5	0.519	52
10	0.385	52
15	0.440	50
20	0.489	47
25	0.409	44
30	0.462	39
35	0.368	38
40	0.486	37
45	0.514	37
50	0.364	33
55	0.545	33

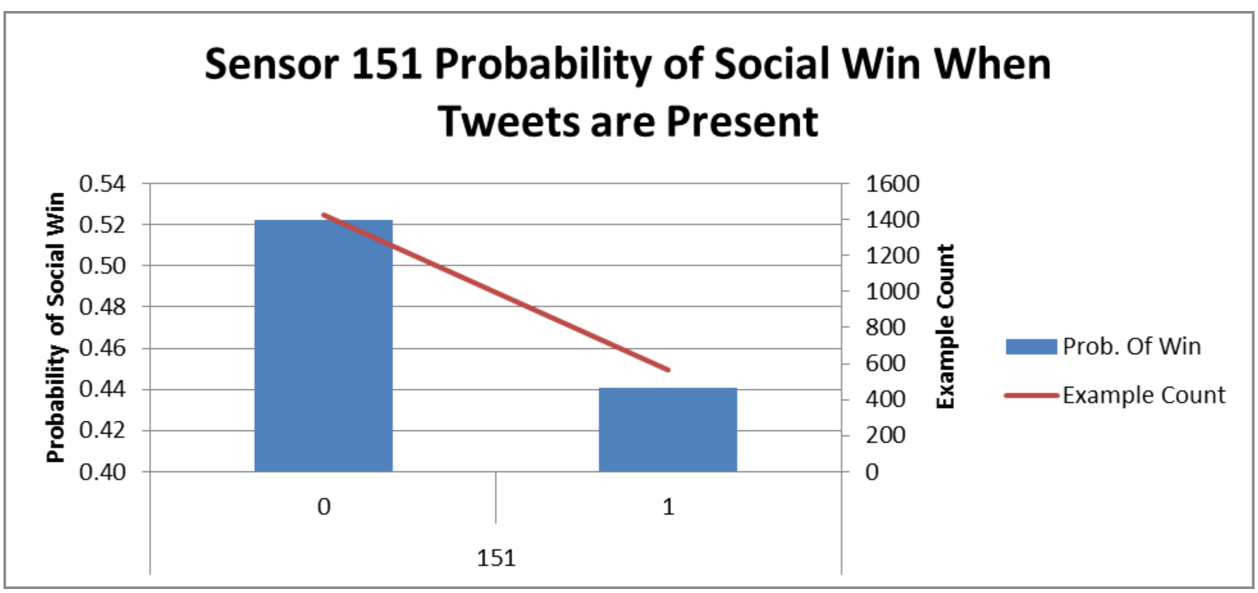
Sensor 151



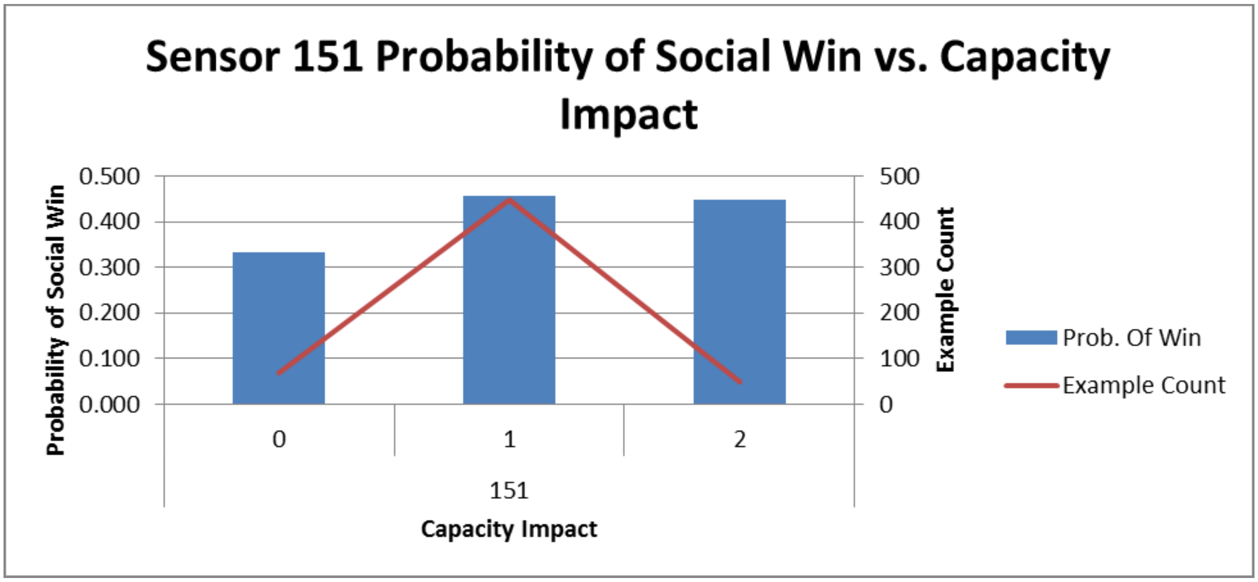
Row Labels	Prob. Of Win	Example Count
1	0.490	288
2	0.493	288
3	0.490	288
4	0.509	271
5	0.512	281
6	0.490	288
7	0.510	288



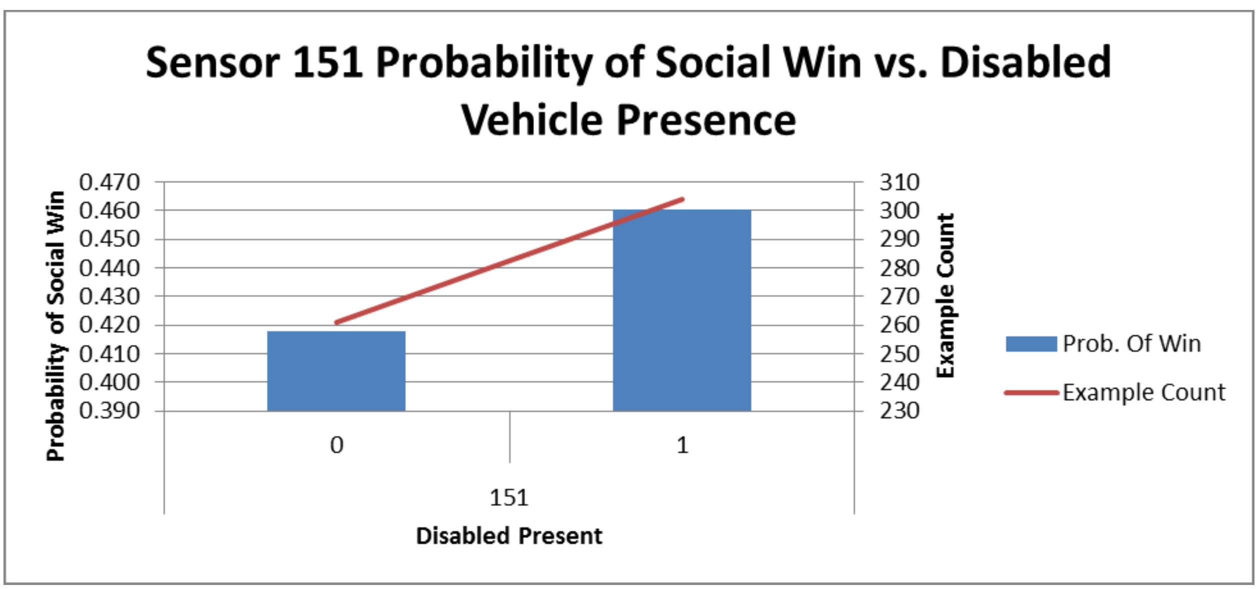
Row Labels	Prob. Of Win	Example Count
0	0.570	79
1	0.512	84
2	0.524	84
3	0.595	84
4	0.619	84
5	0.548	84
6	0.679	84
7	0.512	84
8	0.310	84
9	0.405	84
10	0.500	78
11	0.538	78
12	0.524	84
13	0.524	84
14	0.512	84
15	0.321	84
16	0.476	84
17	0.476	84
18	0.512	84
19	0.429	84
20	0.524	84
21	0.410	78
22	0.458	83
23	0.500	84



Row Labels	Prob. Of Win	Example Count
0	0.52	1427
1	0.44	565

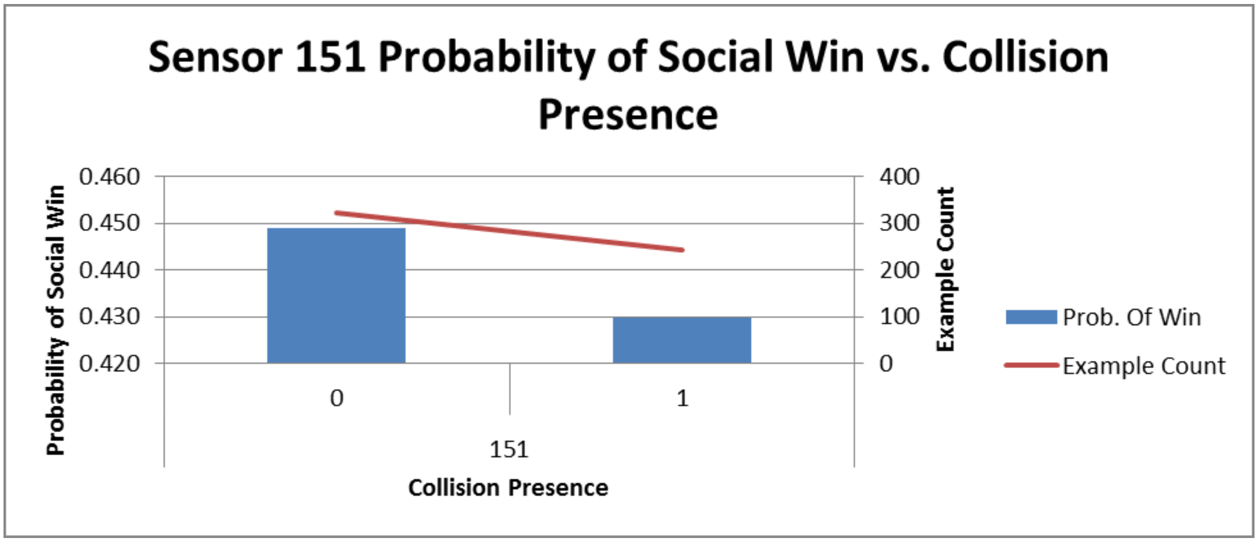


Row Labels	Prob. Of Win	Example Count
0	0.333	69
1	0.456	447
2	0.449	49

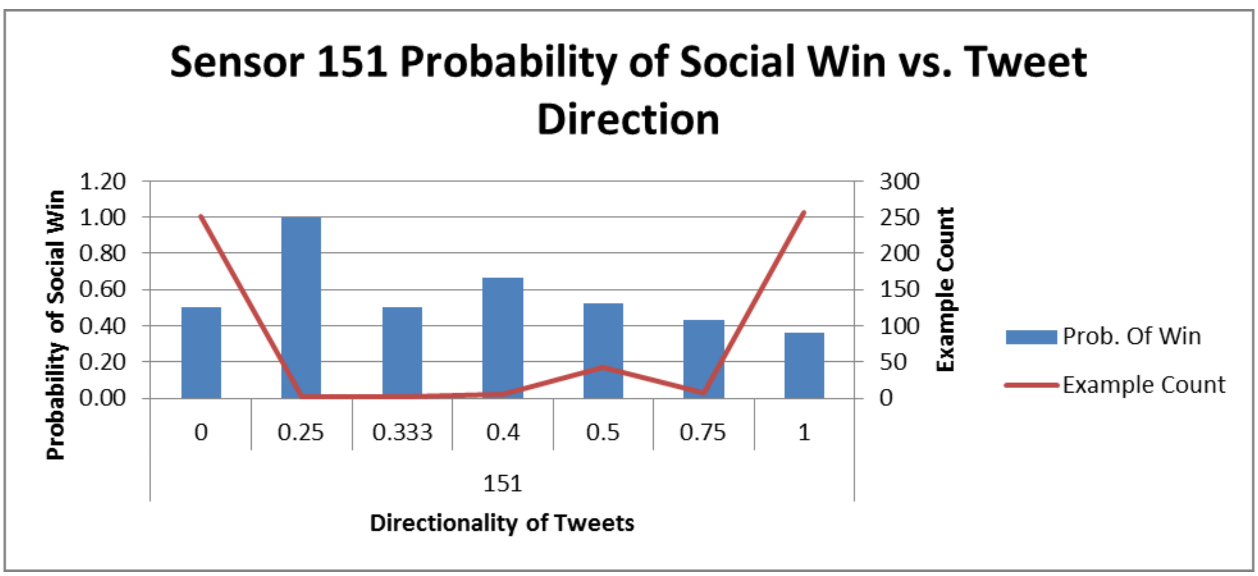


Row Labels	Prob. Of Win	Example Count

0	0.418	261
1	0.461	304

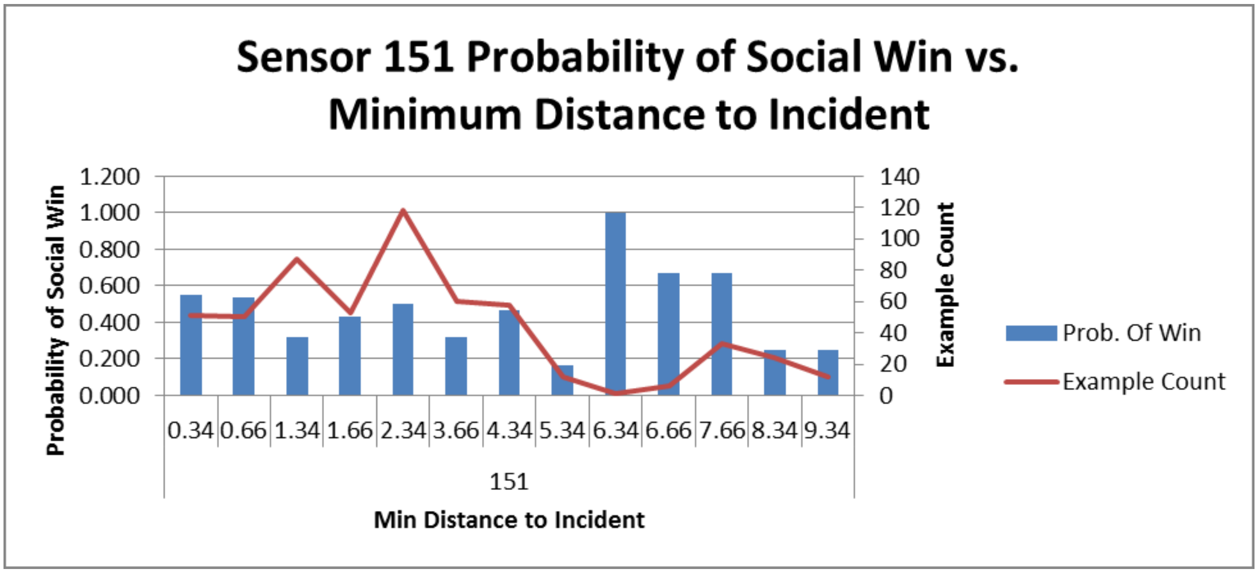


Row Labels	Prob. Of Win	Example Count
0	0.449	323
1	0.430	242

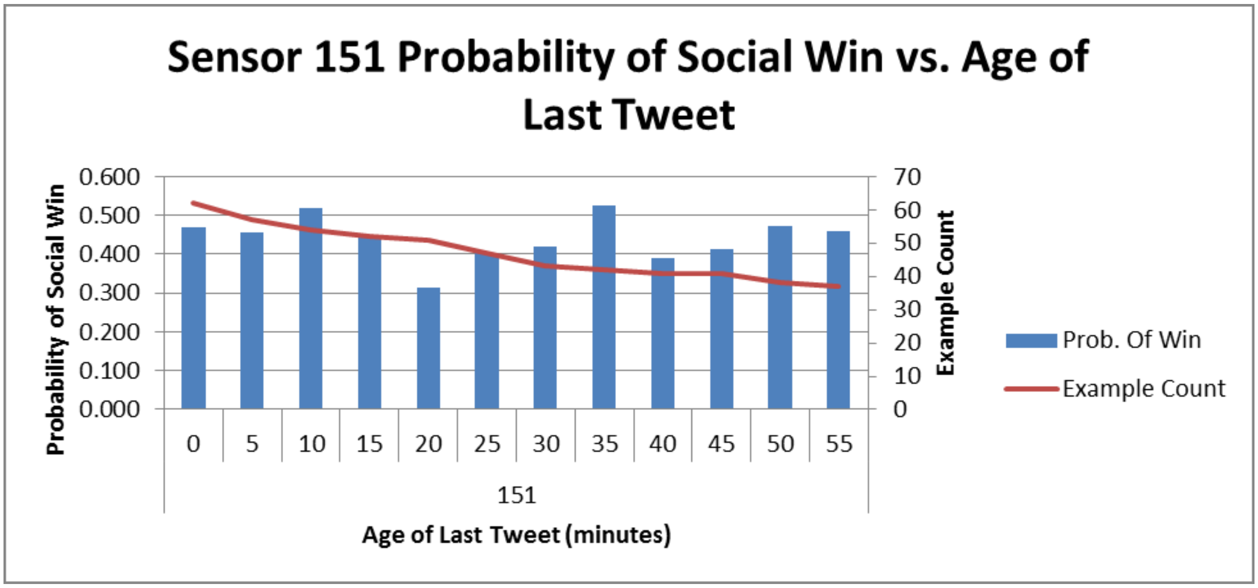


Row Labels	Prob. Of Win	Example Count
0	0.50	251
0.25	1.00	1
0.333	0.50	2

0.4	0.67	6
0.5	0.52	42
0.75	0.43	7
1	0.36	256

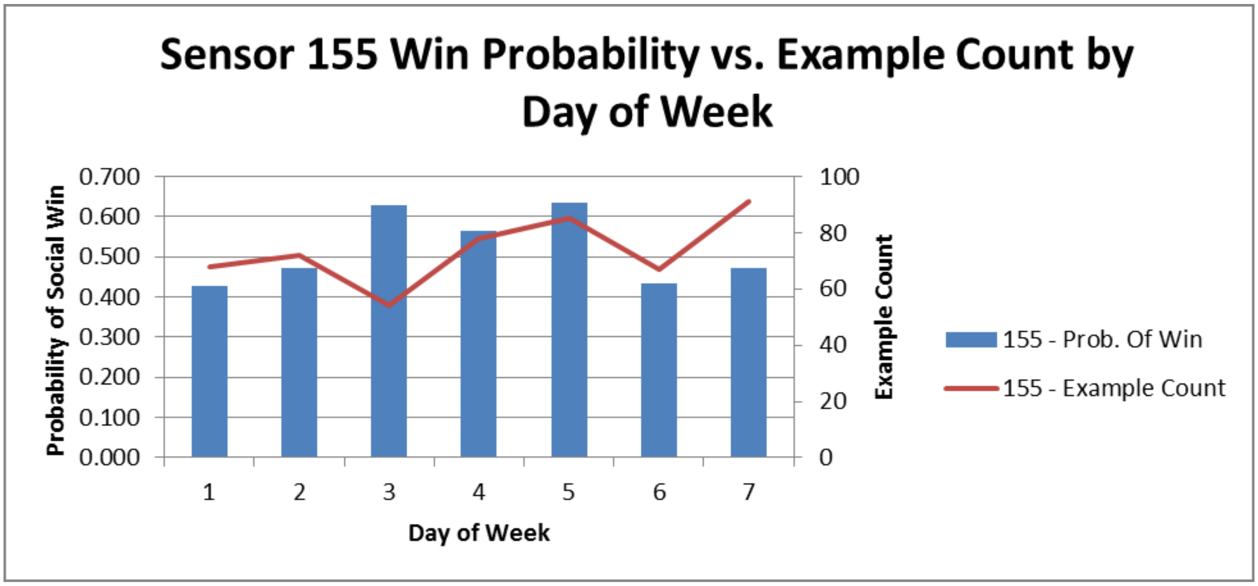


Row Labels	Prob. Of Win	Example Count
0.34	0.549	51
0.66	0.540	50
1.34	0.322	87
1.66	0.434	53
2.34	0.500	118
3.66	0.317	60
4.34	0.466	58
5.34	0.167	12
6.34	1.000	1
6.66	0.667	6
7.66	0.667	33
8.34	0.250	24
9.34	0.250	12

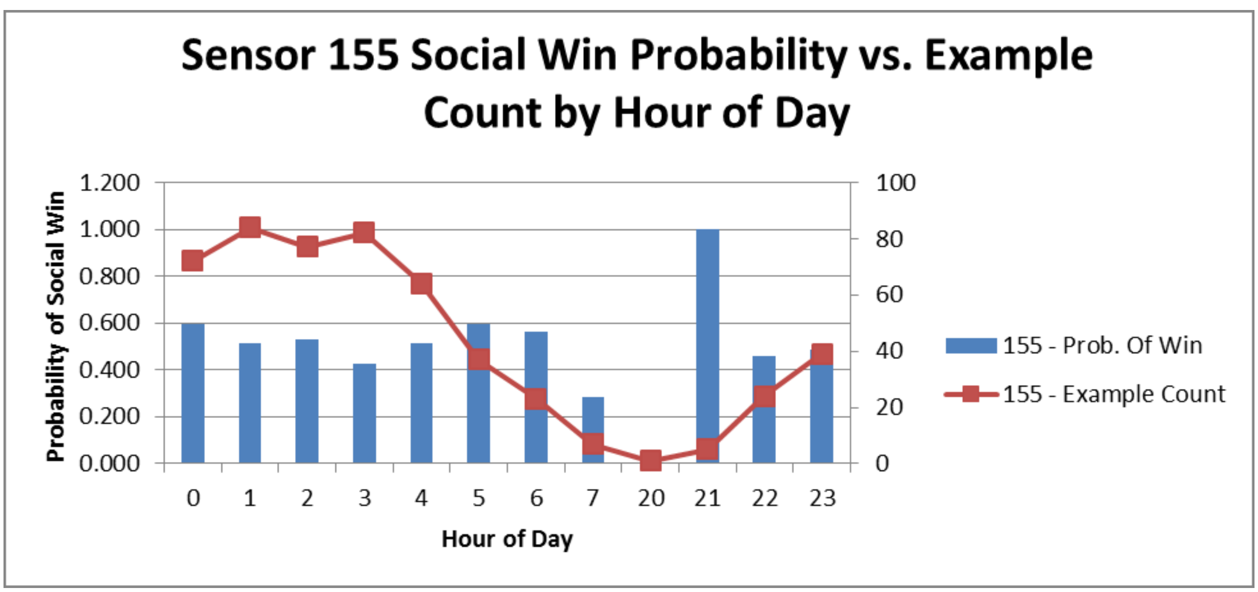


Row Labels	Prob. Of Win	Example Count
0	0.468	62
5	0.456	57
10	0.519	54
15	0.442	52
20	0.314	51
25	0.404	47
30	0.419	43
35	0.524	42
40	0.390	41
45	0.415	41
50	0.474	38
55	0.459	37

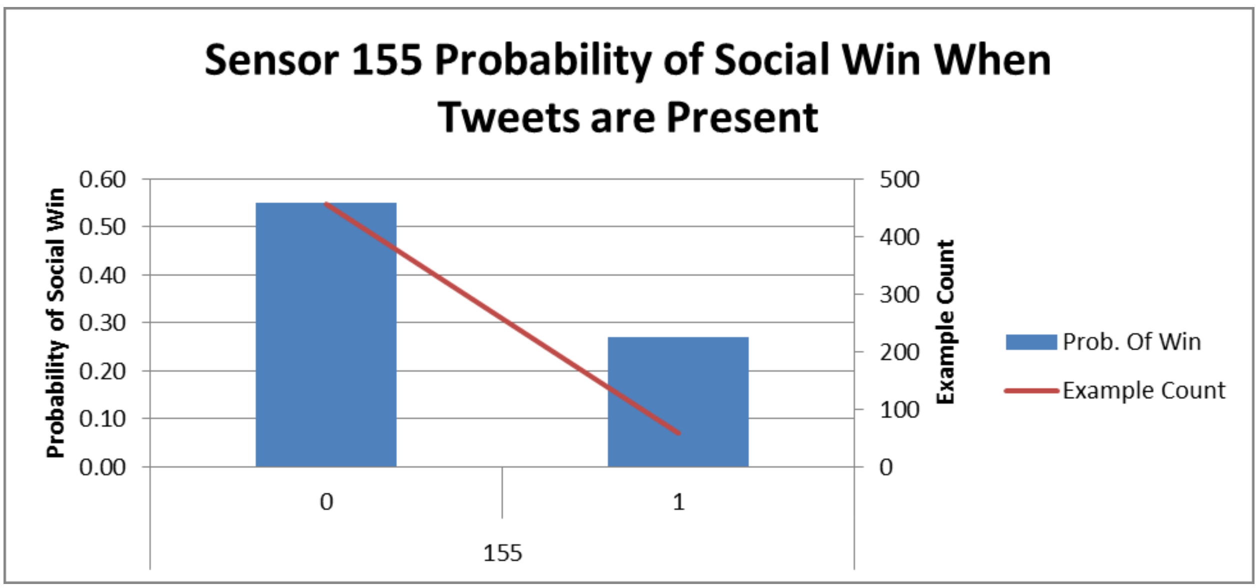
Sensor 155



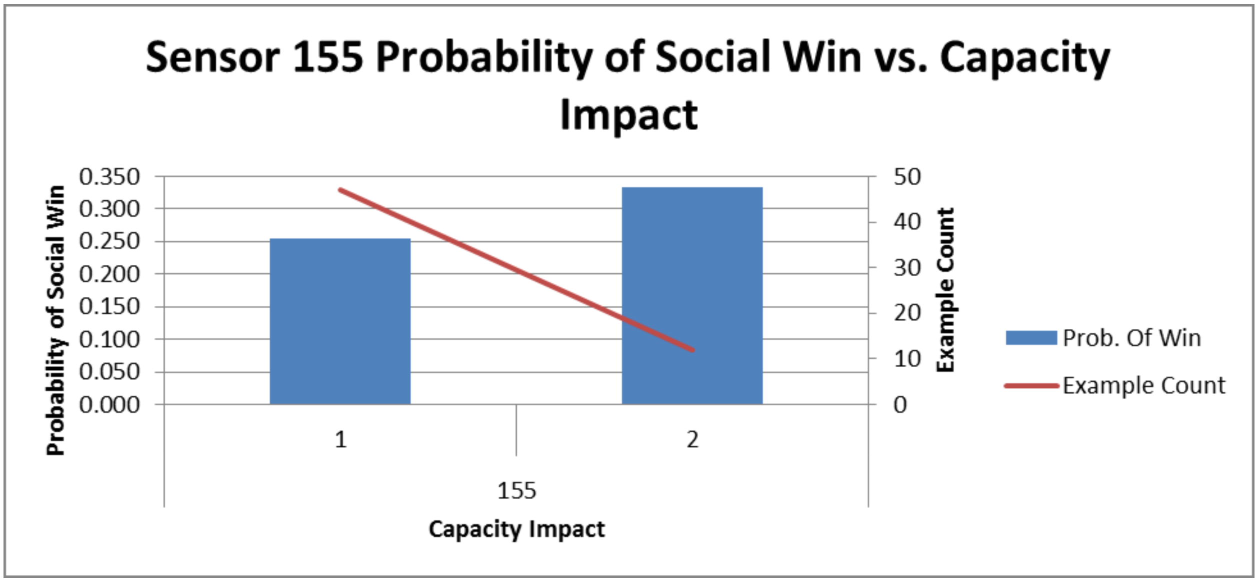
Row Labels	Prob. Of Win	Example Count
1	0.426	68
2	0.472	72
3	0.630	54
4	0.564	78
5	0.635	85
6	0.433	67
7	0.473	91



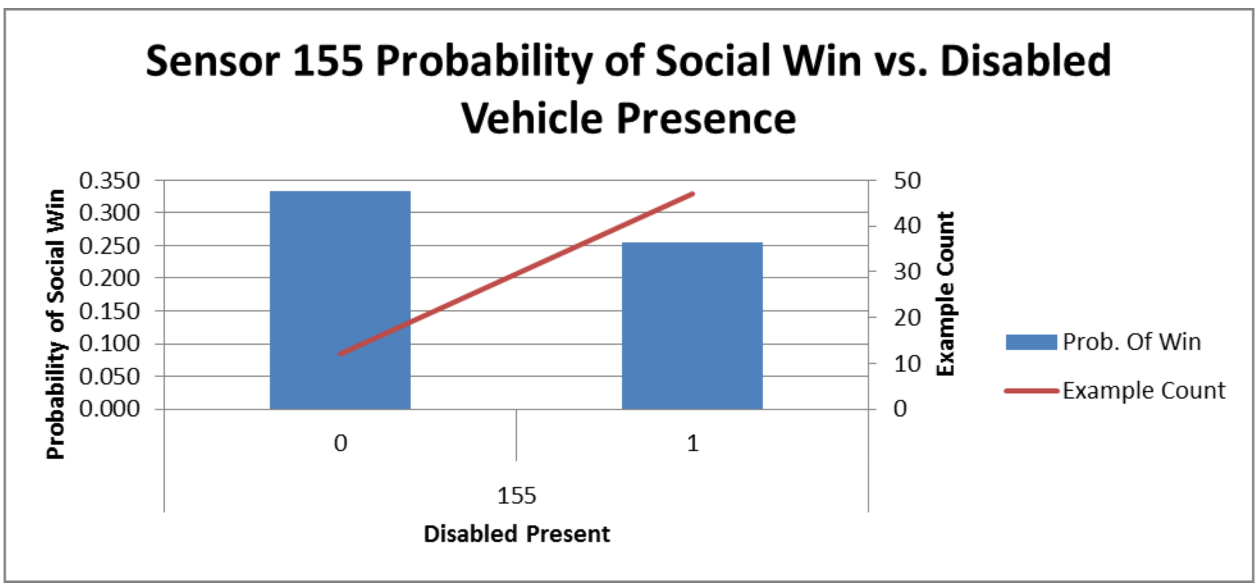
Row Labels	Prob. Of Win	Example Count
0	0.597	72
1	0.512	84
2	0.532	77
3	0.427	82
4	0.516	64
5	0.595	37
6	0.565	23
7	0.286	7
20	0.000	1
21	1.000	5
22	0.458	24
23	0.487	39



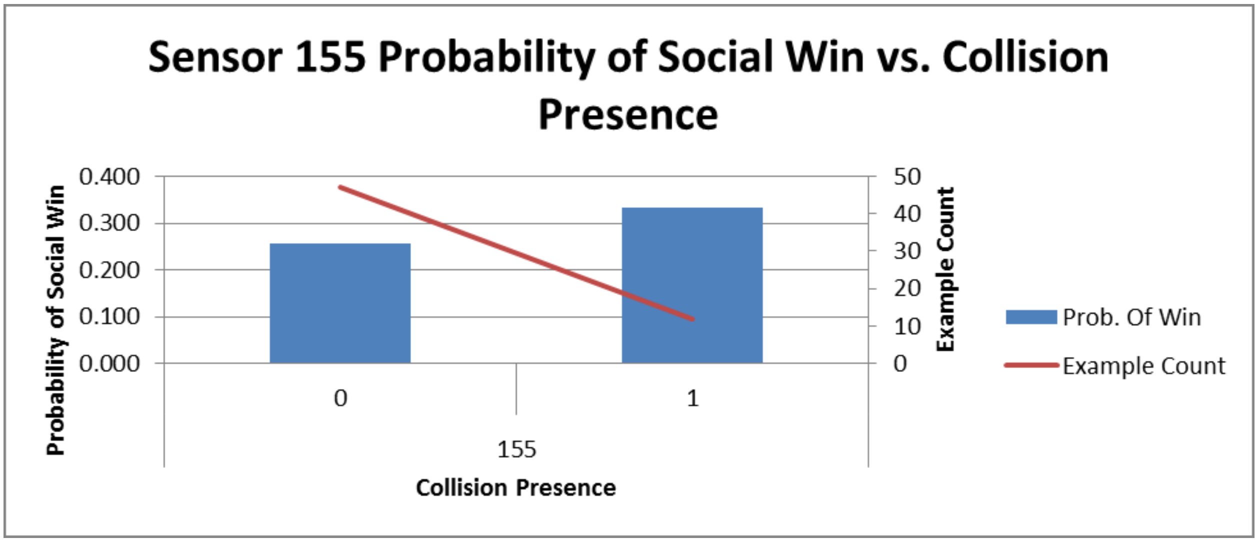
Row Labels	Prob. Of Win	Example Count
0	0.55	456
1	0.27	59



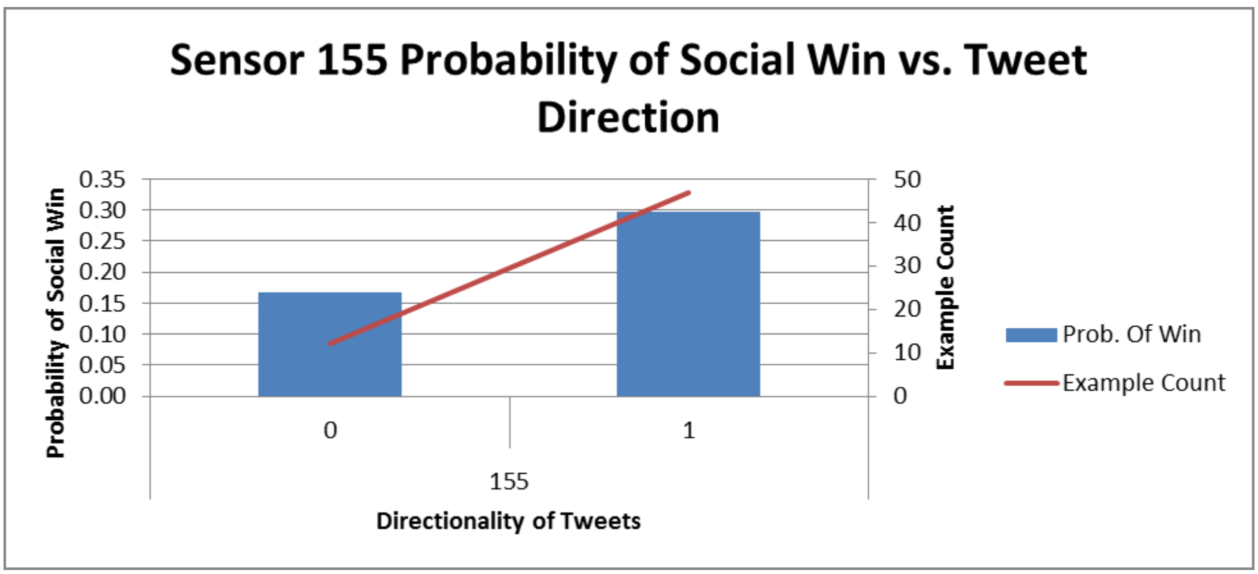
Row Labels	Prob. Of Win	Example Count
1	0.255	47
2	0.333	12



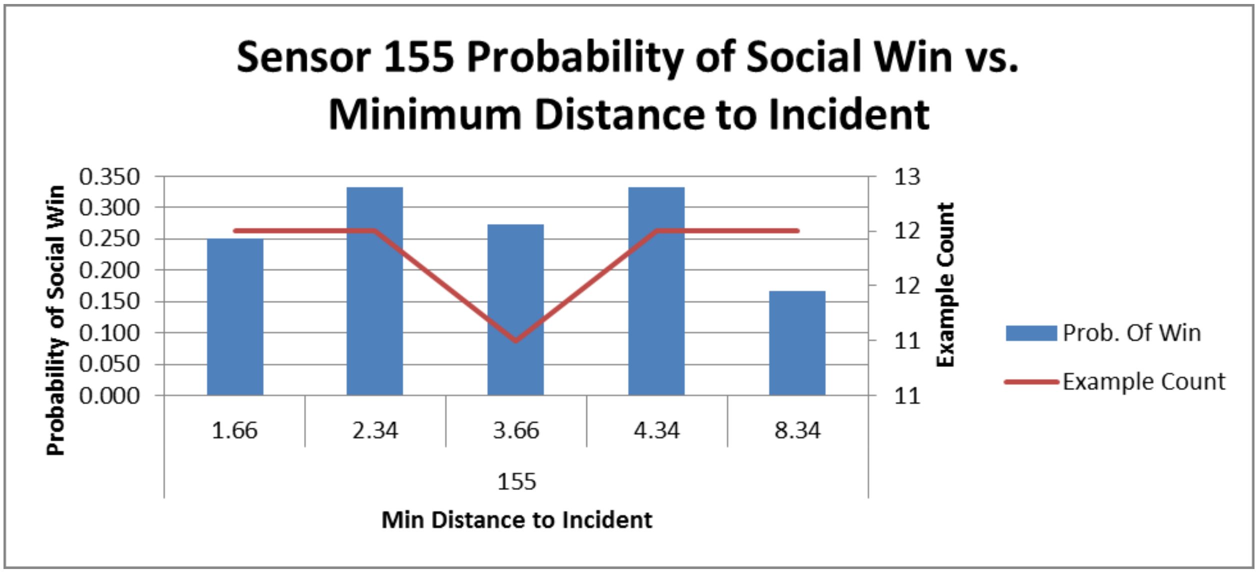
Row Labels	Prob. Of Win	Example Count
0	0.333	12
1	0.255	47



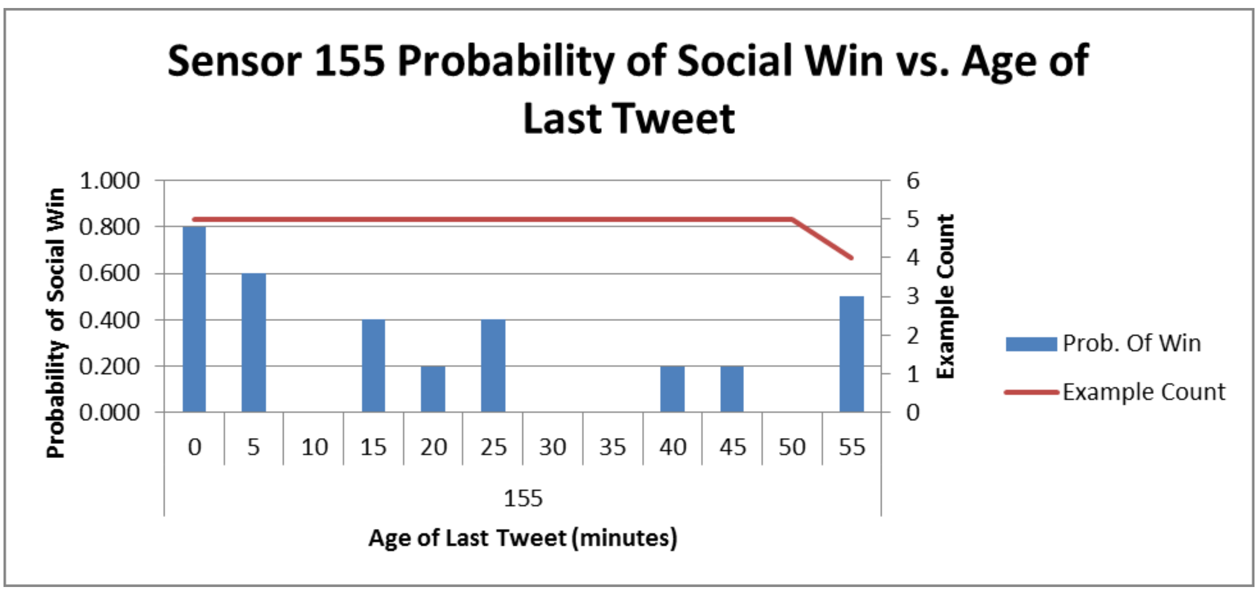
Row Labels	Prob. Of Win	Example Count
0	0.255	47
1	0.333	12



Row Labels	Prob. Of Win	Example Count
0	0.17	12
1	0.30	47



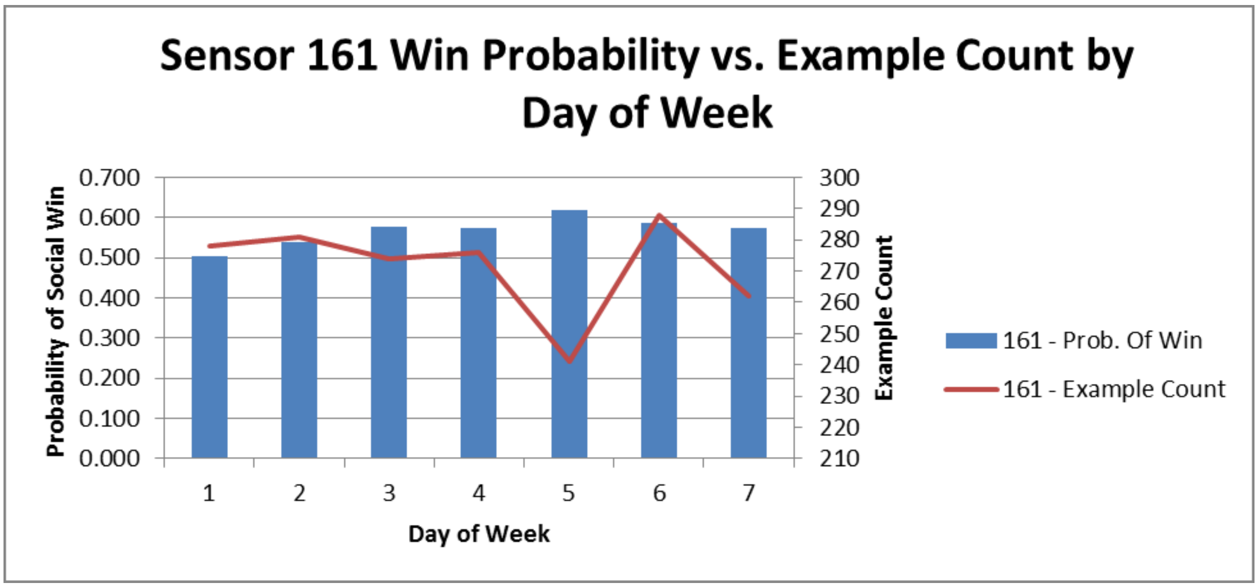
Row Labels	Prob. Of Win	Example Count
1.66	0.250	12
2.34	0.333	12
3.66	0.273	11
4.34	0.333	12
8.34	0.167	12



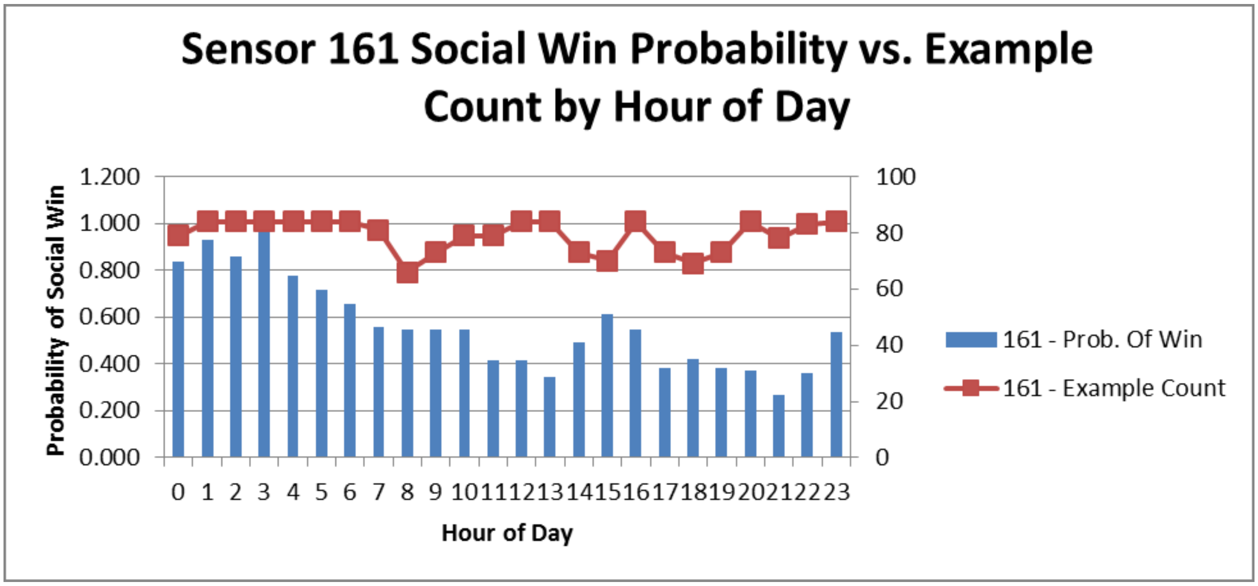
Row Labels	Prob. Of Win	Example Count
0	0.800	5
5	0.600	5
10	0.000	5

15	0.400	5
20	0.200	5
25	0.400	5
30	0.000	5
35	0.000	5
40	0.200	5
45	0.200	5
50	0.000	5
55	0.500	4

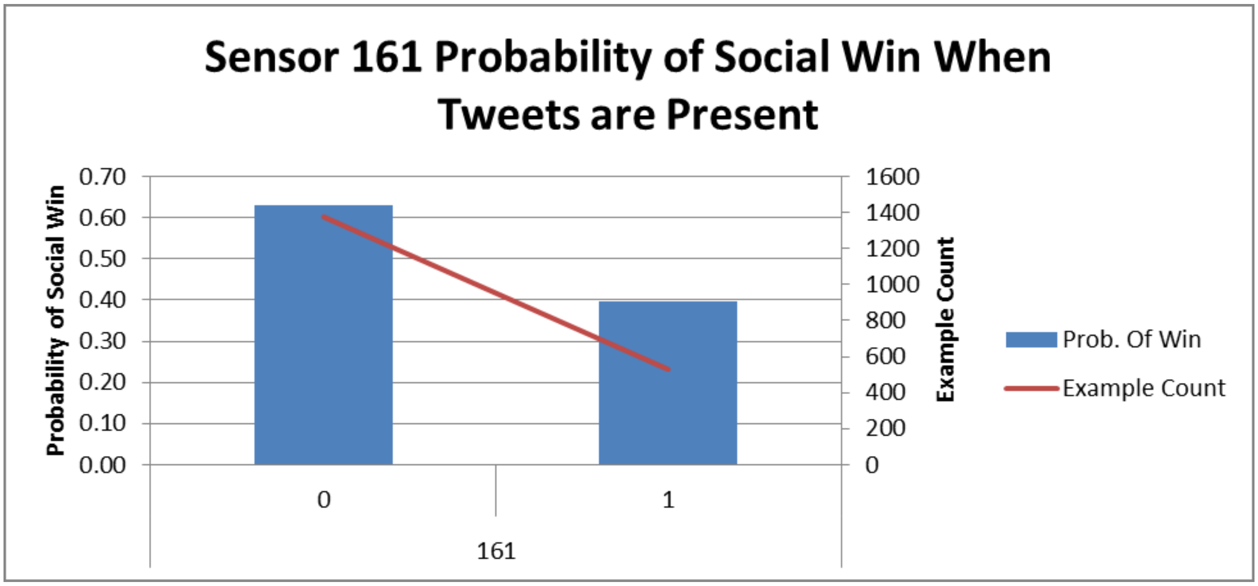
Sensor 161



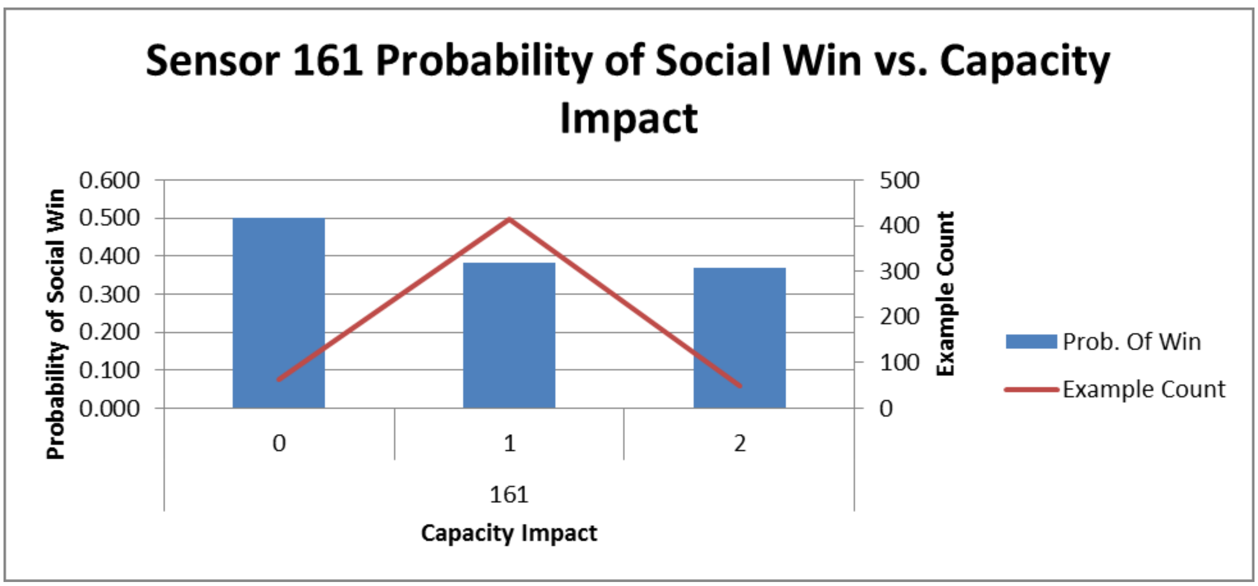
Row Labels	Prob. Of Win	Example Count
1	0.504	278
2	0.537	281
3	0.577	274
4	0.572	276
5	0.618	241
6	0.587	288
7	0.573	262



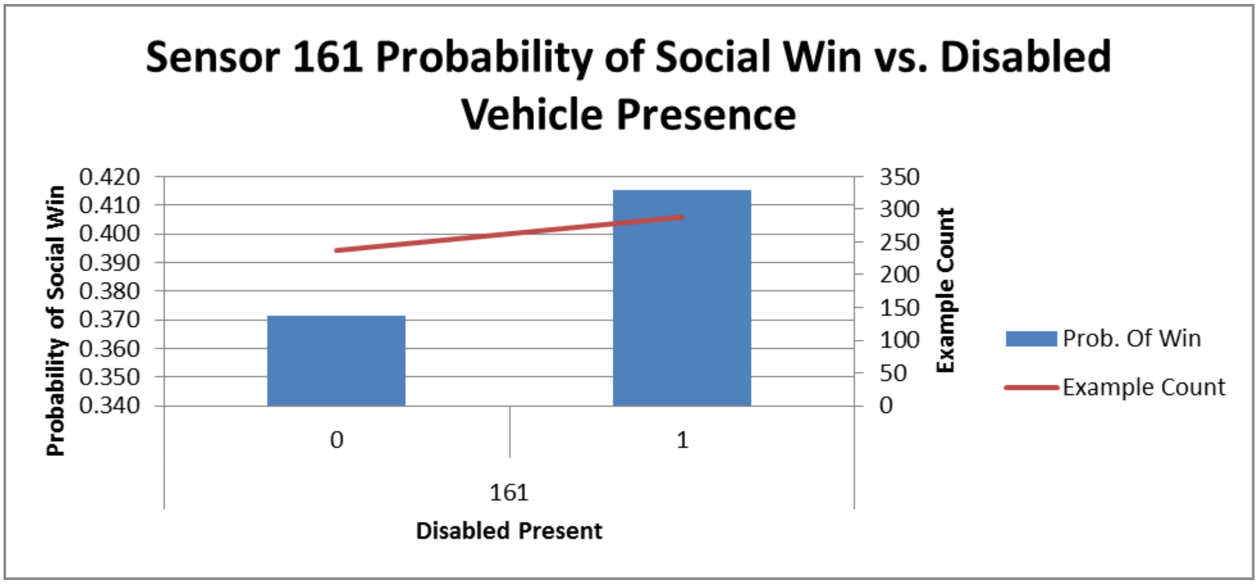
Row Labels	Prob. Of Win	Example Count
0	0.835	79
1	0.929	84
2	0.857	84
3	0.964	84
4	0.774	84
5	0.714	84
6	0.655	84
7	0.556	81
8	0.545	66
9	0.548	73
10	0.544	79
11	0.418	79
12	0.417	84
13	0.345	84
14	0.493	73
15	0.614	70
16	0.548	84
17	0.384	73
18	0.420	69
19	0.384	73
20	0.369	84
21	0.269	78
22	0.361	83
23	0.536	84



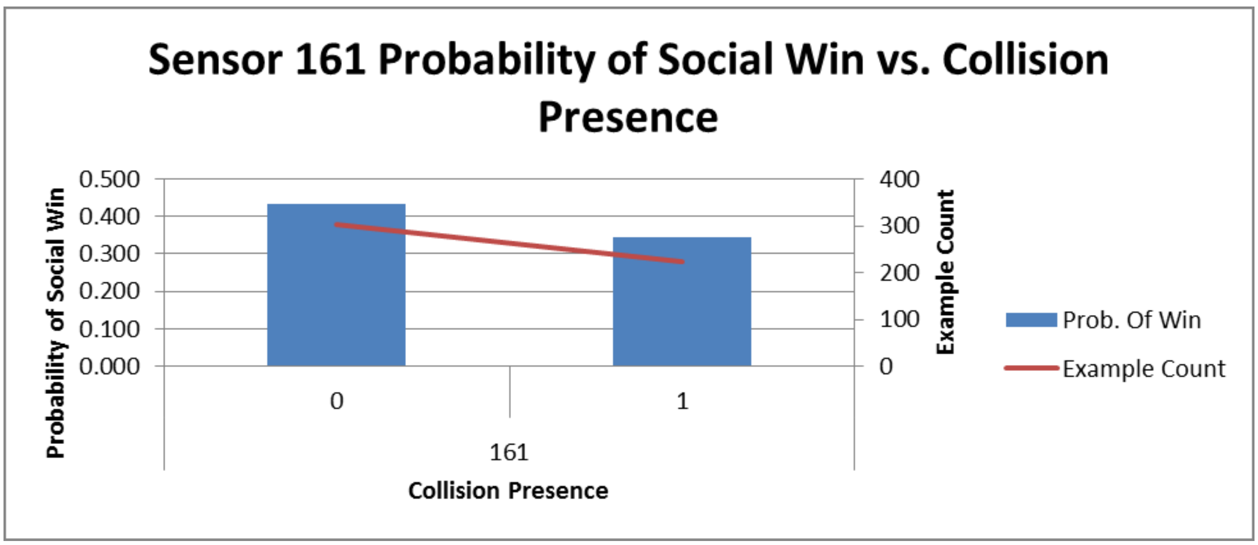
Row Labels	Prob. Of Win	Example Count
0	0.63	1374
1	0.40	526



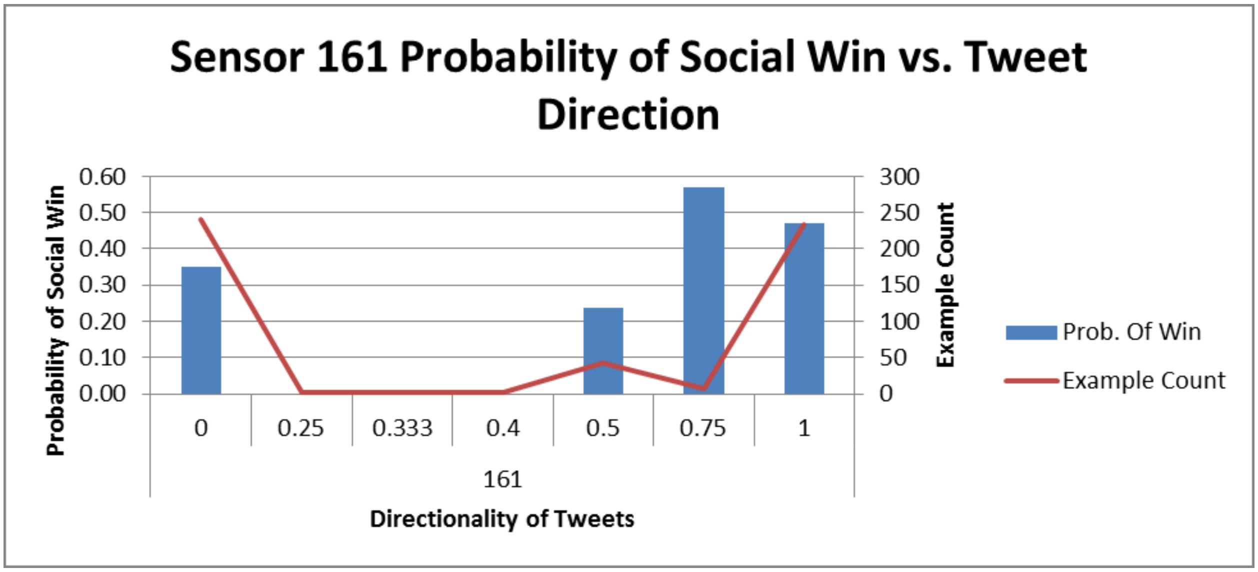
Row Labels	Prob. Of Win	Example Count
0	0.500	62
1	0.383	415
2	0.367	49



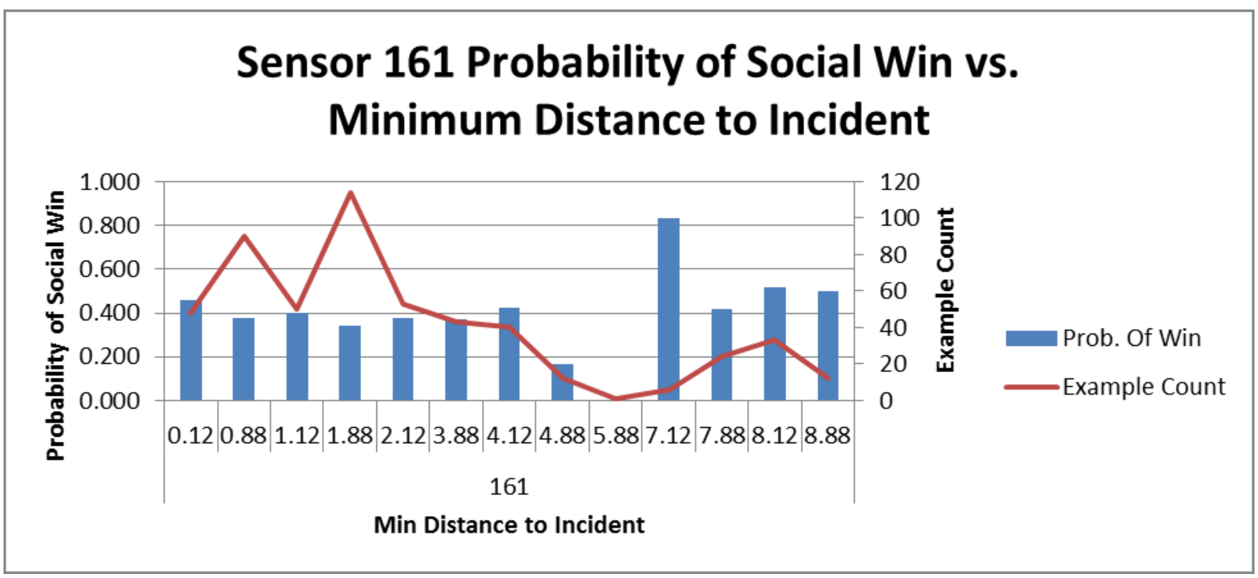
Row Labels	Prob. Of Win	Example Count
0	0.371	237
1	0.415	289



Row Labels	Prob. Of Win	Example Count
0	0.432	303
1	0.345	223

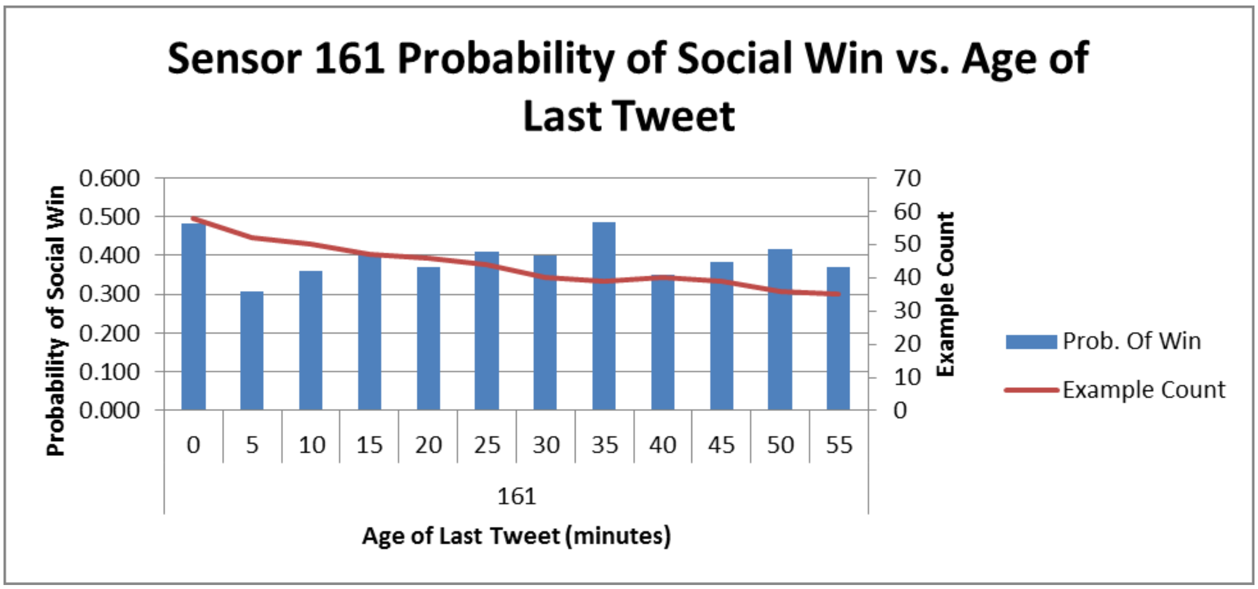


Row Labels	Prob. Of Win	Example Count
0	0.35	240
0.25	0.00	1
0.333	0.00	2
0.4	0.00	1
0.5	0.24	42
0.75	0.57	7
1	0.47	233



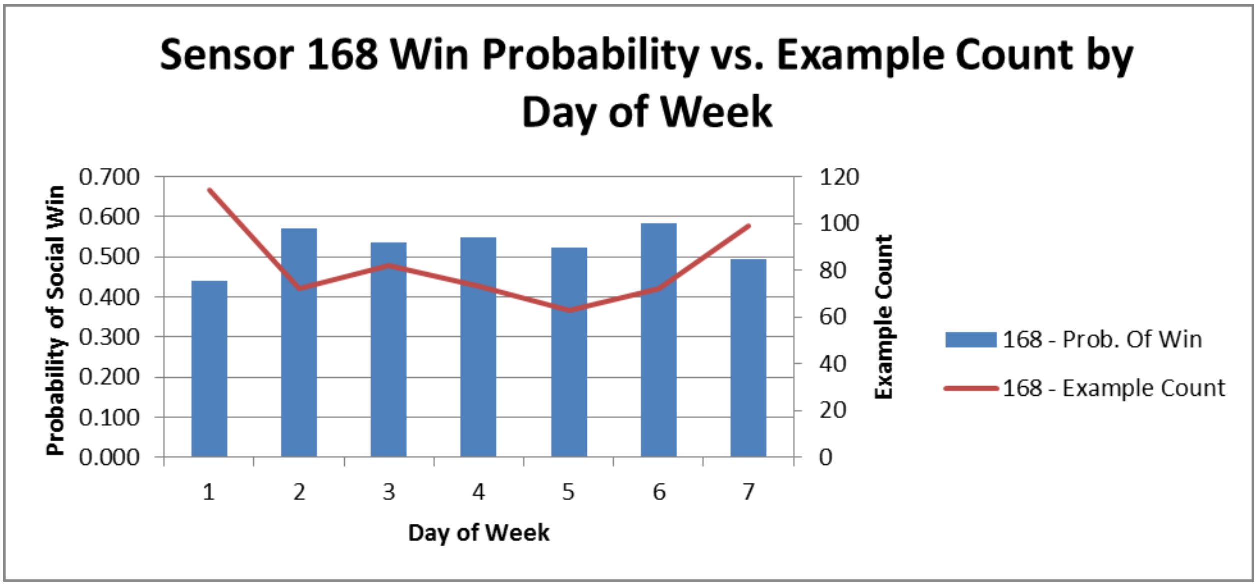
Row Labels	Prob. Of Win	Example Count
0.12	0.458	48

0.88	0.378	90
1.12	0.400	50
1.88	0.342	114
2.12	0.377	53
3.88	0.372	43
4.12	0.425	40
4.88	0.167	12
5.88	0.000	1
7.12	0.833	6
7.88	0.417	24
8.12	0.515	33
8.88	0.500	12

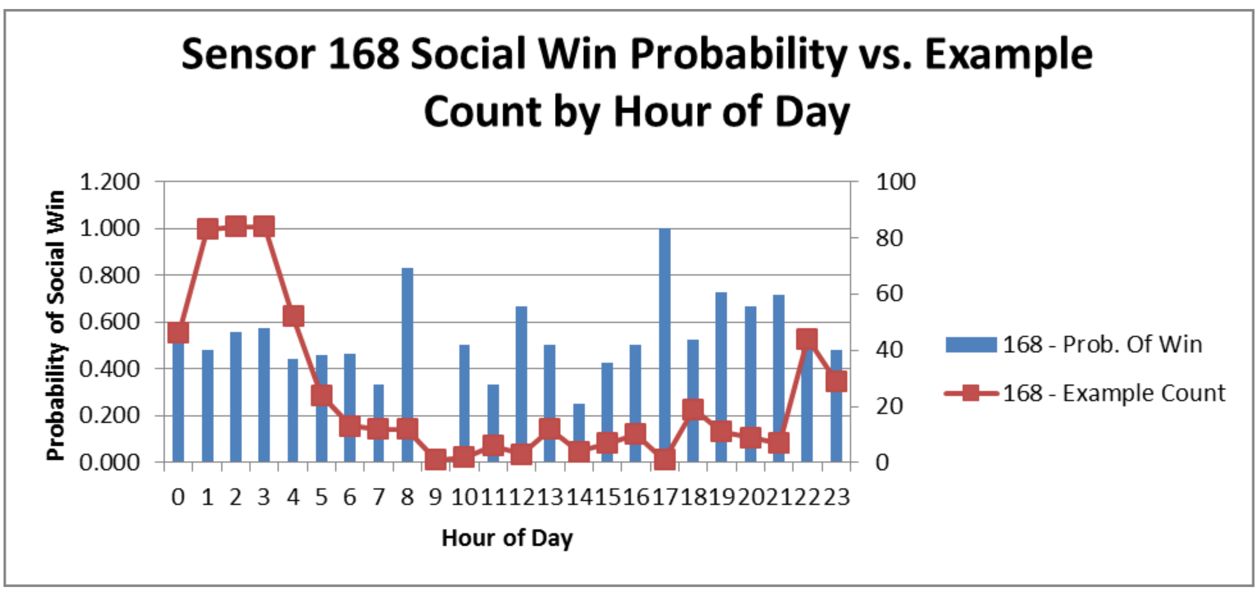


Row Labels	Prob. Of Win	Example Count
0	0.483	58
5	0.308	52
10	0.360	50
15	0.404	47
20	0.370	46
25	0.409	44
30	0.400	40
35	0.487	39
40	0.350	40
45	0.385	39
50	0.417	36
55	0.371	35

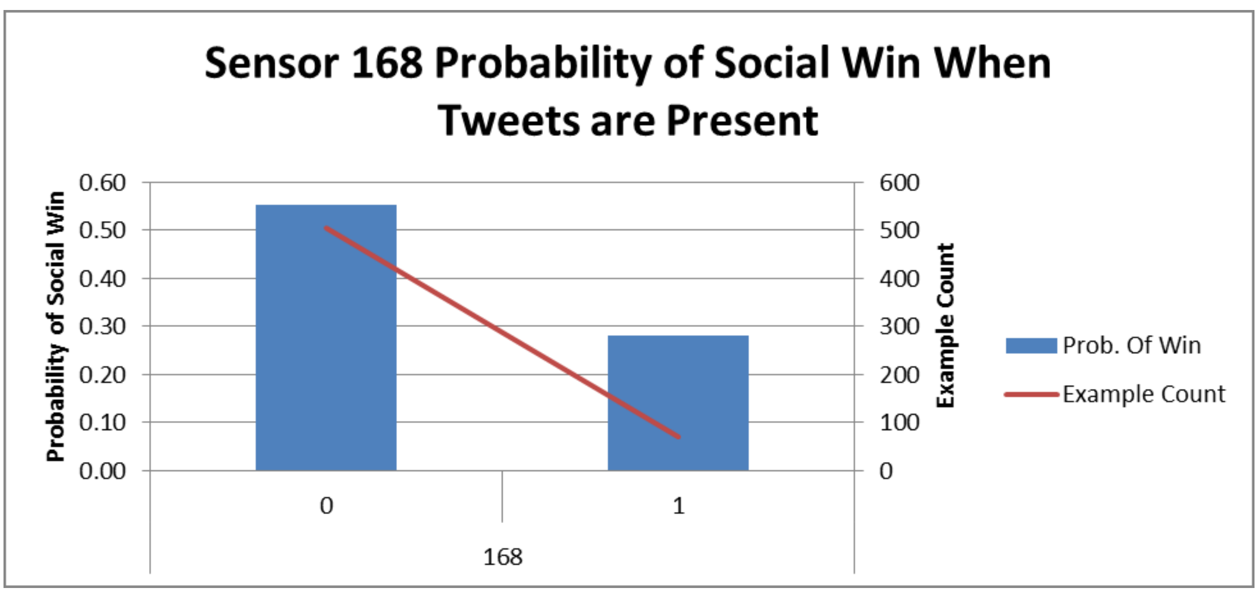
Sensor 168



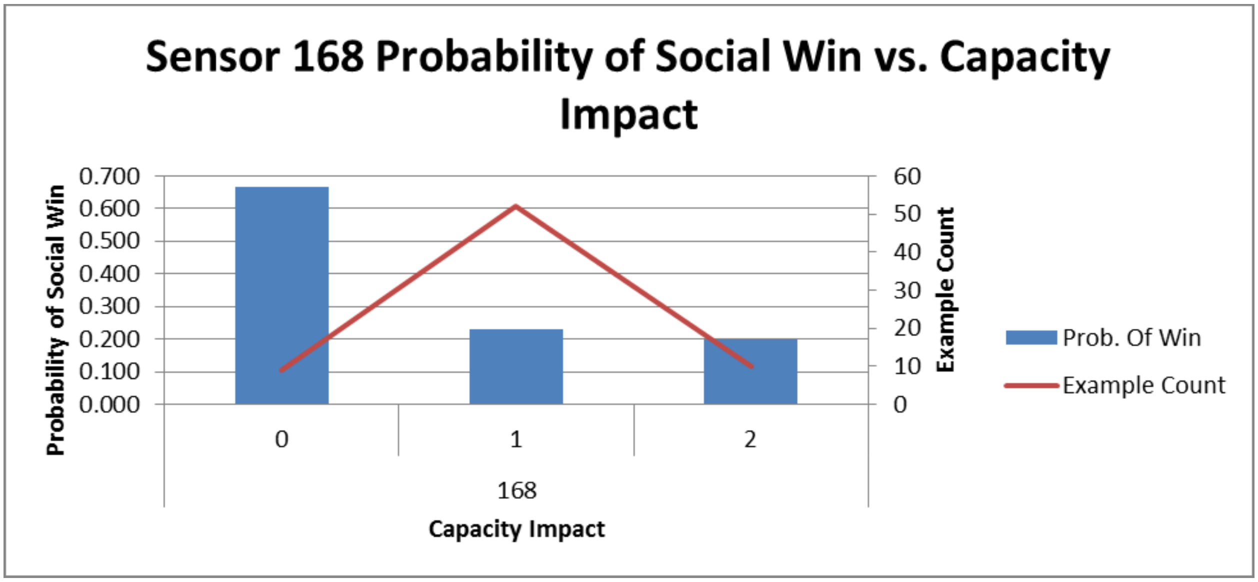
Row Labels	Prob. Of Win	Example Count
1	0.439	114
2	0.569	72
3	0.537	82
4	0.548	73
5	0.524	63
6	0.583	72
7	0.495	99



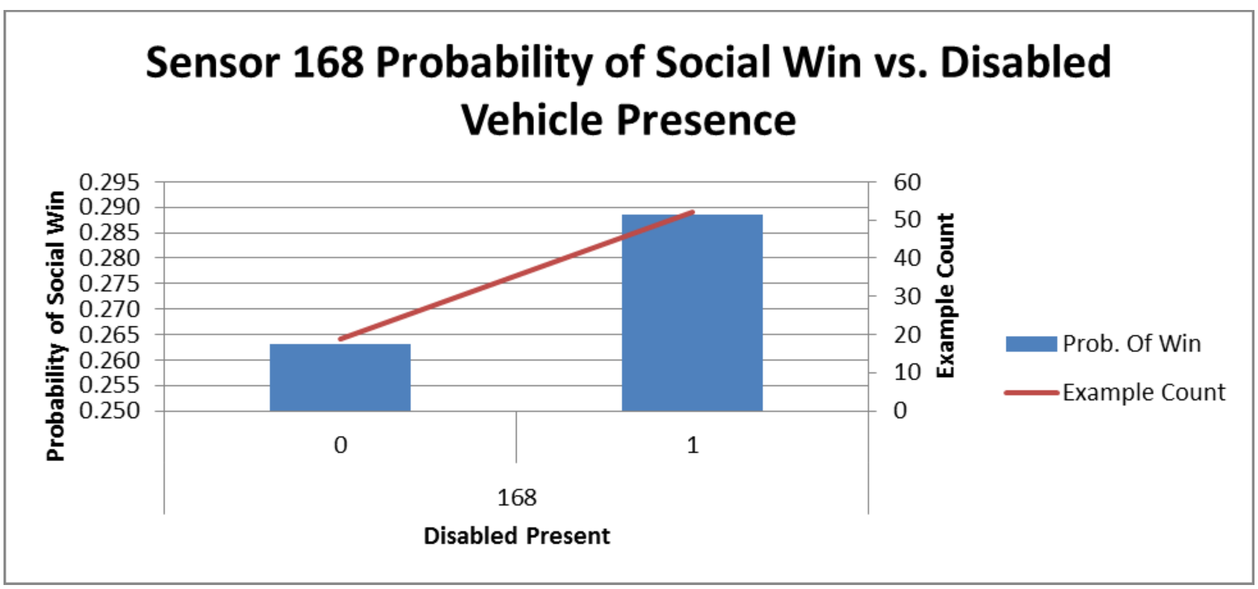
Row Labels	Prob. Of Win	Example Count
0	0.522	46
1	0.482	83
2	0.560	84
3	0.571	84
4	0.442	52
5	0.458	24
6	0.462	13
7	0.333	12
8	0.833	12
9	0.000	1
10	0.500	2
11	0.333	6
12	0.667	3
13	0.500	12
14	0.250	4
15	0.429	7
16	0.500	10
17	1.000	1
18	0.526	19
19	0.727	11
20	0.667	9
21	0.714	7
22	0.500	44
23	0.483	29



Row Labels	Prob. Of Win	Example Count
0	0.55	504
1	0.28	71

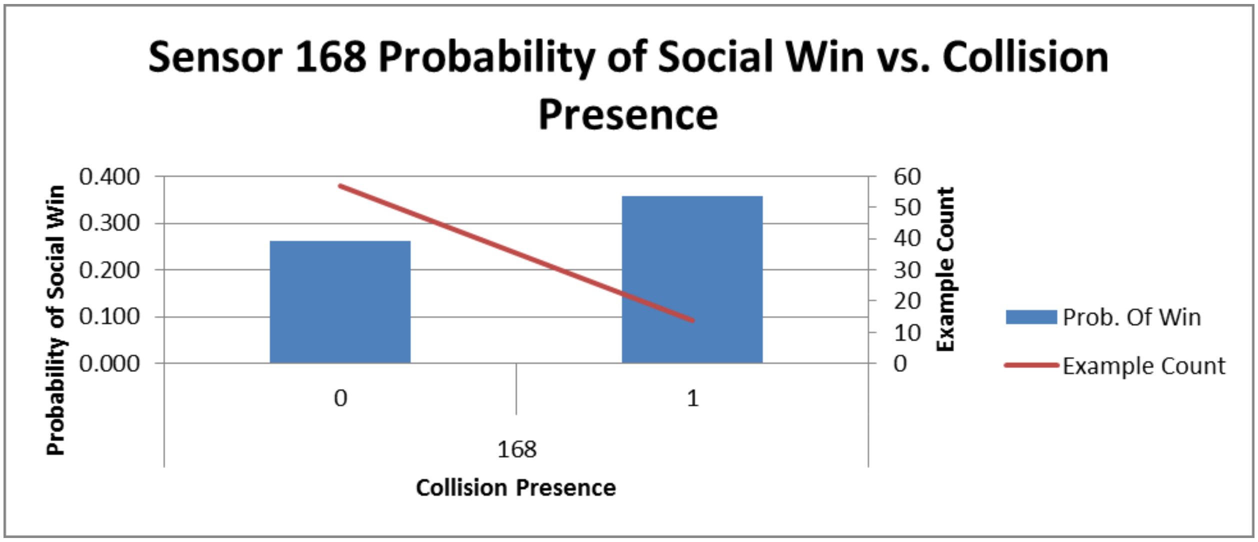


Row Labels	Prob. Of Win	Example Count
0	0.667	9
1	0.231	52
2	0.200	10

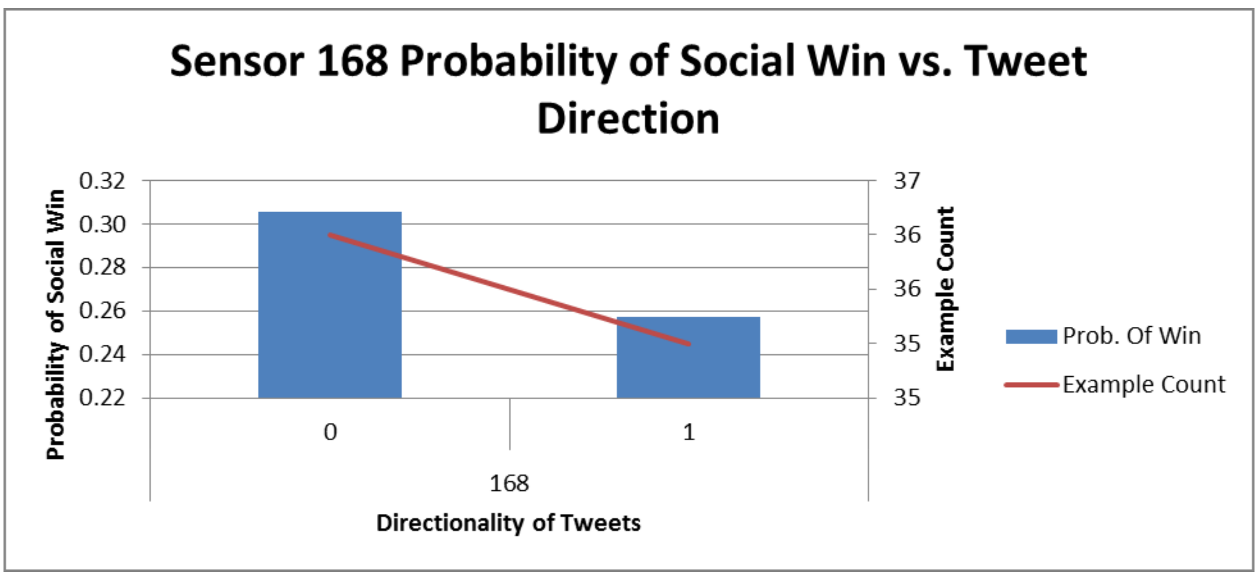


Row Labels	Prob. Of Win	Example Count

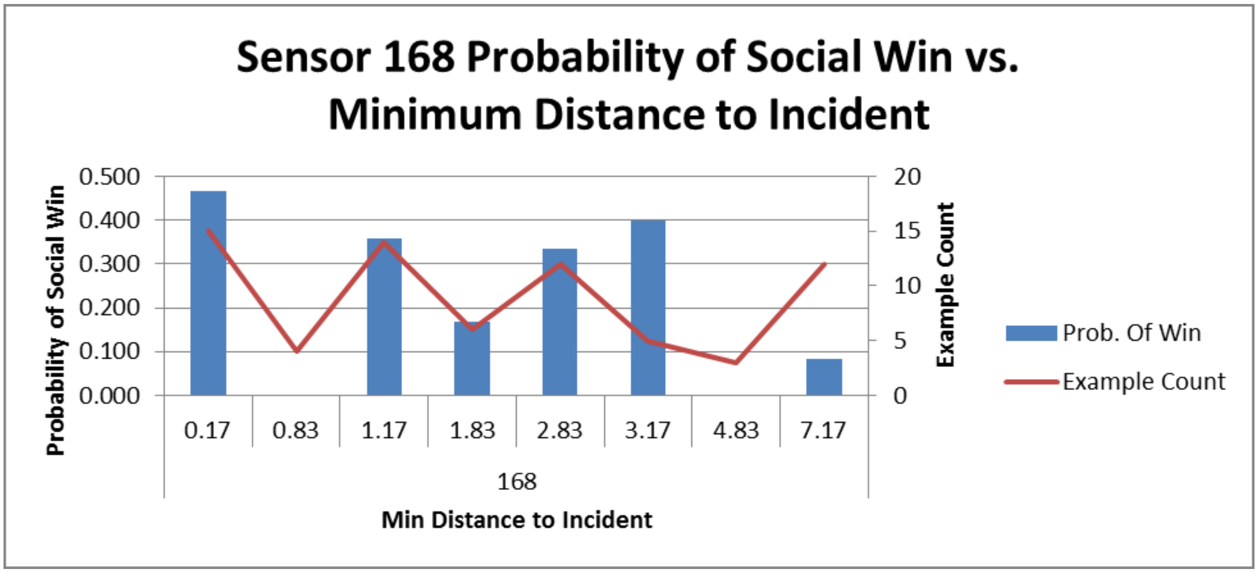
0	0.263	19
1	0.288	52



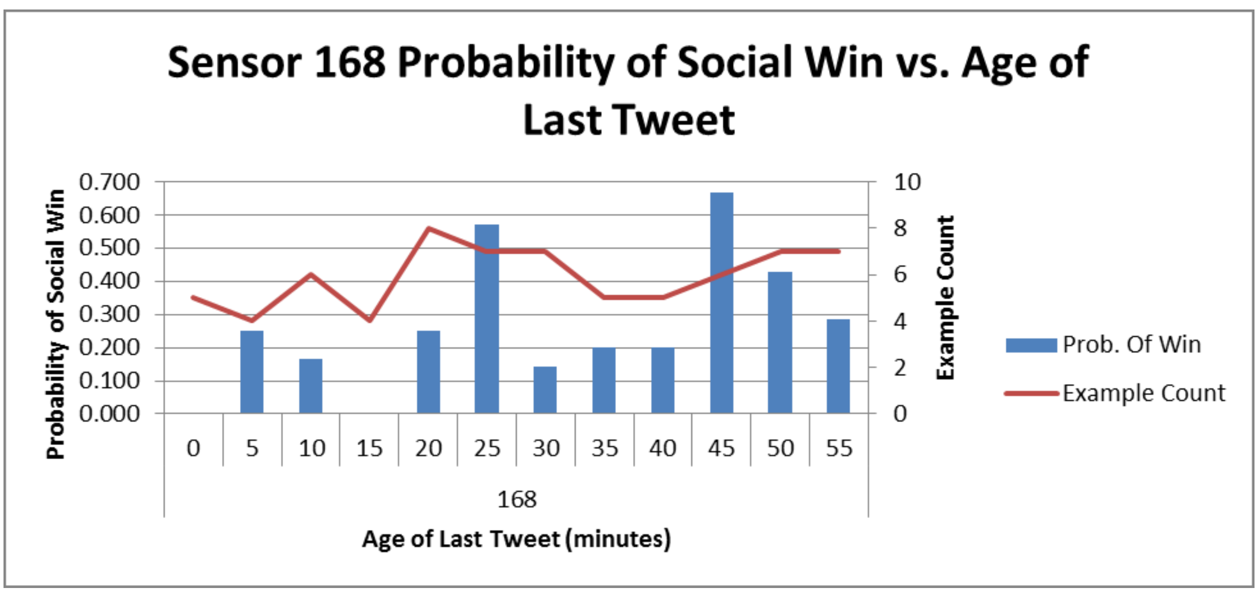
Row Labels	Prob. Of Win	Example Count
0	0.263	57
1	0.357	14



Row Labels	Prob. Of Win	Example Count
0	0.31	36
1	0.26	35



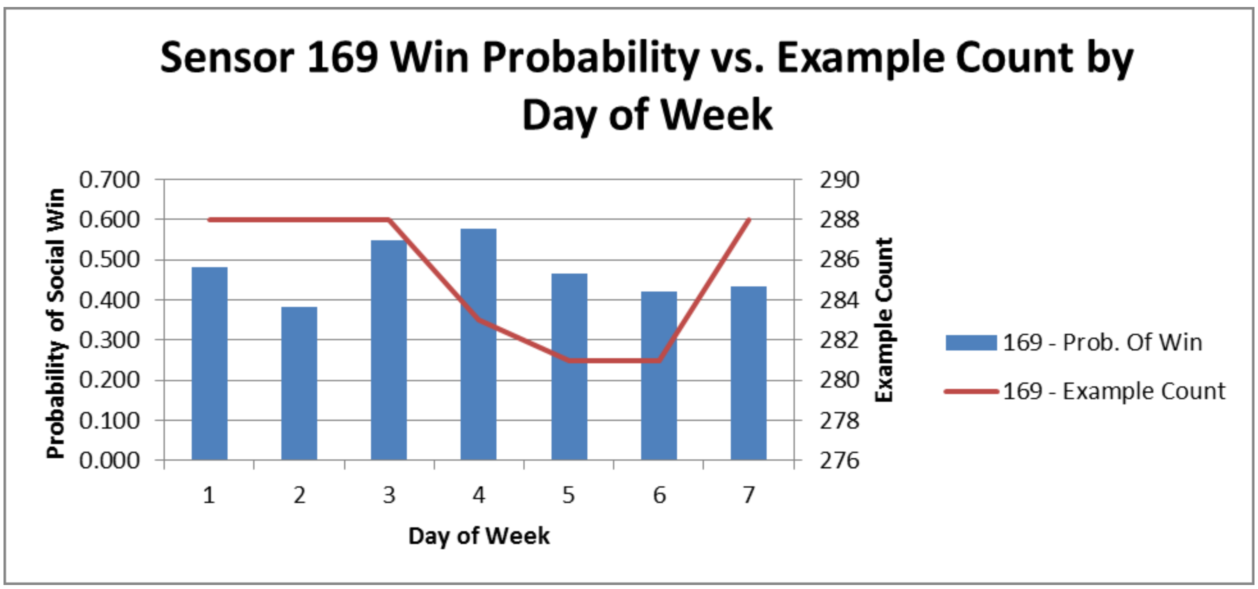
Row Labels	Prob. Of Win	Example Count
0.17	0.467	15
0.83	0.000	4
1.17	0.357	14
1.83	0.167	6
2.83	0.333	12
3.17	0.400	5
4.83	0.000	3
7.17	0.083	12



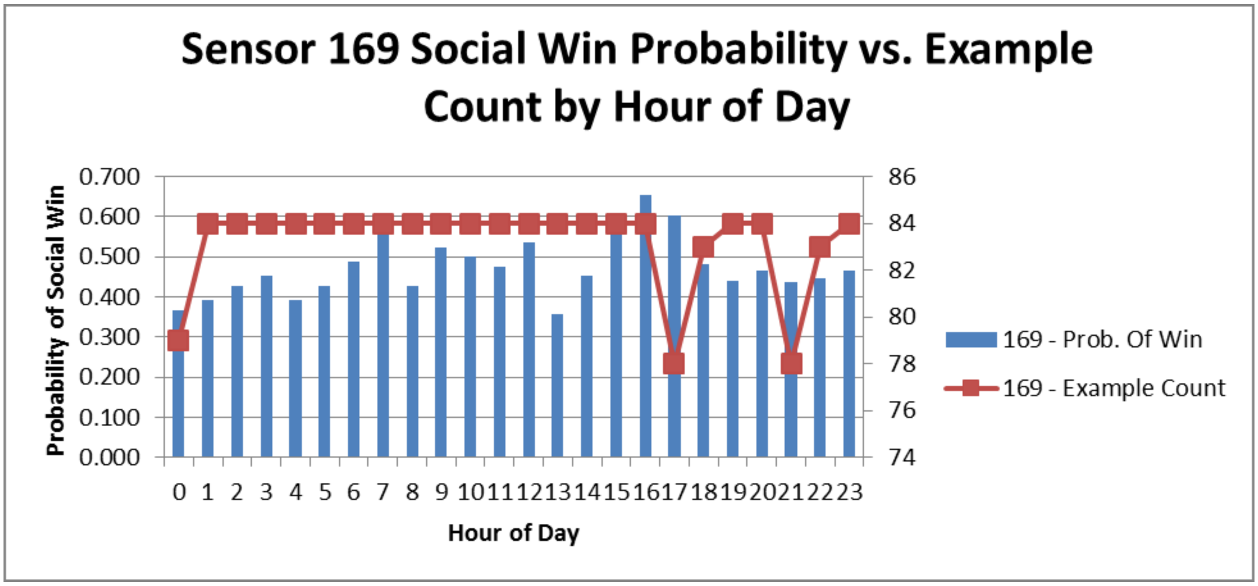
Row Labels	Prob. Of Win	Example Count

0	0.000	5
5	0.250	4
10	0.167	6
15	0.000	4
20	0.250	8
25	0.571	7
30	0.143	7
35	0.200	5
40	0.200	5
45	0.667	6
50	0.429	7
55	0.286	7

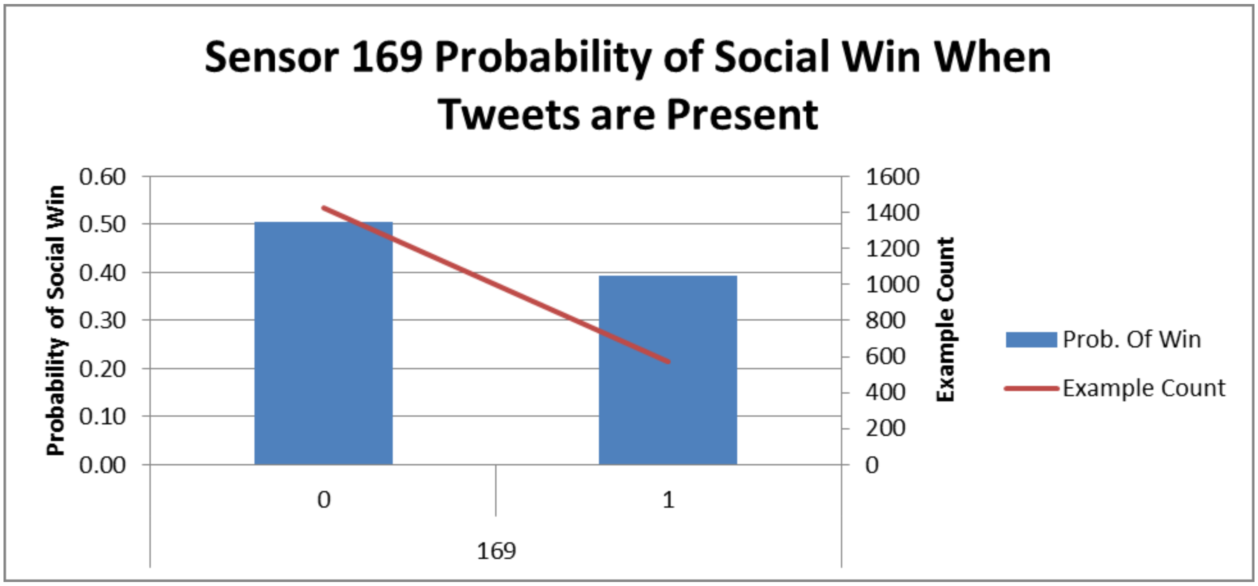
Sensor 169



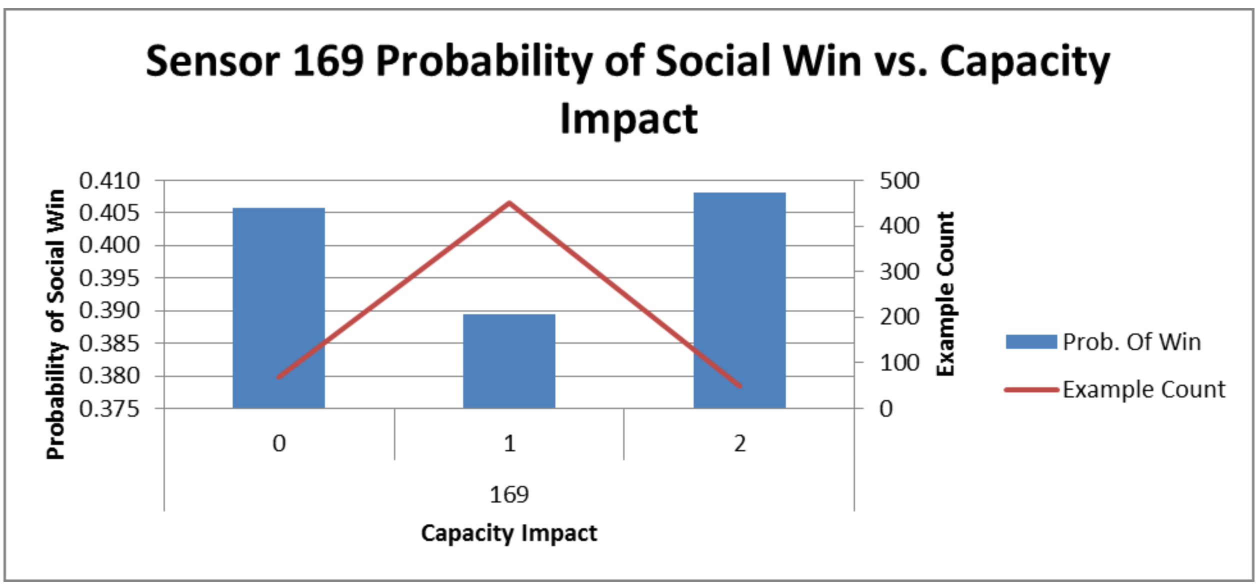
Row Labels	Prob. Of Win	Example Count
1	0.483	288
2	0.382	288
3	0.549	288
4	0.576	283
5	0.466	281
6	0.420	281
7	0.434	288



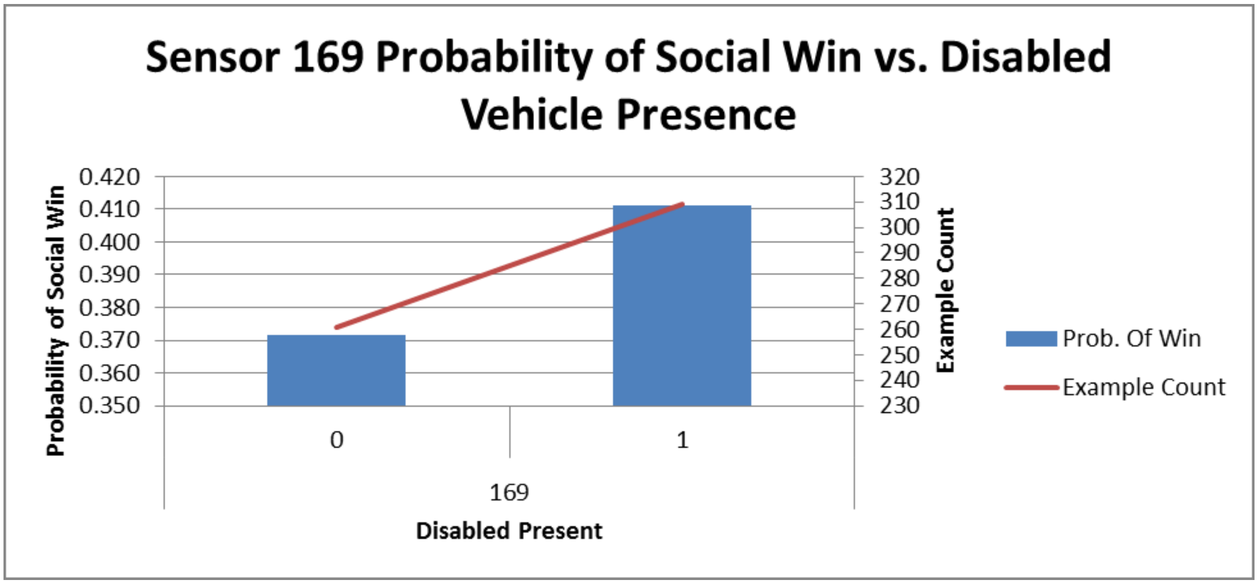
Row Labels	Prob. Of Win	Example Count
0	0.367	79
1	0.393	84
2	0.429	84
3	0.452	84
4	0.393	84
5	0.429	84
6	0.488	84
7	0.560	84
8	0.429	84
9	0.524	84
10	0.500	84
11	0.476	84
12	0.536	84
13	0.357	84
14	0.452	84
15	0.571	84
16	0.655	84
17	0.603	78
18	0.482	83
19	0.440	84
20	0.464	84
21	0.436	78
22	0.446	83
23	0.464	84



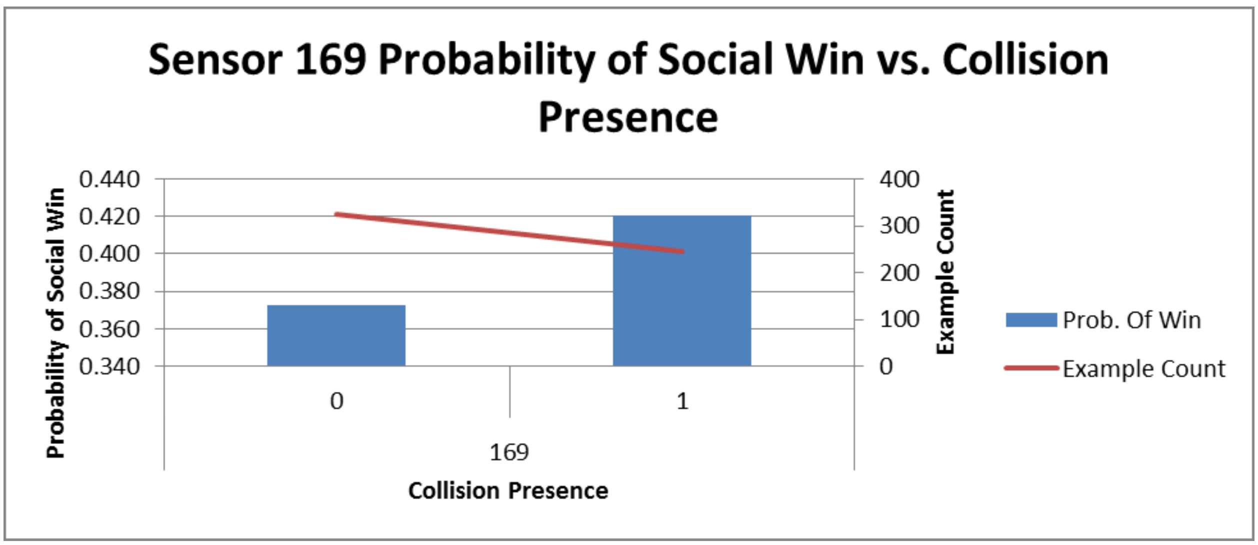
Row Labels	Prob. Of Win	Example Count
0	0.50	1427
1	0.39	570



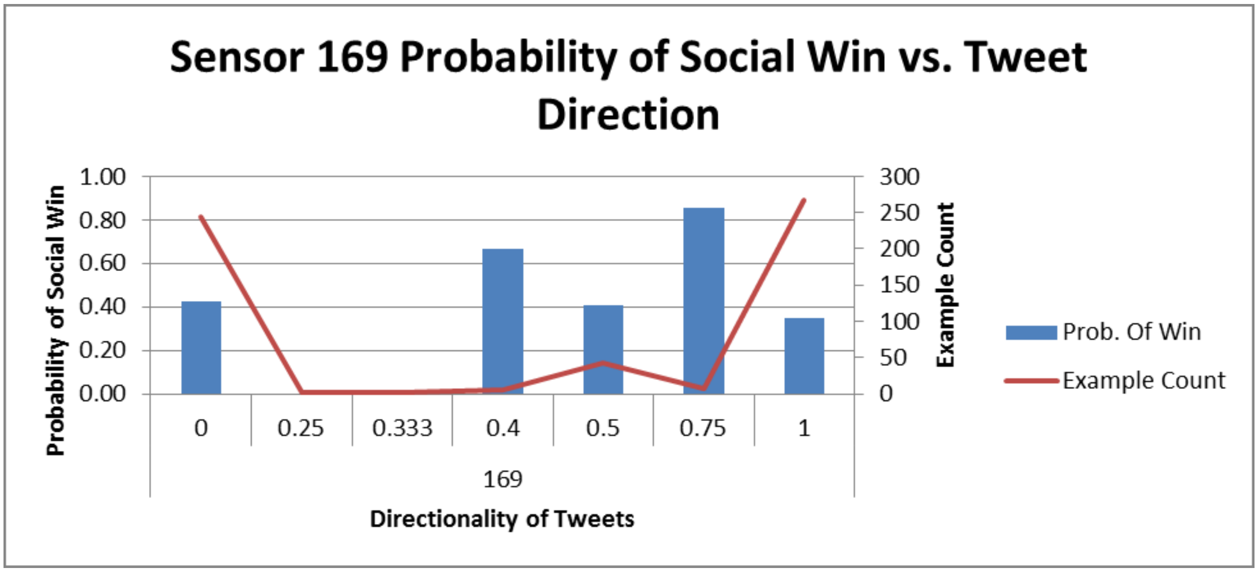
Row Labels	Prob. Of Win	Example Count
0	0.406	69
1	0.389	452
2	0.408	49



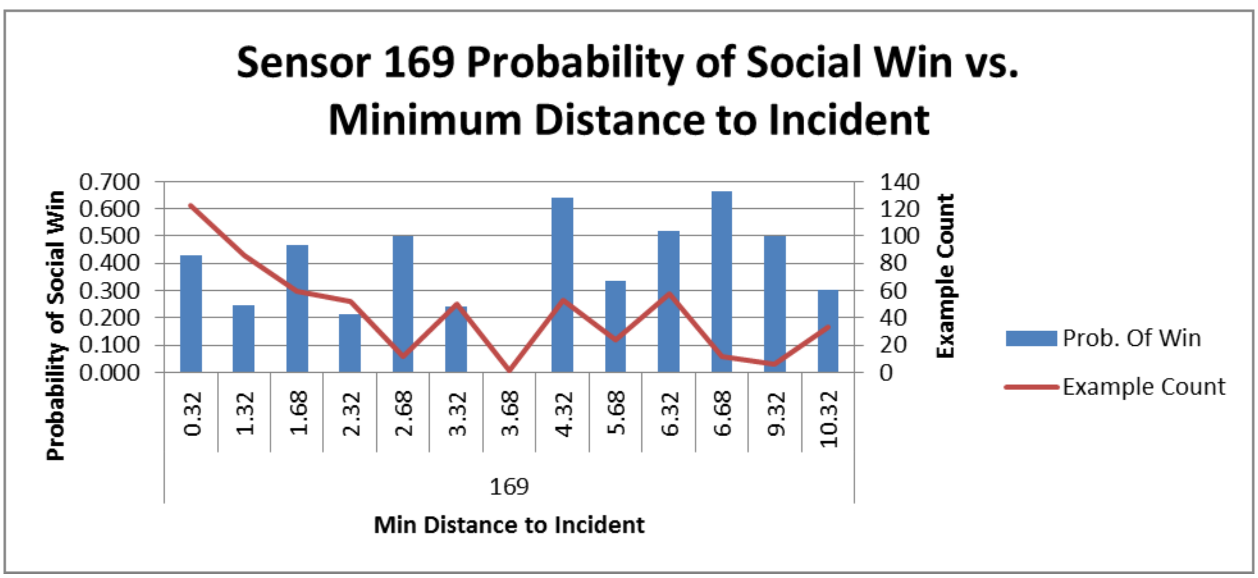
Row Labels	Prob. Of Win	Example Count
0	0.372	261
1	0.411	309



Row Labels	Prob. Of Win	Example Count
0	0.372	325
1	0.420	245

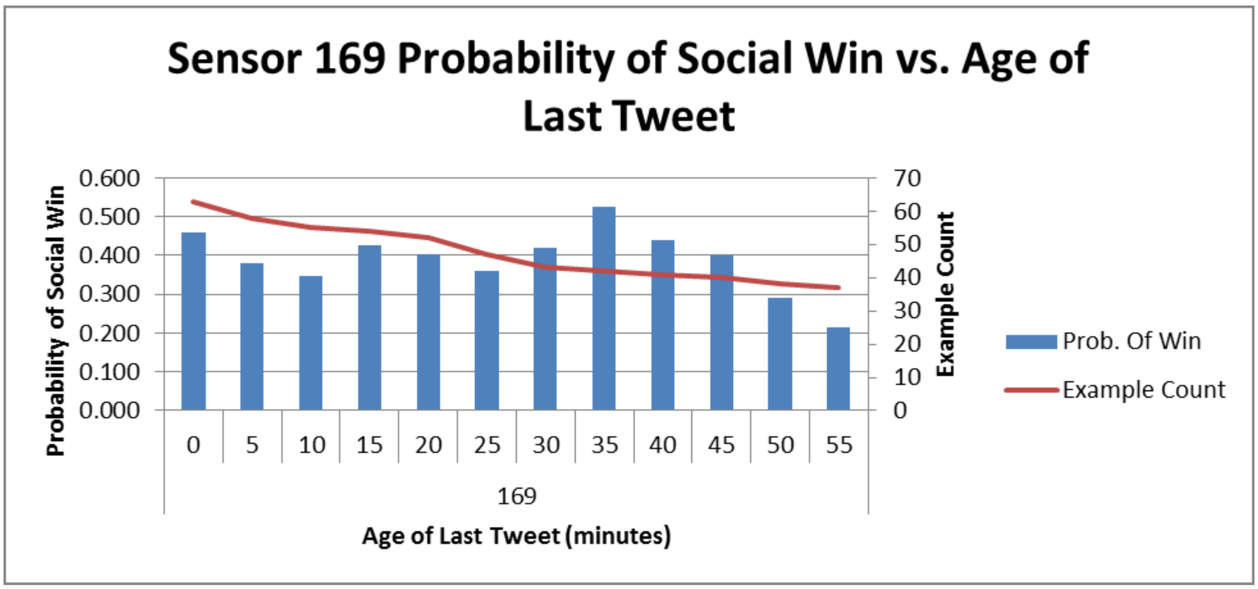


Row Labels	Prob. Of Win	Example Count
0	0.43	244
0.25	0.00	1
0.333	0.00	2
0.4	0.67	6
0.5	0.40	42
0.75	0.86	7
1	0.35	268



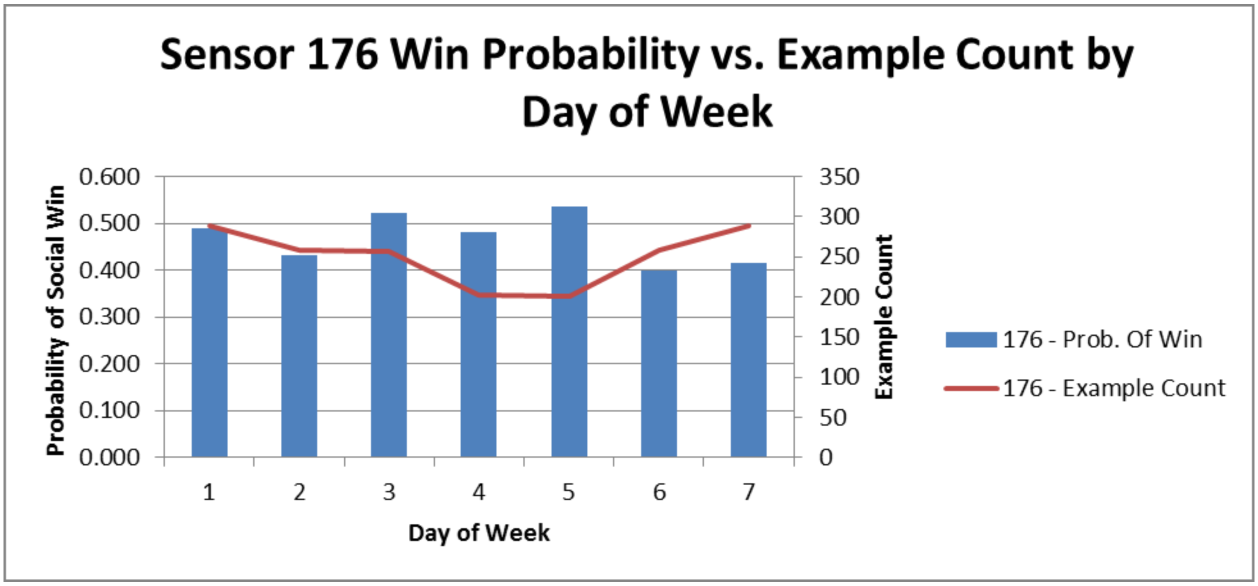
Row Labels	Prob. Of Win	Example Count
0.32	0.431	123

1.32	0.244	86
1.68	0.467	60
2.32	0.212	52
2.68	0.500	12
3.32	0.240	50
3.68	0.000	1
4.32	0.642	53
5.68	0.333	24
6.32	0.517	58
6.68	0.667	12
9.32	0.500	6
10.32	0.303	33

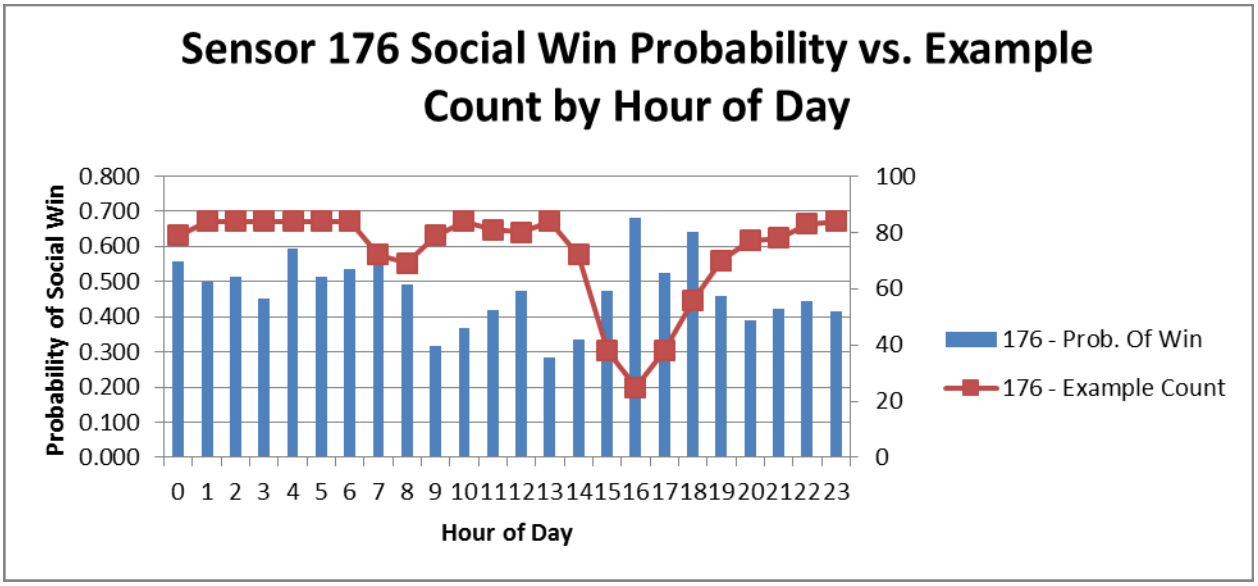


Row Labels	Prob. Of Win	Example Count
0	0.460	63
5	0.379	58
10	0.345	55
15	0.426	54
20	0.404	52
25	0.362	47
30	0.419	43
35	0.524	42
40	0.439	41
45	0.400	40
50	0.289	38
55	0.216	37

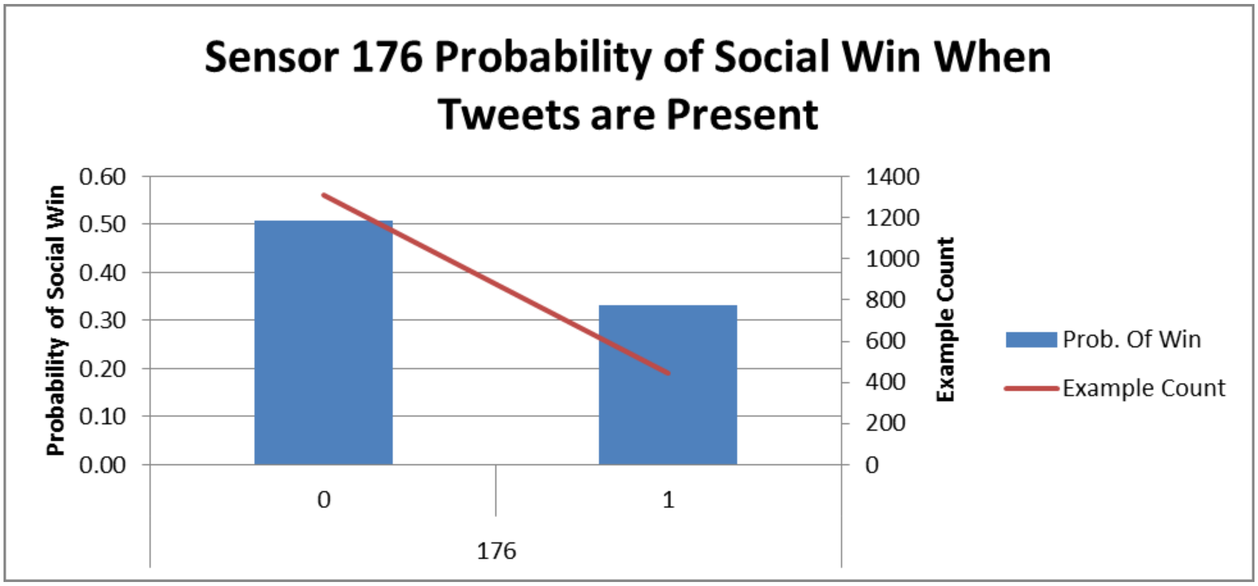
Sensor 176



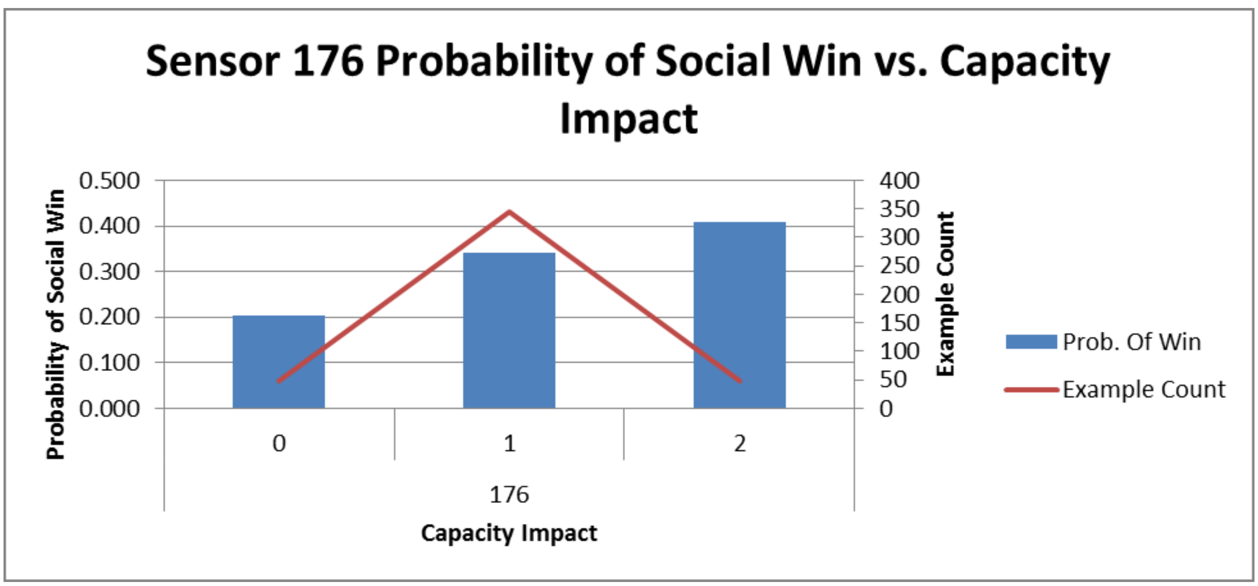
Row Labels	Prob. Of Win	Example Count
1	0.490	288
2	0.432	259
3	0.521	257
4	0.480	202
5	0.535	200
6	0.398	259
7	0.417	288



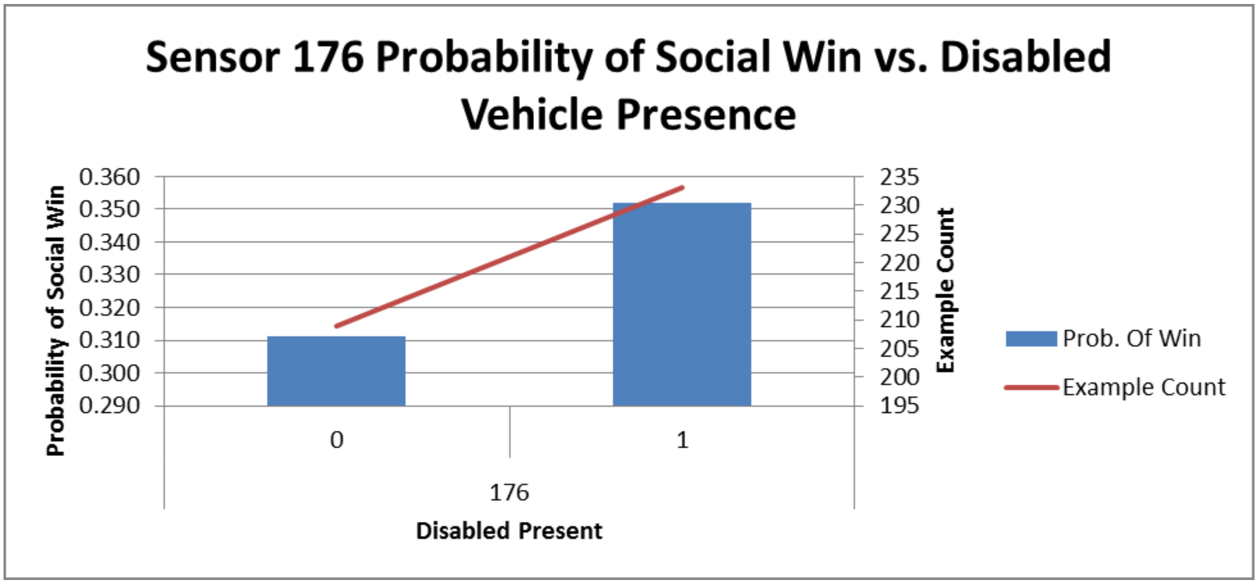
Row Labels	Prob. Of Win	Example Count
0	0.557	79
1	0.500	84
2	0.512	84
3	0.452	84
4	0.595	84
5	0.512	84
6	0.536	84
7	0.569	72
8	0.493	69
9	0.316	79
10	0.369	84
11	0.420	81
12	0.475	80
13	0.286	84
14	0.333	72
15	0.474	38
16	0.680	25
17	0.526	38
18	0.643	56
19	0.457	70
20	0.390	77
21	0.423	78
22	0.446	83
23	0.417	84



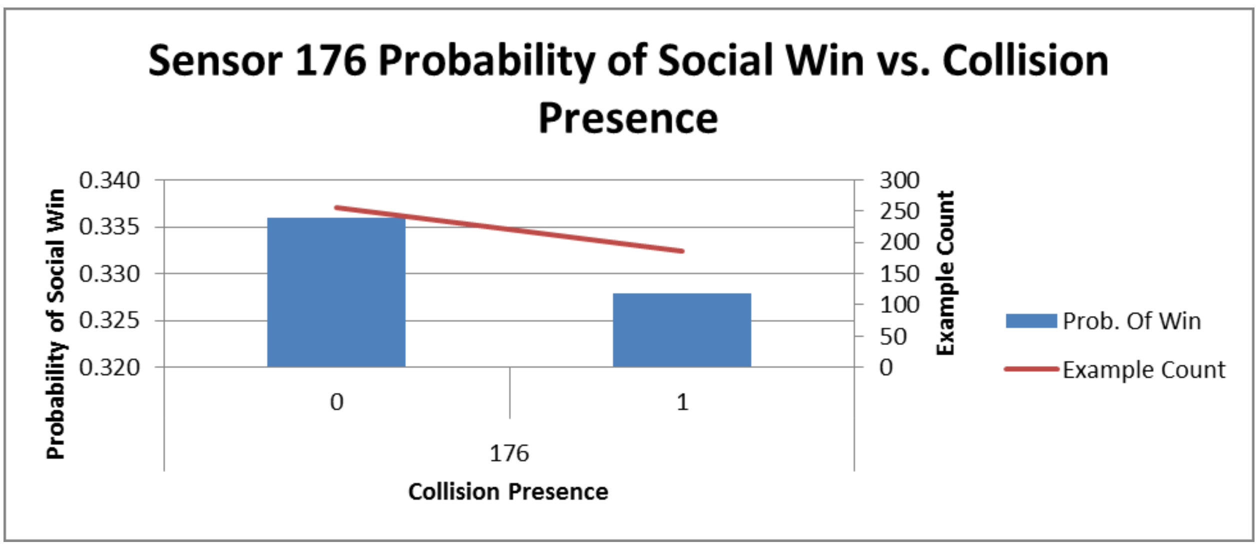
Row Labels	Prob. Of Win	Example Count
0	0.51	1311
1	0.33	442



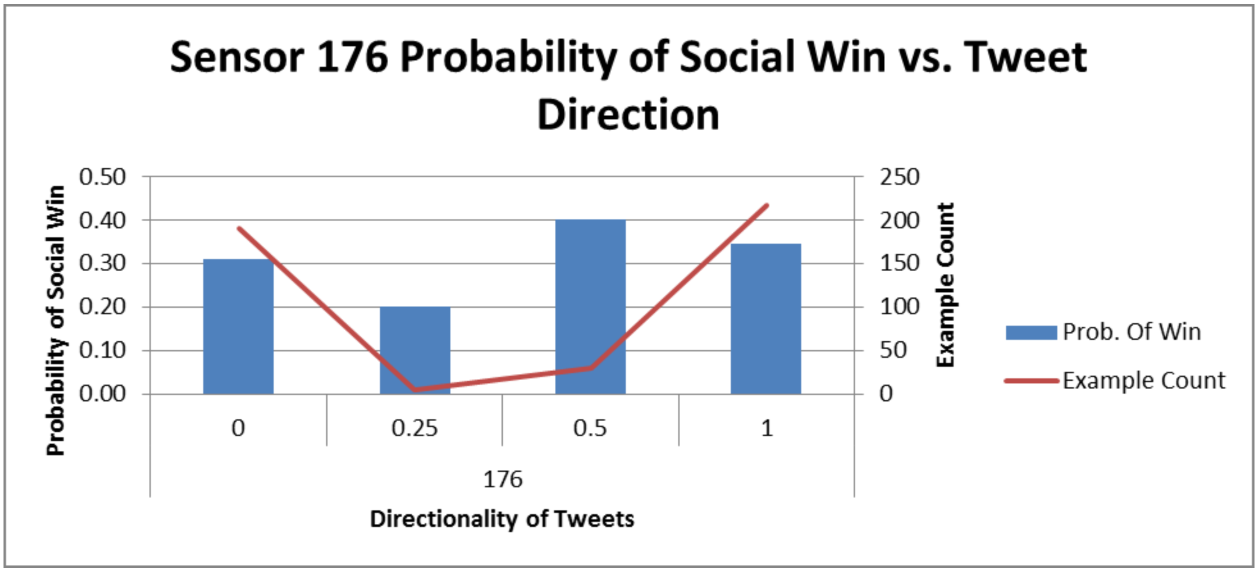
Row Labels	Prob. Of Win	Example Count
0	0.204	49
1	0.340	344
2	0.408	49



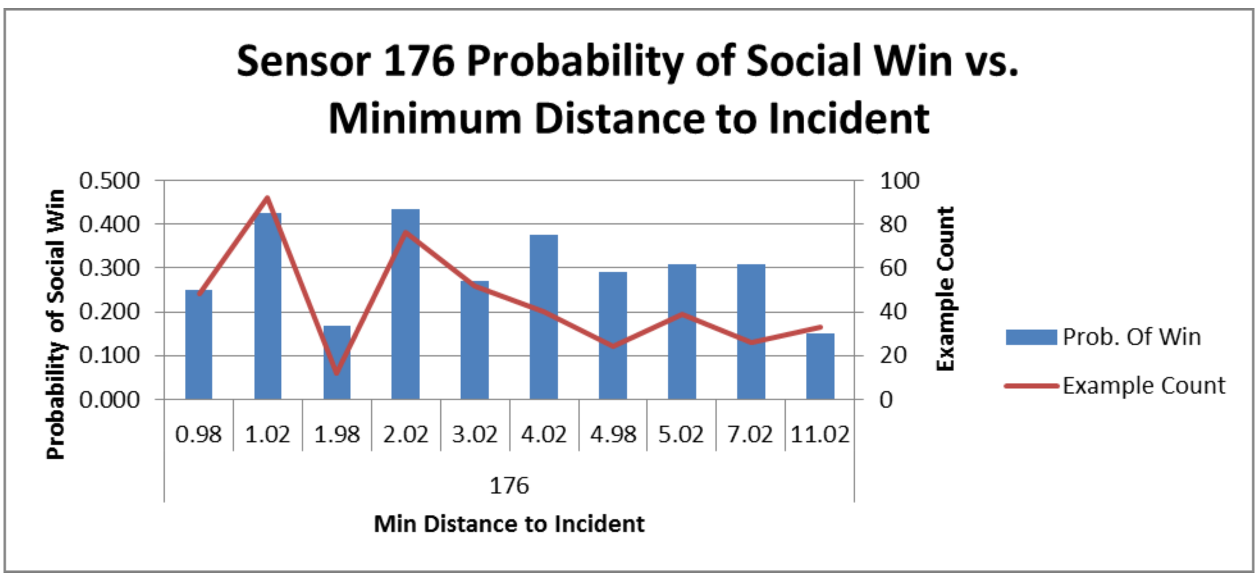
Row Labels	Prob. Of Win	Example Count
0	0.311	209
1	0.352	233



Row Labels	Prob. Of Win	Example Count
0	0.336	256
1	0.328	186

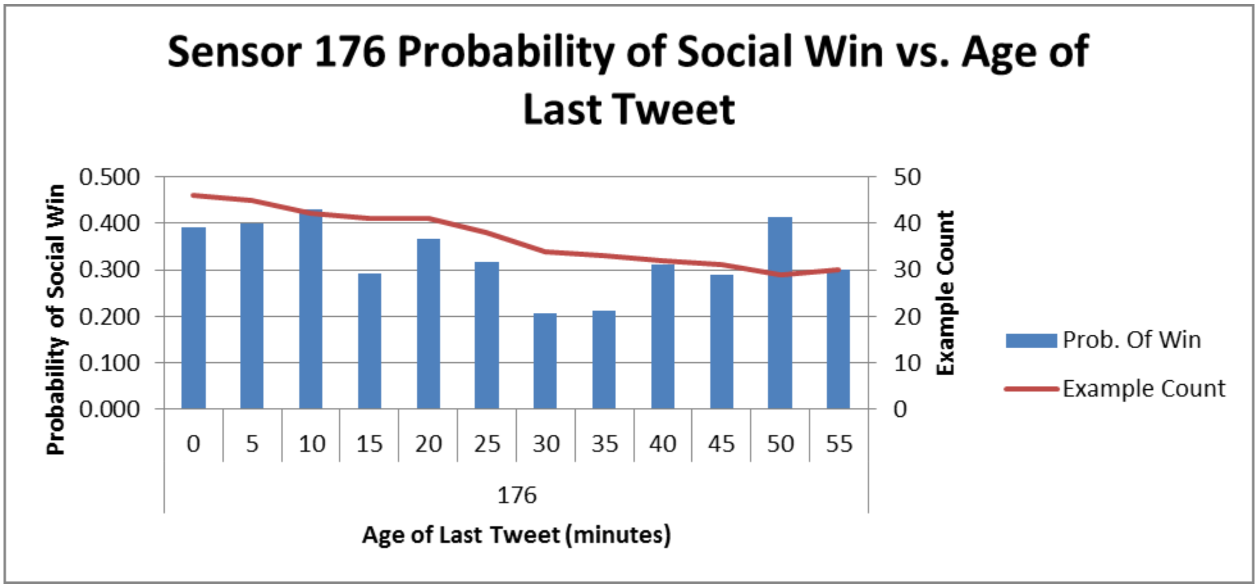


Row Labels	Prob. Of Win	Example Count
0	0.31	190
0.25	0.20	5
0.5	0.40	30
1	0.35	217



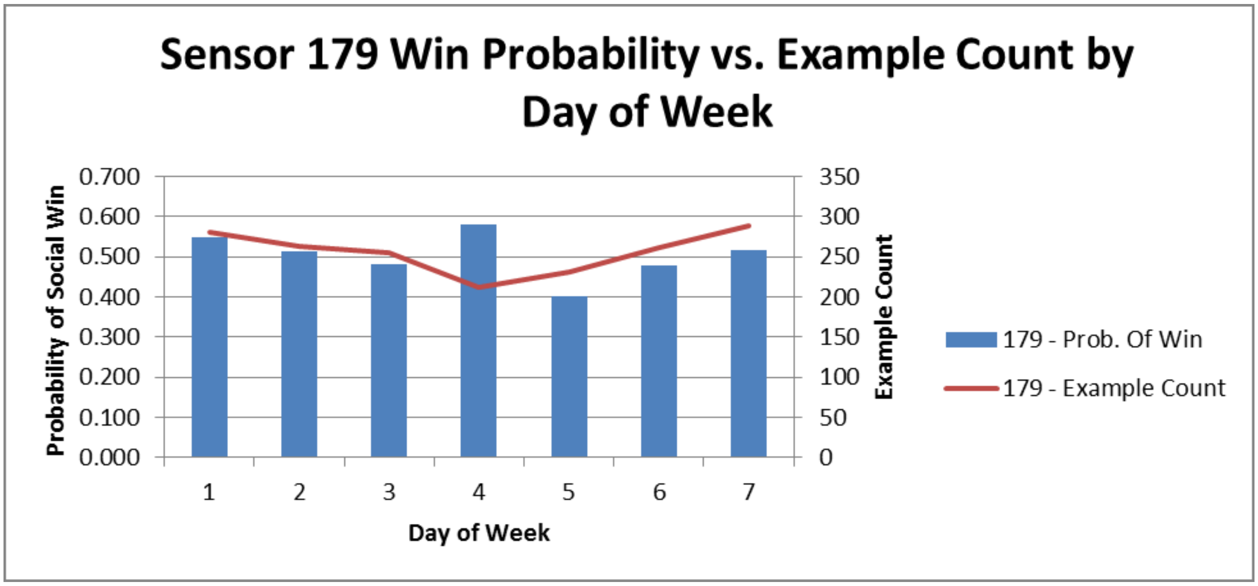
Row Labels	Prob. Of Win	Example Count
0.98	0.250	48
1.02	0.424	92
1.98	0.167	12
2.02	0.434	76

3.02	0.269	52
4.02	0.375	40
4.98	0.292	24
5.02	0.308	39
7.02	0.308	26
11.02	0.152	33

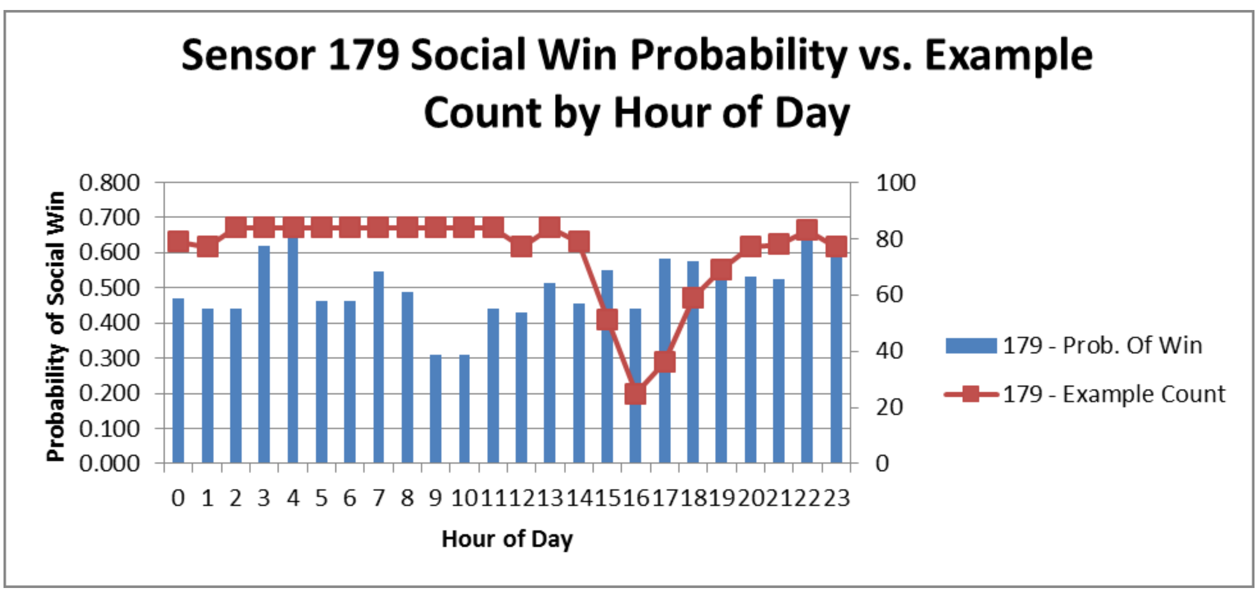


Row Labels	Prob. Of Win	Example Count
0	0.391	46
5	0.400	45
10	0.429	42
15	0.293	41
20	0.366	41
25	0.316	38
30	0.206	34
35	0.212	33
40	0.313	32
45	0.290	31
50	0.414	29
55	0.300	30

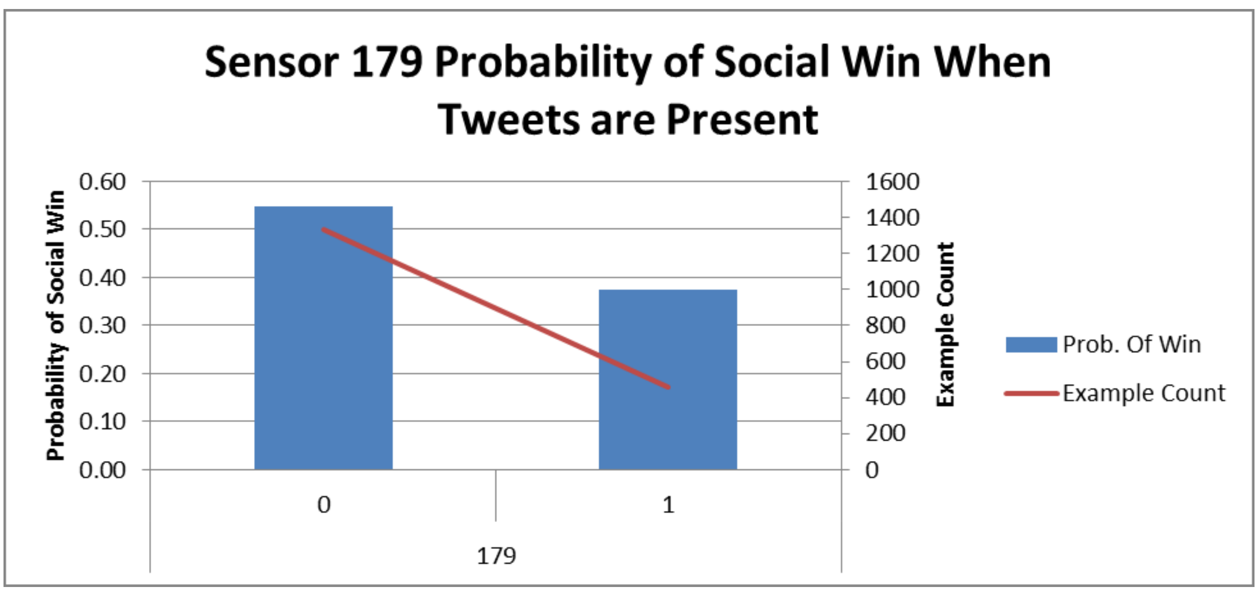
Sensor 179



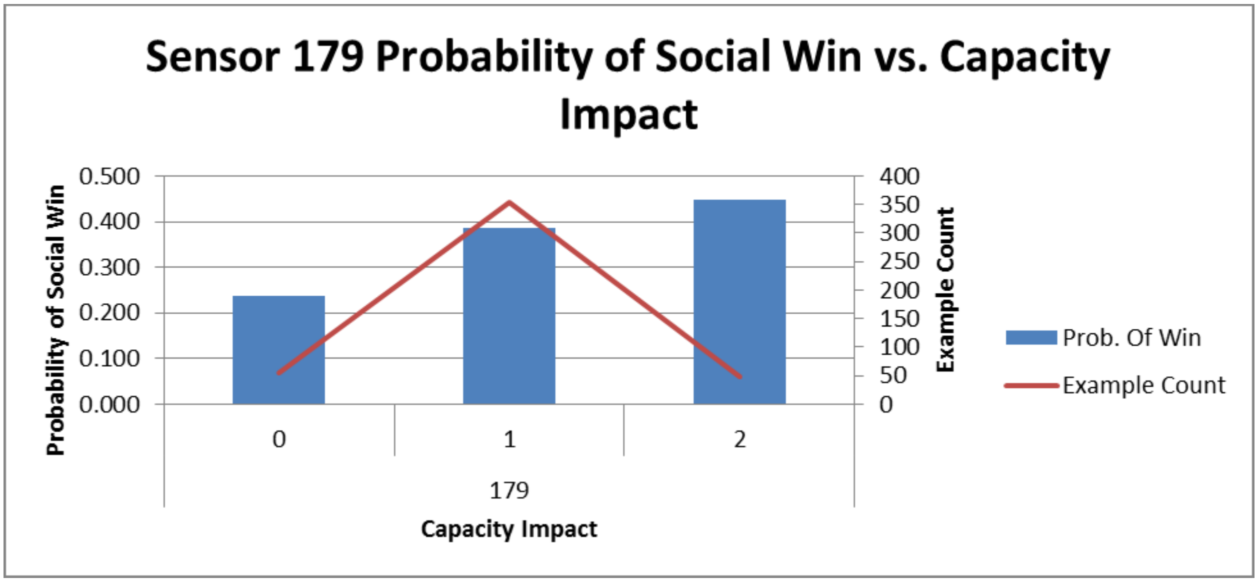
Row Labels	Prob. Of Win	Example Count
1	0.548	281
2	0.513	263
3	0.482	255
4	0.580	212
5	0.403	231
6	0.479	261
7	0.517	288



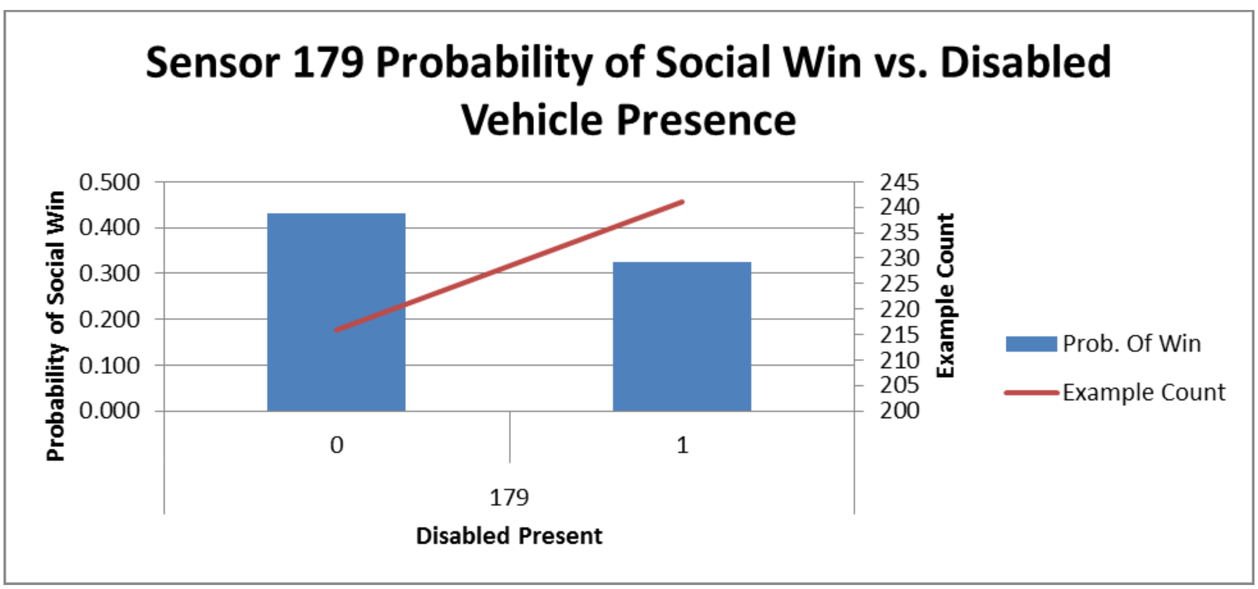
Row Labels	Prob. Of Win	Example Count
0	0.468	79
1	0.442	77
2	0.440	84
3	0.619	84
4	0.679	84
5	0.464	84
6	0.464	84
7	0.548	84
8	0.488	84
9	0.310	84
10	0.310	84
11	0.440	84
12	0.429	77
13	0.512	84
14	0.456	79
15	0.549	51
16	0.440	25
17	0.583	36
18	0.576	59
19	0.565	69
20	0.532	77
21	0.526	78
22	0.687	83
23	0.610	77



Row Labels	Prob. Of Win	Example Count
0	0.55	1334
1	0.37	457

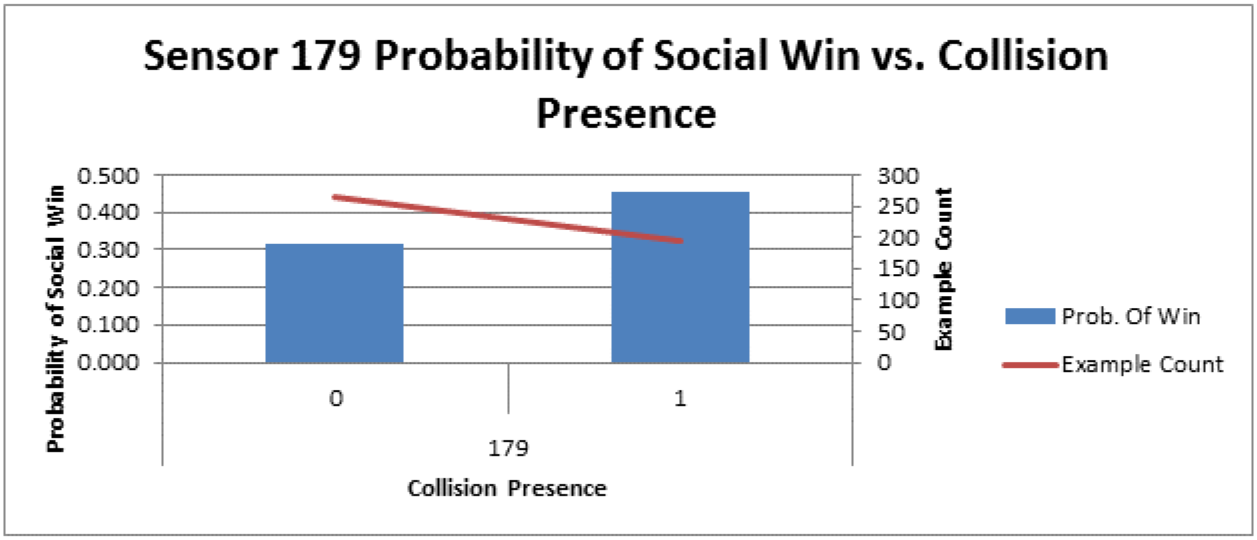


Row Labels	Prob. Of Win	Example Count
0	0.236	55
1	0.385	353
2	0.449	49

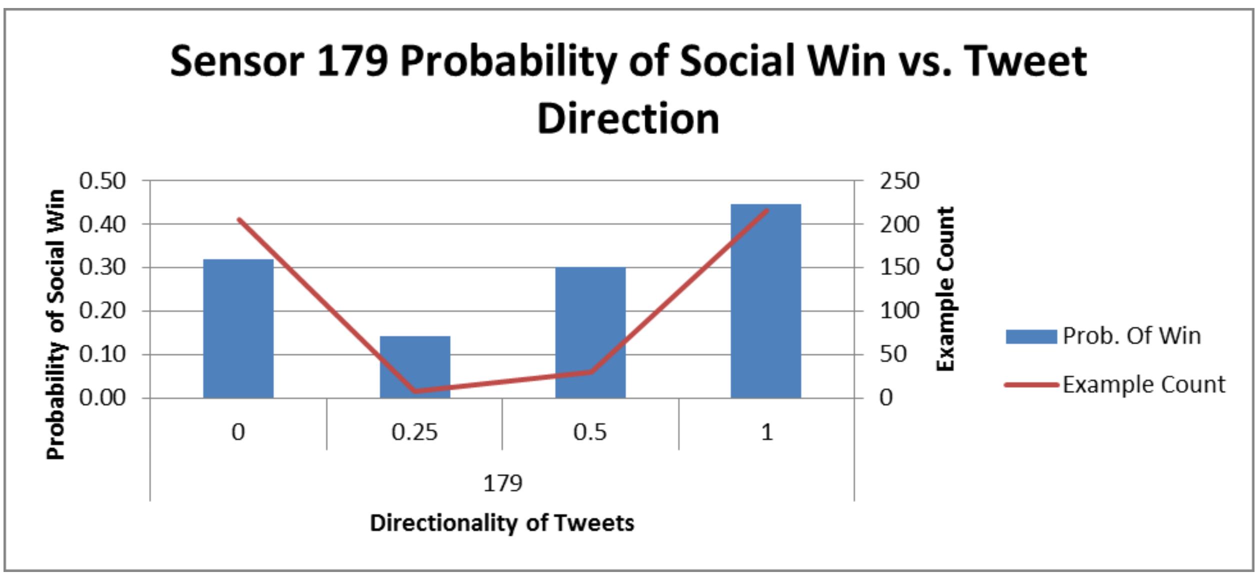


Row Labels	Prob. Of Win	Example Count

0	0.431	216
1	0.324	241

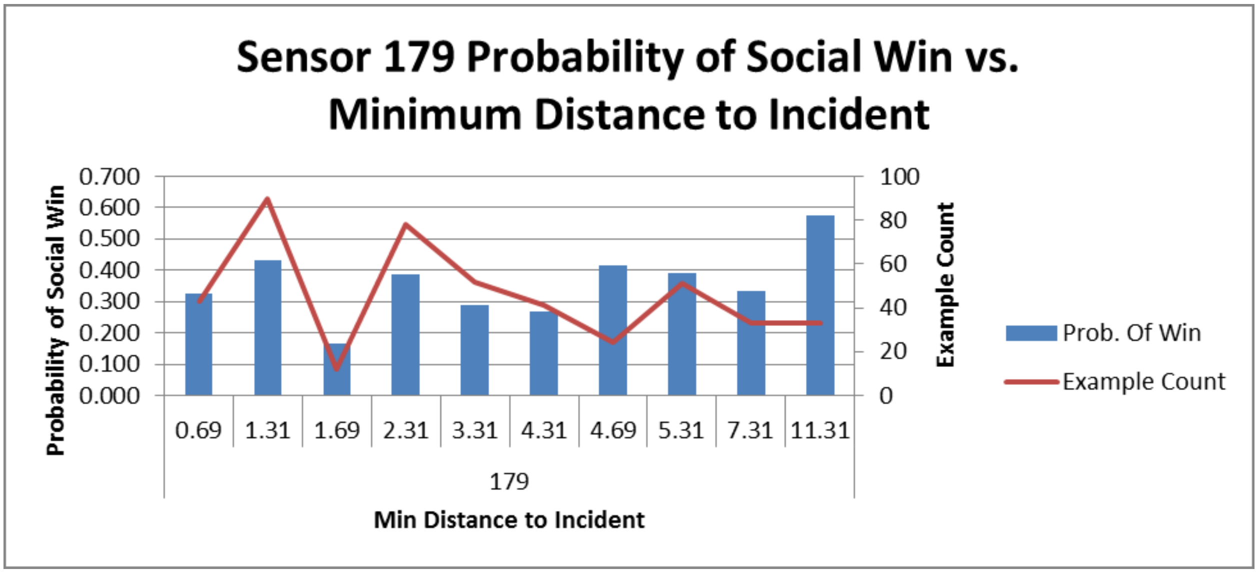


Row Labels	Prob. Of Win	Example Count
0	0.314	264
1	0.456	193

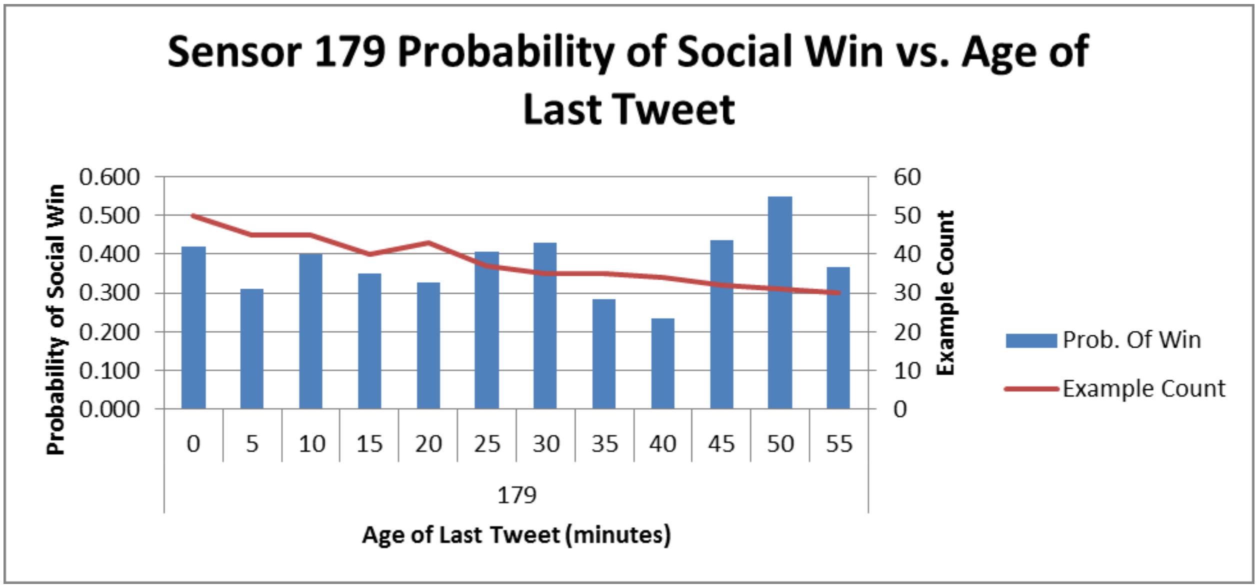


Row Labels	Prob. Of Win	Example Count
0	0.32	205
0.25	0.14	7
0.5	0.30	30

1	0.45	215

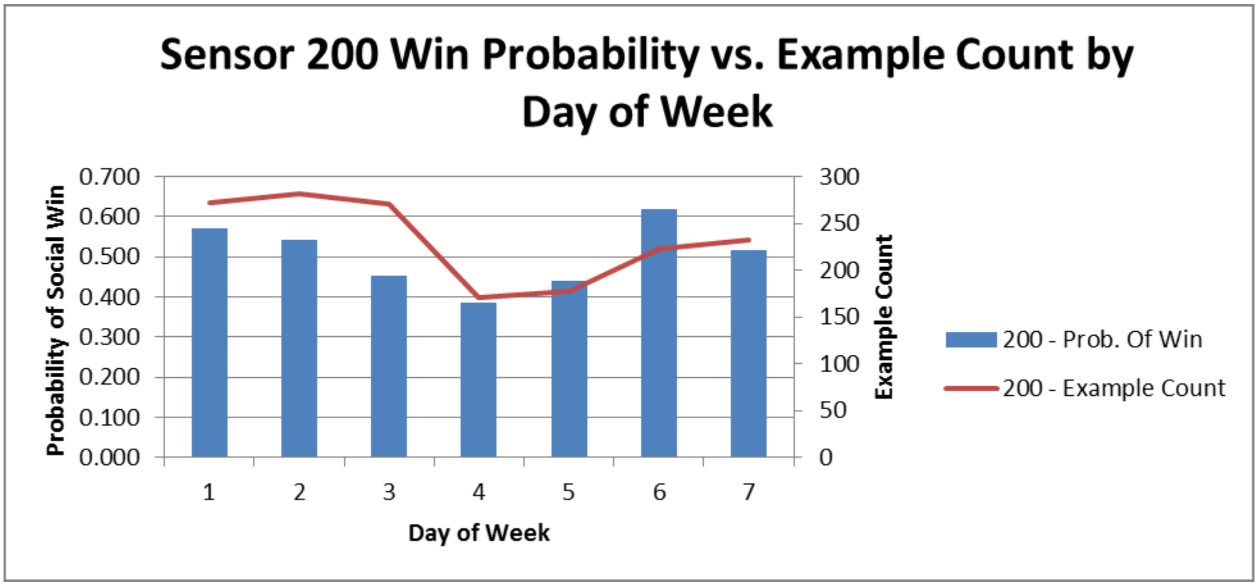


Row Labels	Prob. Of Win	Example Count
0.69	0.326	43
1.31	0.433	90
1.69	0.167	12
2.31	0.385	78
3.31	0.288	52
4.31	0.268	41
4.69	0.417	24
5.31	0.392	51
7.31	0.333	33
11.31	0.576	33

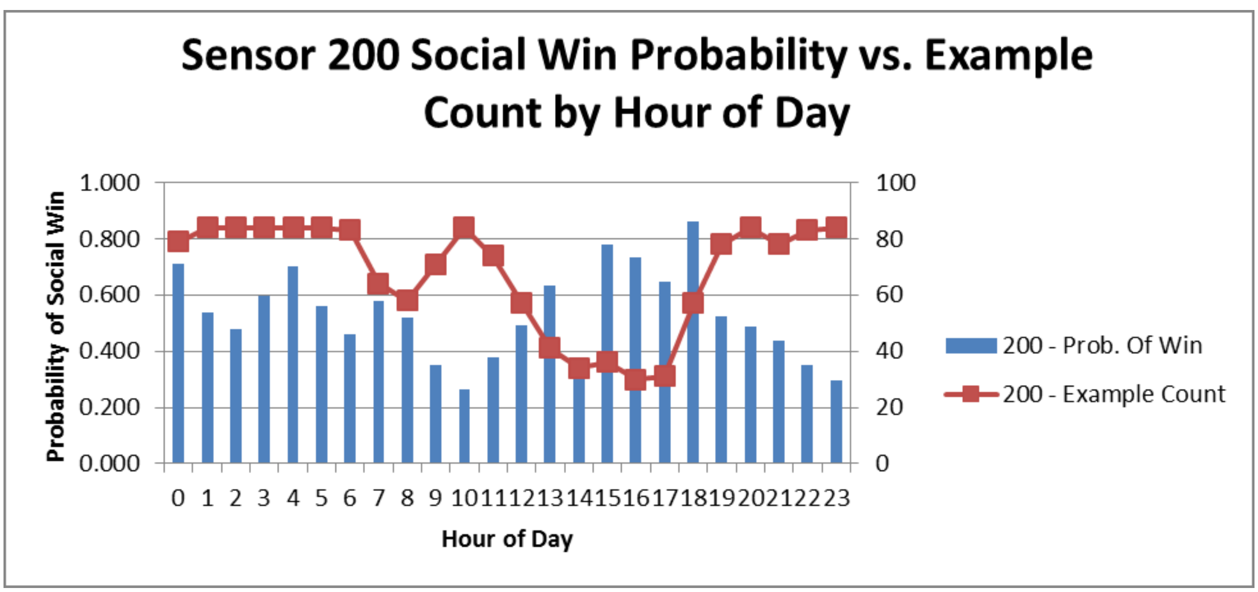


Row Labels	Prob. Of Win	Example Count
0	0.420	50
5	0.311	45
10	0.400	45
15	0.350	40
20	0.326	43
25	0.405	37
30	0.429	35
35	0.286	35
40	0.235	34
45	0.438	32
50	0.548	31
55	0.367	30

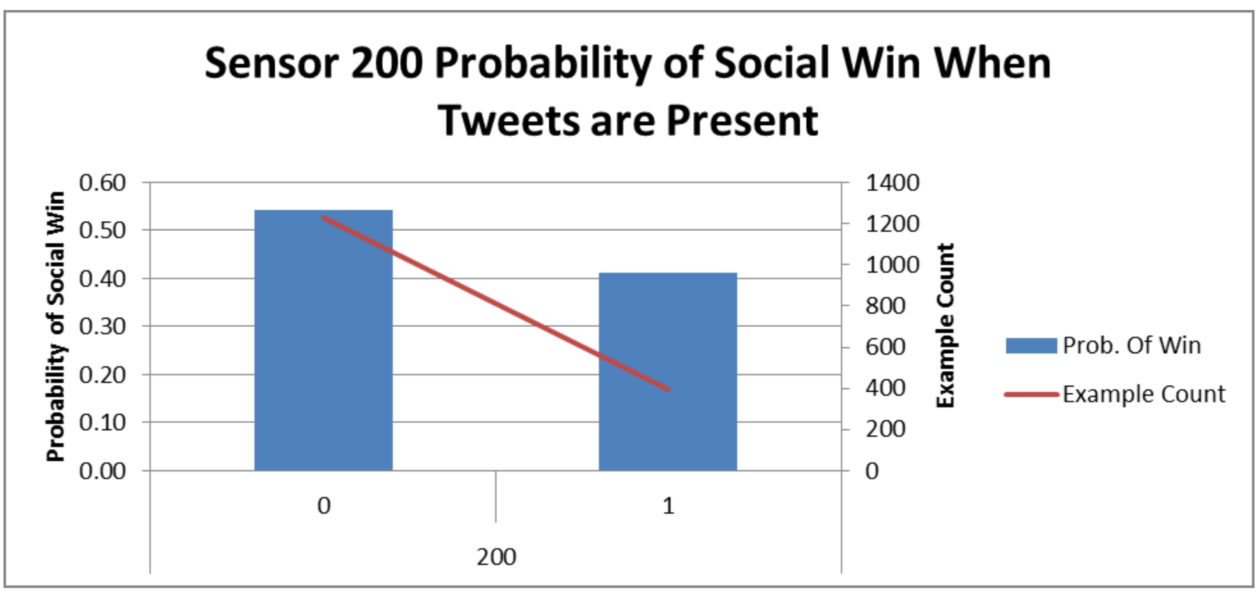
Sensor 200



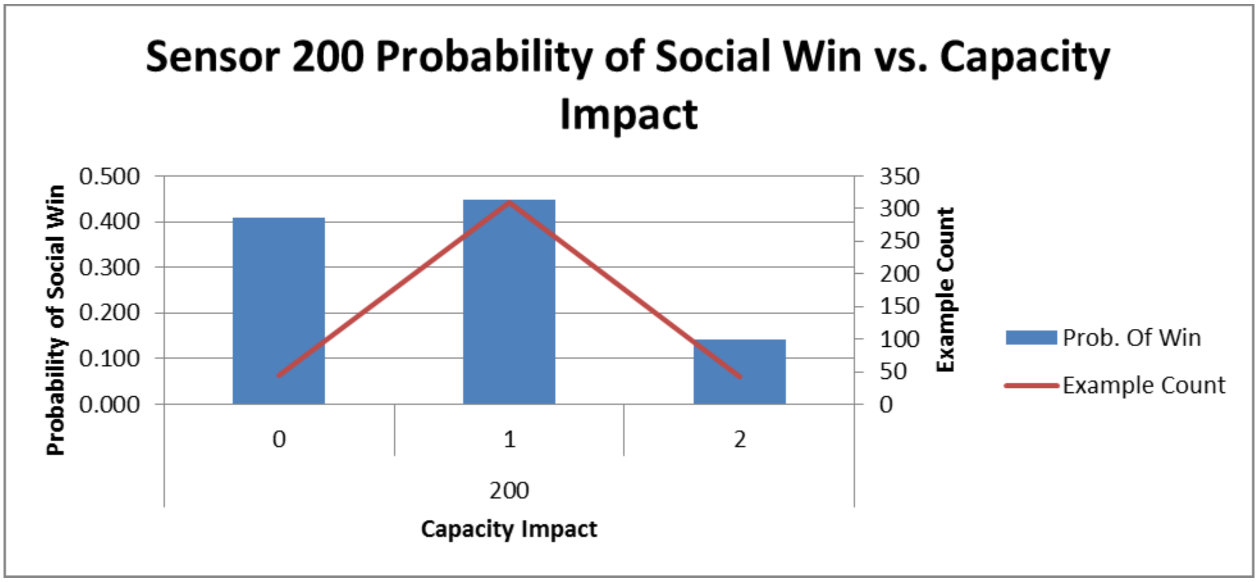
Row Labels	Prob. Of Win	Example Count
1	0.570	272
2	0.541	281
3	0.452	270
4	0.386	171
5	0.441	177
6	0.619	223
7	0.517	232



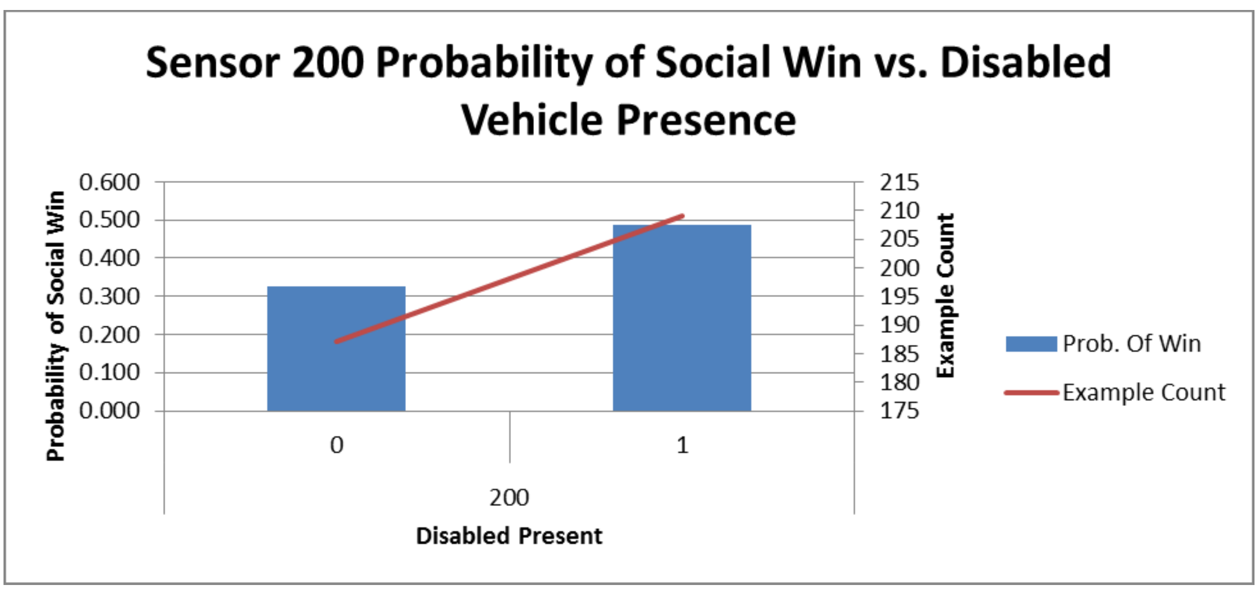
Row Labels	Prob. Of Win	Example Count
0	0.709	79
1	0.536	84
2	0.476	84
3	0.595	84
4	0.702	84
5	0.560	84
6	0.458	83
7	0.578	64
8	0.517	58
9	0.352	71
10	0.262	84
11	0.378	74
12	0.491	57
13	0.634	41
14	0.324	34
15	0.778	36
16	0.733	30
17	0.645	31
18	0.860	57
19	0.526	78
20	0.488	84
21	0.436	78
22	0.349	83
23	0.298	84



Row Labels	Prob. Of Win	Example Count
0	0.54	1230
1	0.41	396

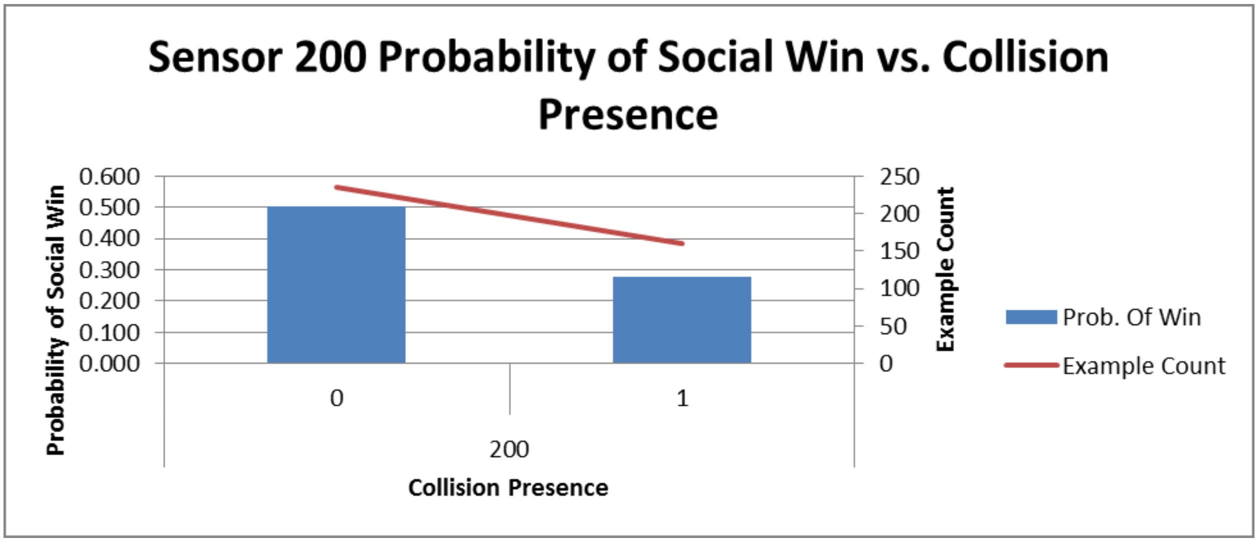


Row Labels	Prob. Of Win	Example Count
0	0.409	44
1	0.448	310
2	0.143	42

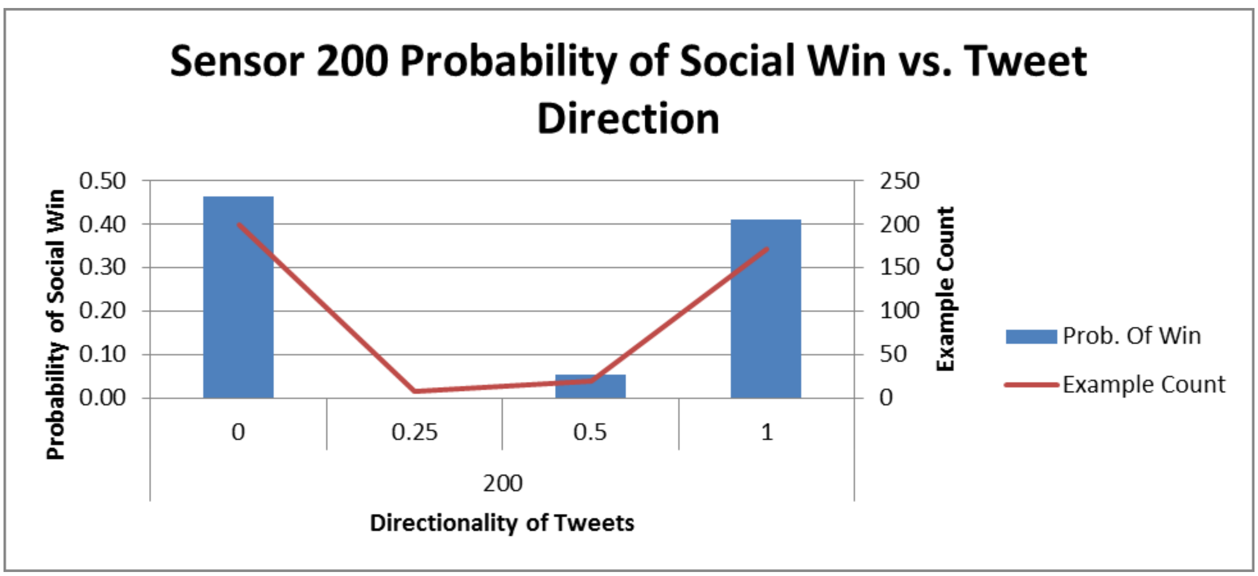


Row Labels	Prob. Of Win	Example Count

0	0.326	187
1	0.488	209

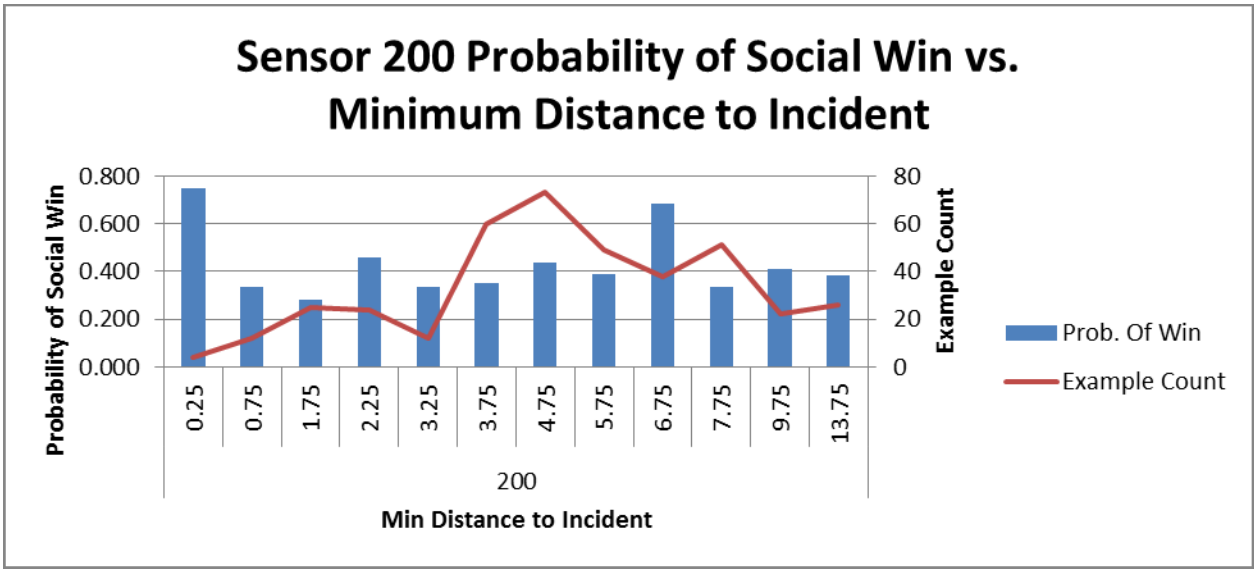


Row Labels	Prob. Of Win	Example Count
0	0.502	235
1	0.280	161

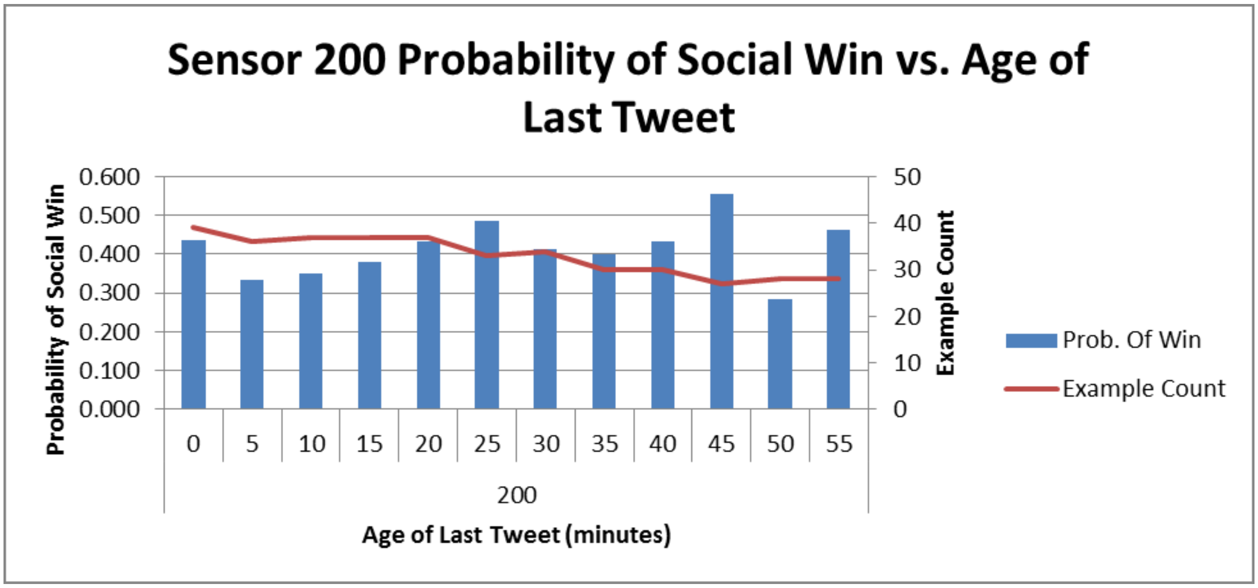


Row Labels	Prob. Of Win	Example Count
0	0.46	199
0.25	0.00	7
0.5	0.05	19

1	0.41	171

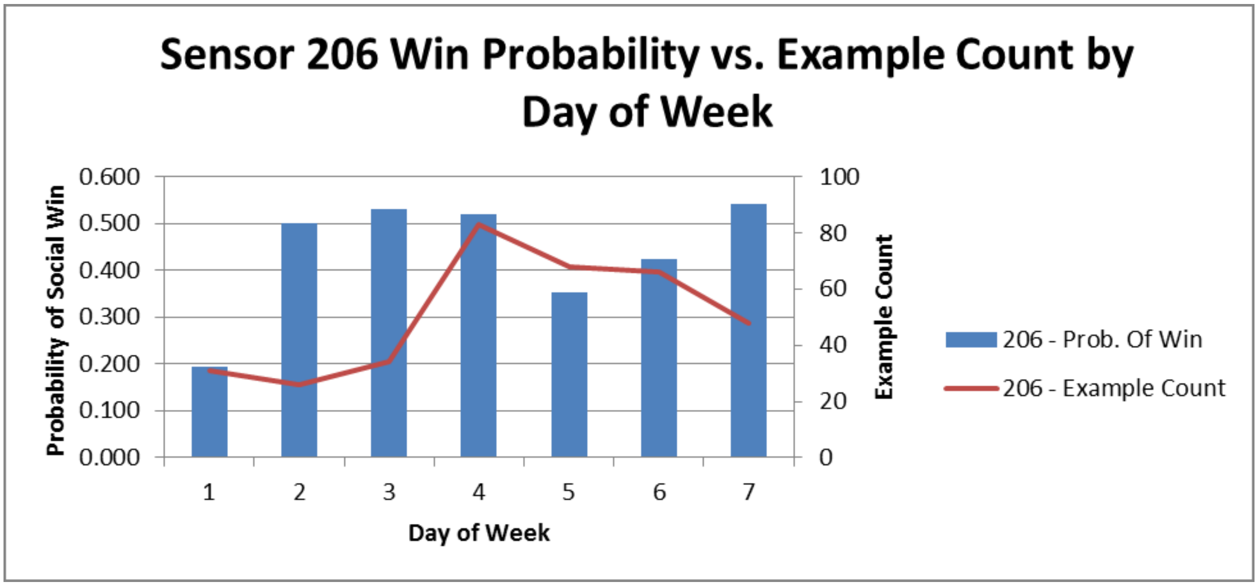


Row Labels	Prob. Of Win	Example Count
0.25	0.750	4
0.75	0.333	12
1.75	0.280	25
2.25	0.458	24
3.25	0.333	12
3.75	0.350	60
4.75	0.438	73
5.75	0.388	49
6.75	0.684	38
7.75	0.333	51
9.75	0.409	22
13.75	0.385	26

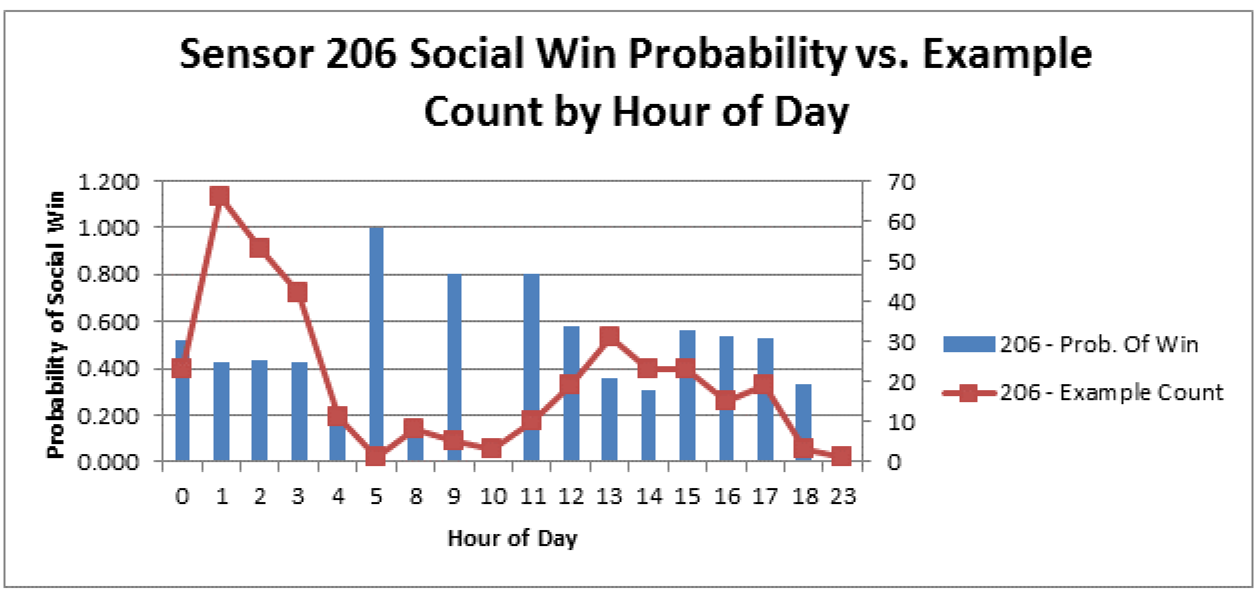


Row Labels	Prob. Of Win	Example Count
0	0.436	39
5	0.333	36
10	0.351	37
15	0.378	37
20	0.432	37
25	0.485	33
30	0.412	34
35	0.400	30
40	0.433	30
45	0.556	27
50	0.286	28
55	0.464	28

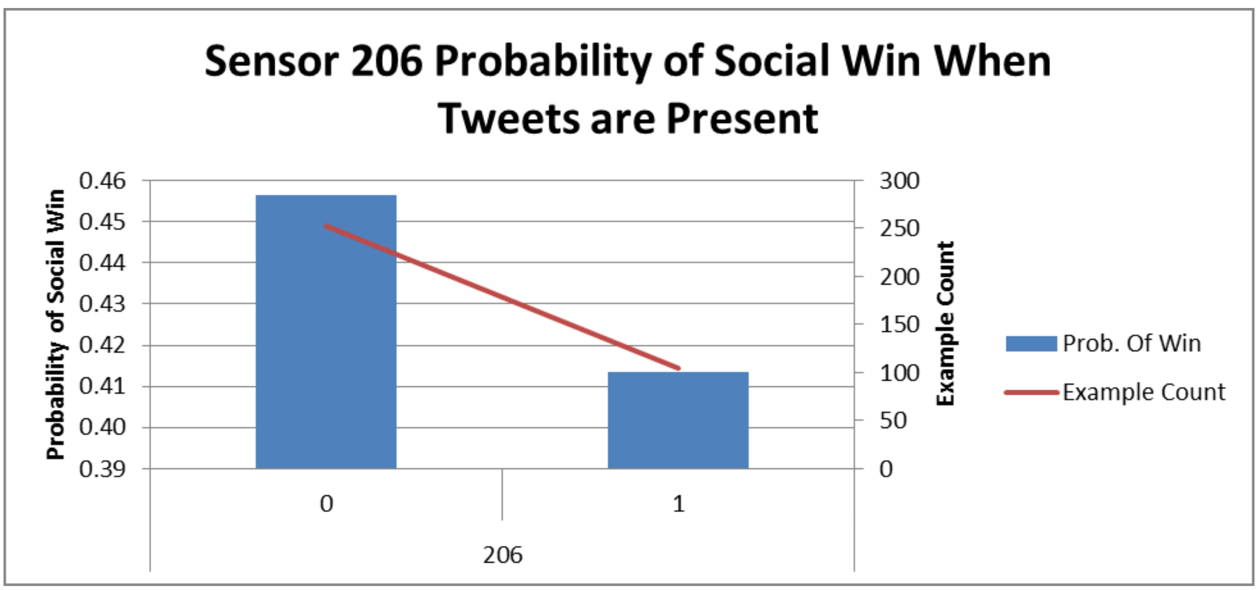
Sensor 206



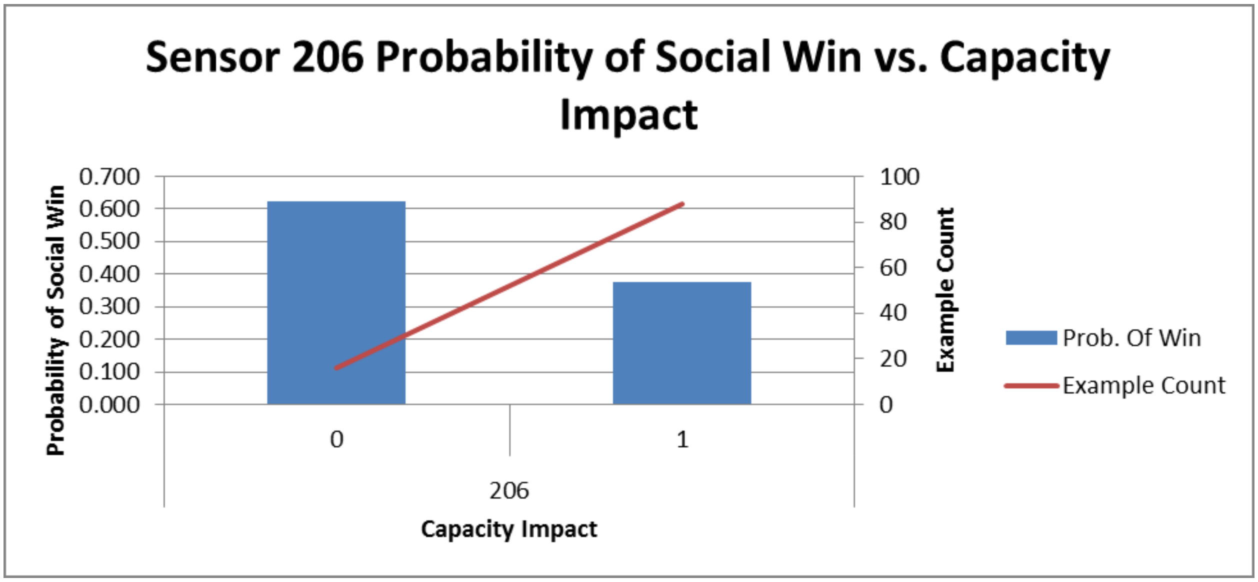
Row Labels	Prob. Of Win	Example Count
1	0.194	31
2	0.500	26
3	0.529	34
4	0.518	83
5	0.353	68
6	0.424	66
7	0.542	48



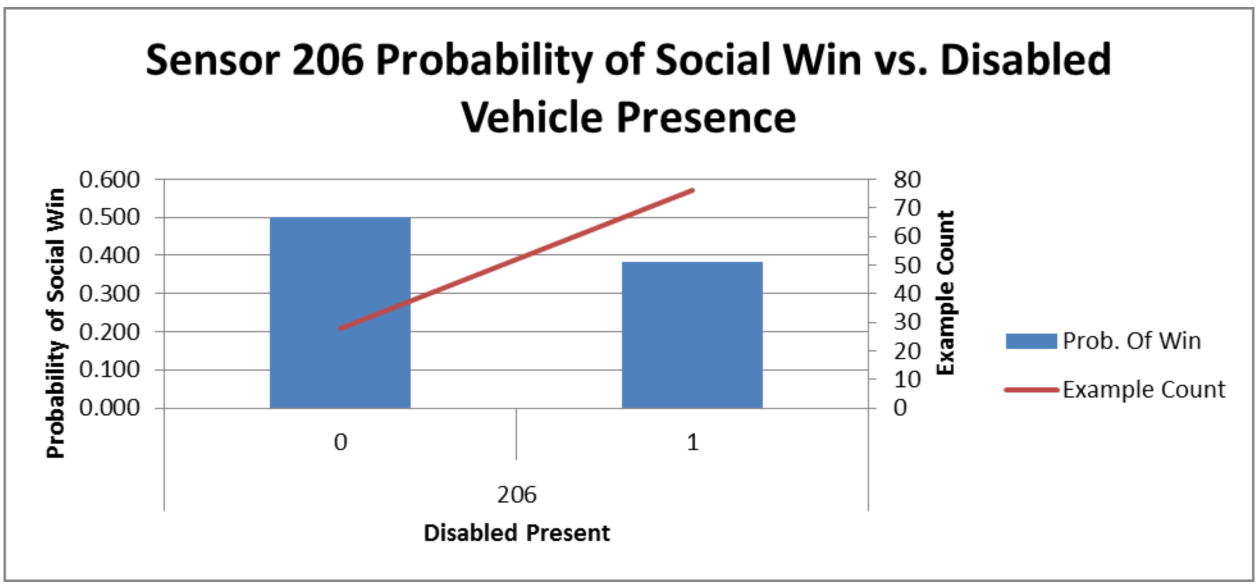
Row Labels	Prob. Of Win	Example Count
0	0.522	23
1	0.424	66
2	0.434	53
3	0.429	42
4	0.182	11
5	1.000	1
8	0.125	8
9	0.800	5
10	0.000	3
11	0.800	10
12	0.579	19
13	0.355	31
14	0.304	23
15	0.565	23
16	0.533	15
17	0.526	19
18	0.333	3
23	0.000	1



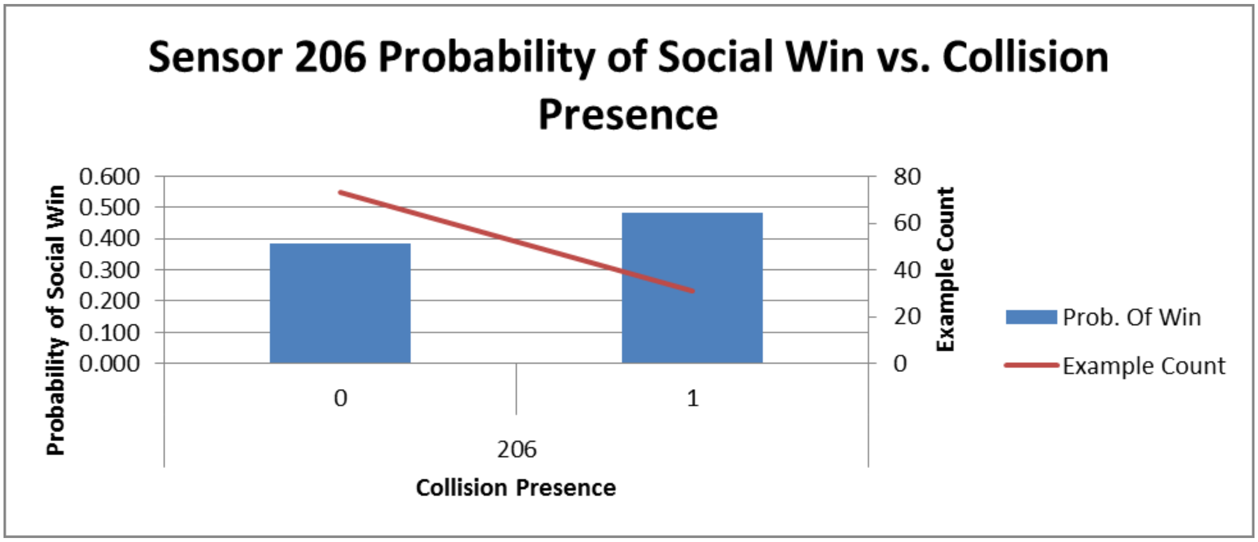
Row Labels	Prob. Of Win	Example Count
0	0.46	252
1	0.41	104



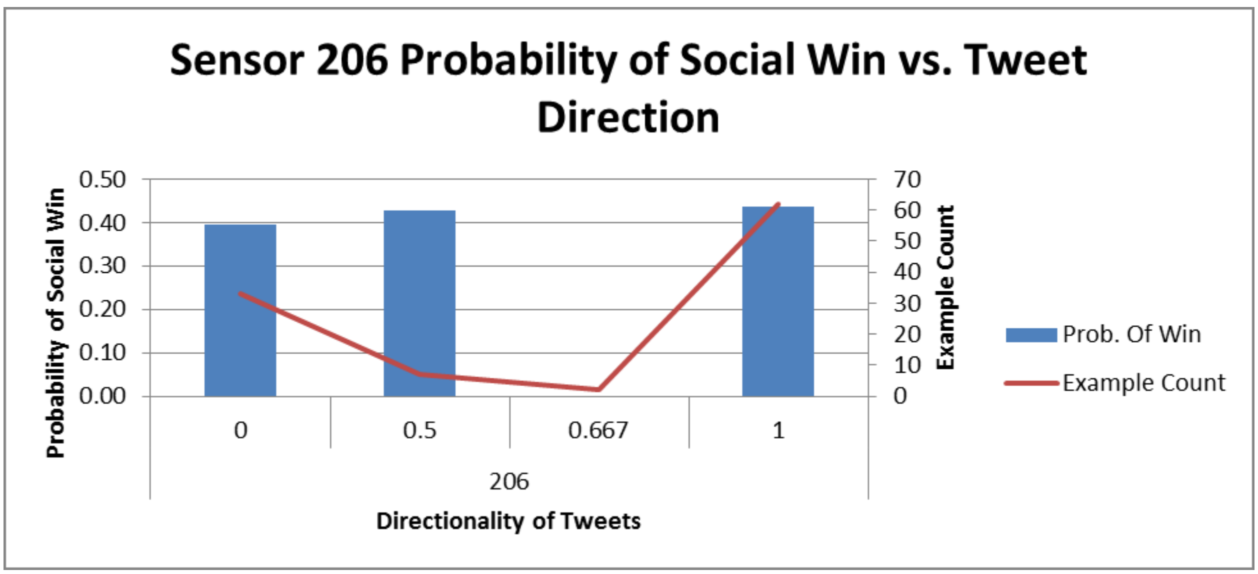
Row Labels	Prob. Of Win	Example Count
0	0.625	16
1	0.375	88



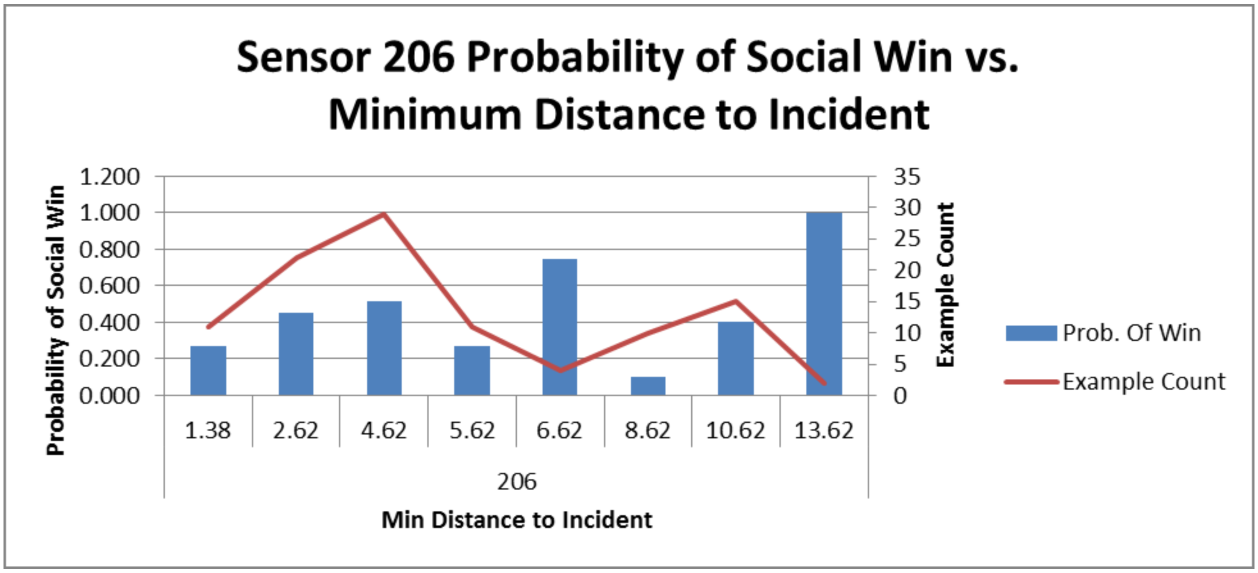
Row Labels	Prob. Of Win	Example Count
0	0.500	28
1	0.382	76



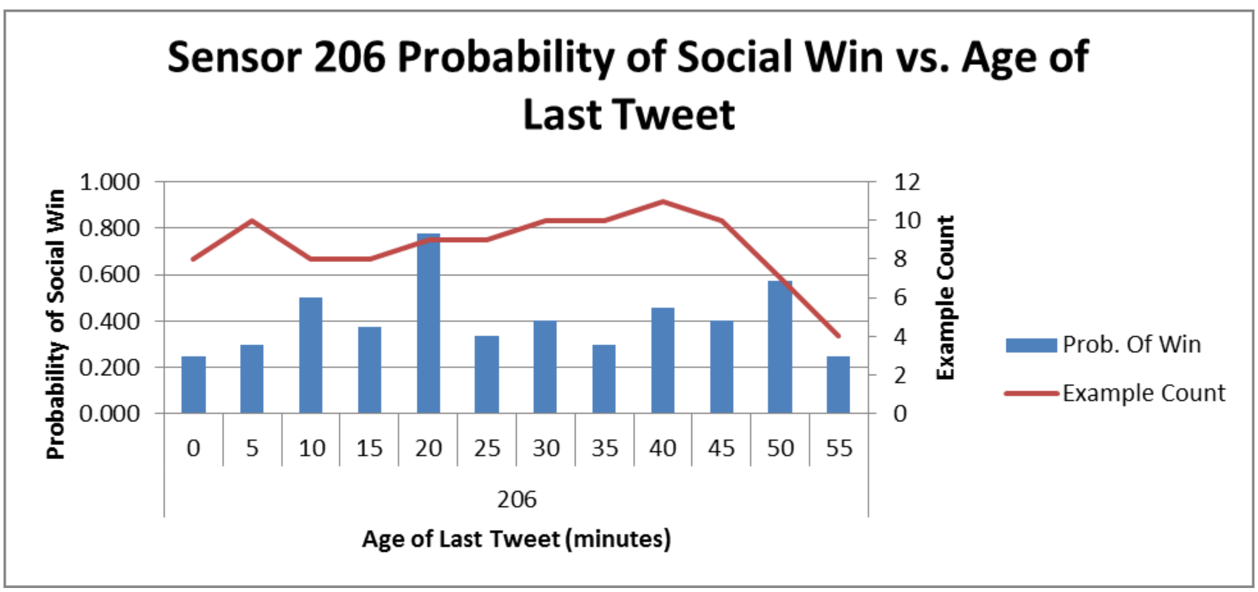
Row Labels	Prob. Of Win	Example Count
0	0.384	73
1	0.484	31



Row Labels	Prob. Of Win	Example Count
0	0.39	33
0.5	0.43	7
0.667	0.00	2
1	0.44	62



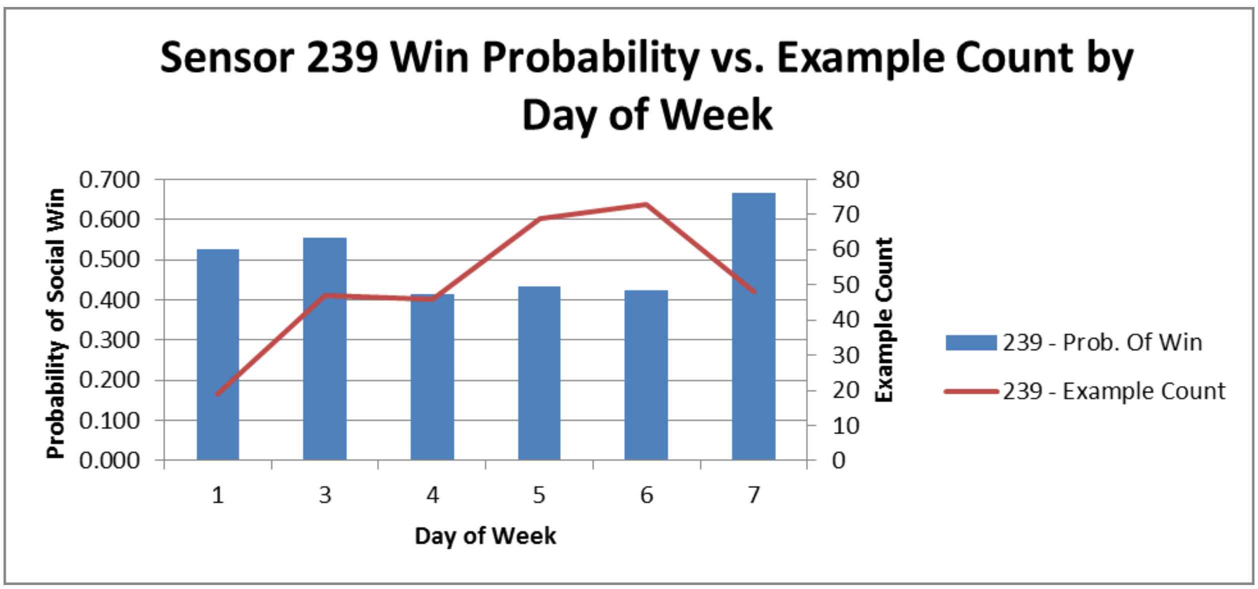
Row Labels	Prob. Of Win	Example Count
1.38	0.273	11
2.62	0.455	22
4.62	0.517	29
5.62	0.273	11
6.62	0.750	4
8.62	0.100	10
10.62	0.400	15
13.62	1.000	2



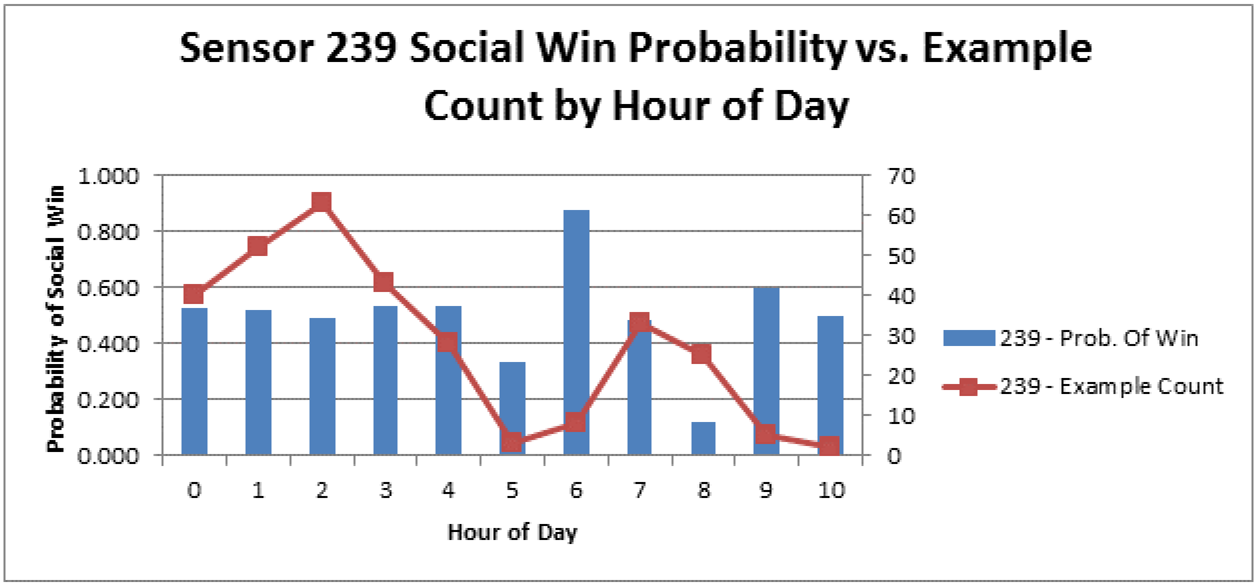
Row Labels	Prob. Of Win	Example Count

0	0.250	8
5	0.300	10
10	0.500	8
15	0.375	8
20	0.778	9
25	0.333	9
30	0.400	10
35	0.300	10
40	0.455	11
45	0.400	10
50	0.571	7
55	0.250	4

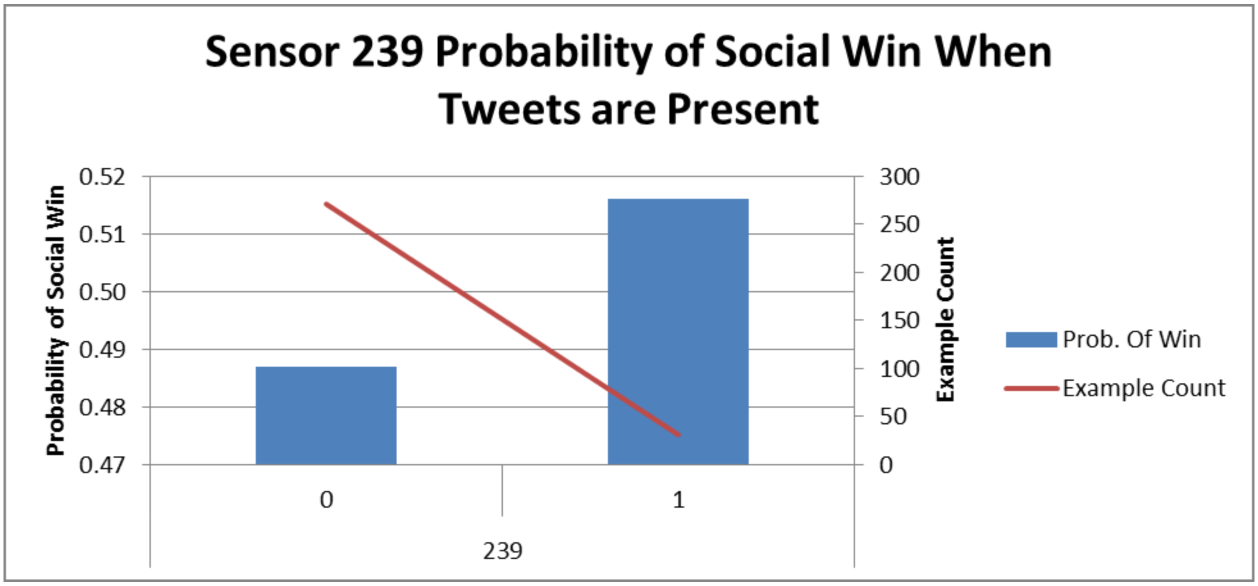
Sensor 239



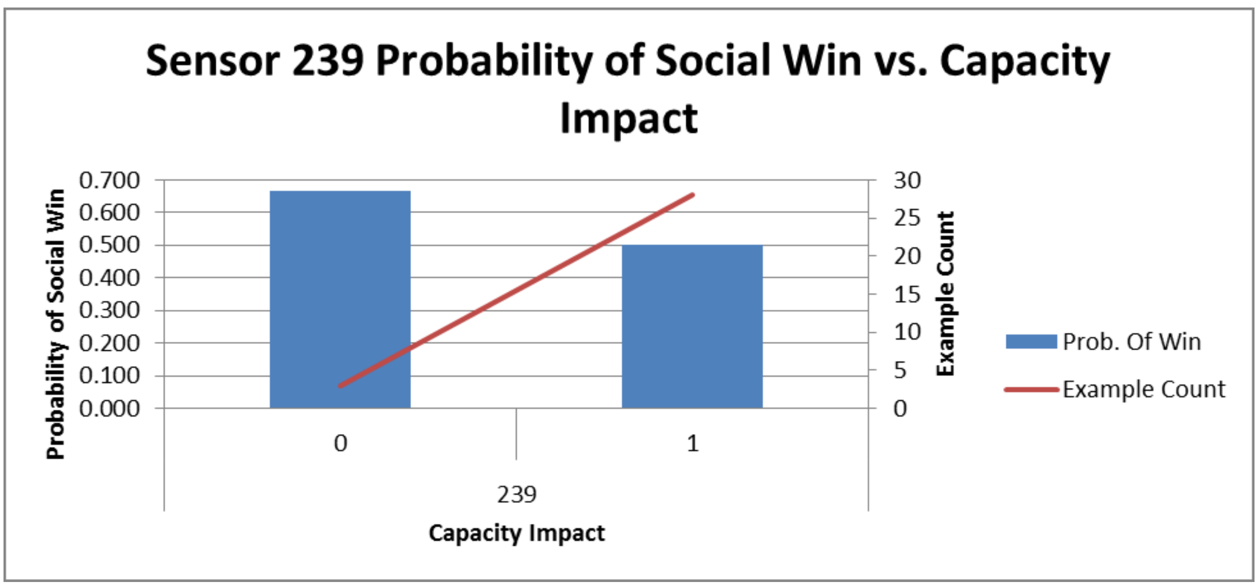
Row Labels	Prob. Of Win	Example Count
1	0.526	19
3	0.553	47
4	0.413	46
5	0.435	69
6	0.425	73
7	0.667	48



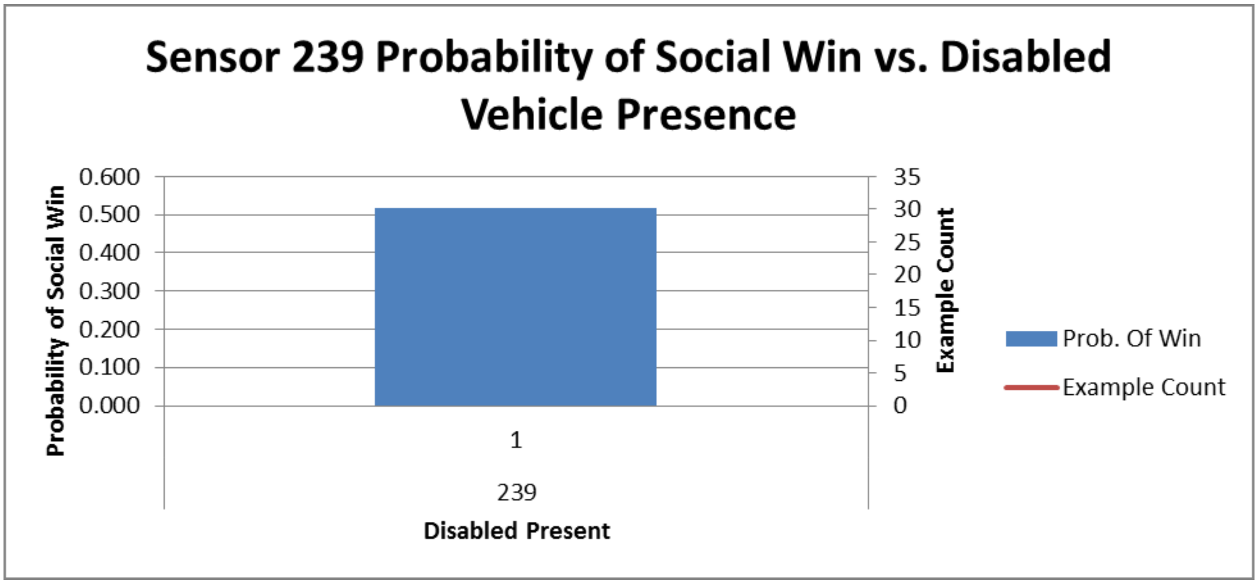
Row Labels	Prob. Of Win	Example Count
0	0.525	40
1	0.519	52
2	0.492	63
3	0.535	43
4	0.536	28
5	0.333	3
6	0.875	8
7	0.485	33
8	0.120	25
9	0.600	5
10	0.500	2



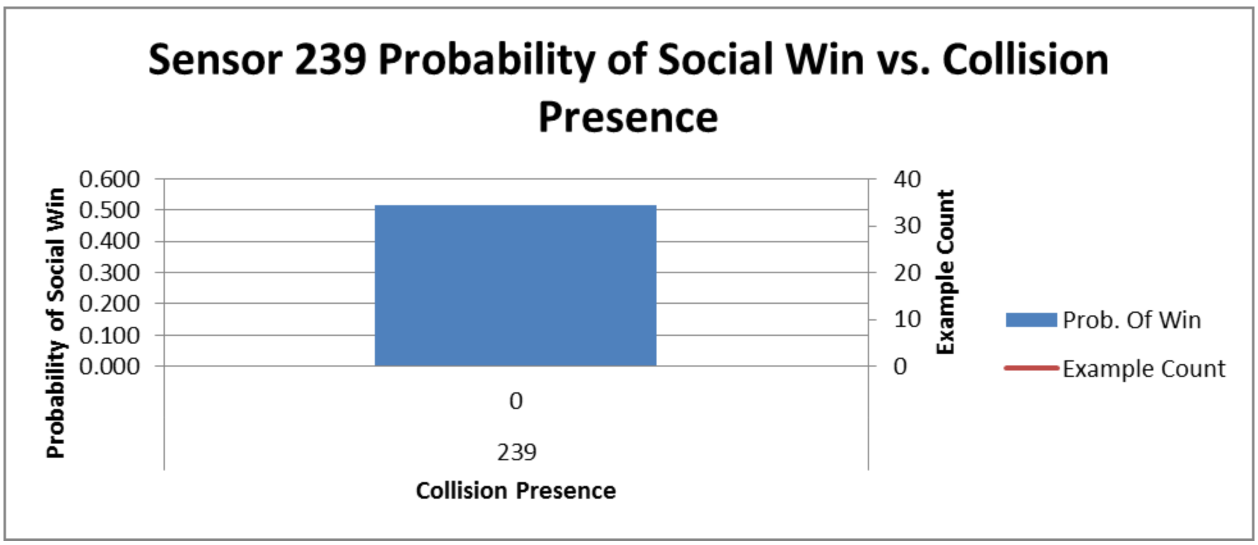
Row Labels	Prob. Of Win	Example Count
0	0.49	271
1	0.52	31



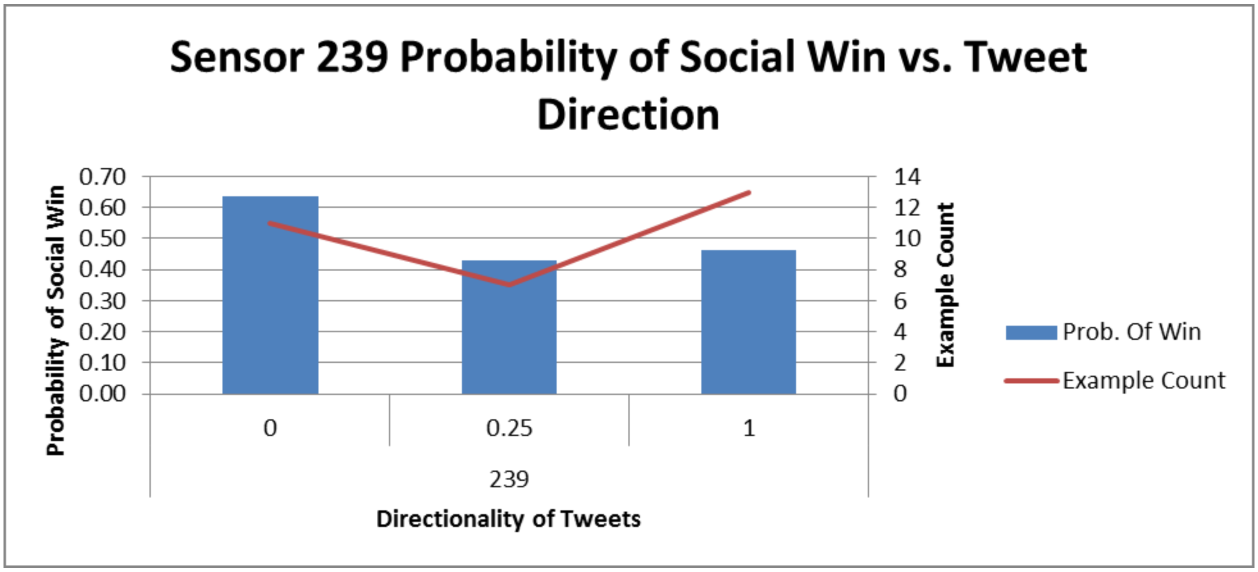
Row Labels	Prob. Of Win	Example Count
0	0.667	3
1	0.500	28



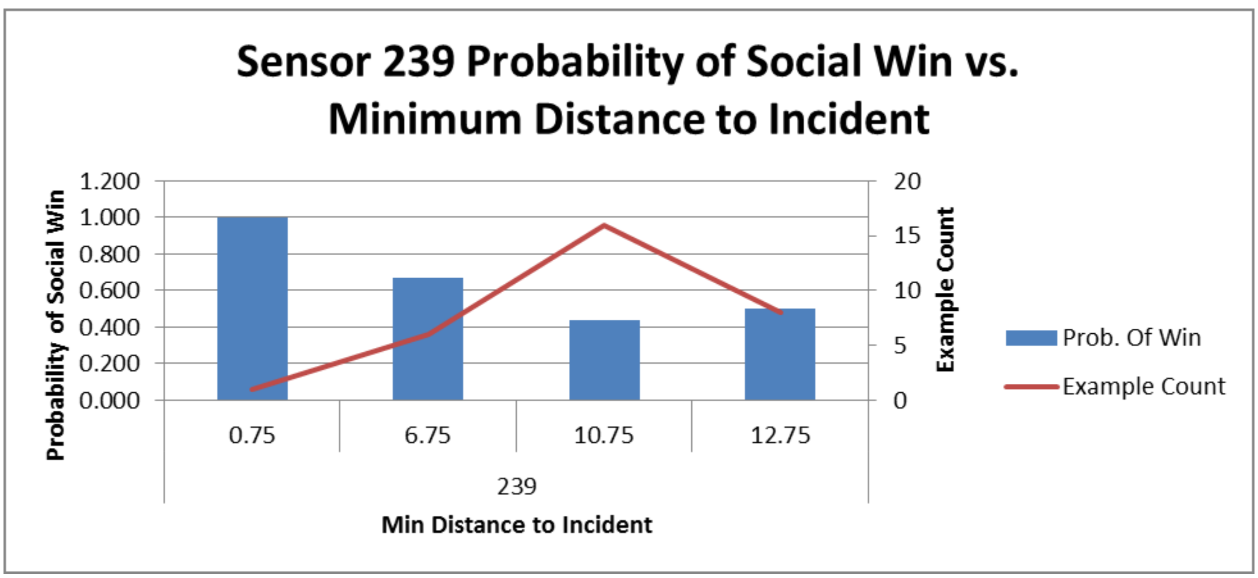
Row Labels	Prob. Of Win	Example Count
1	0.516	31



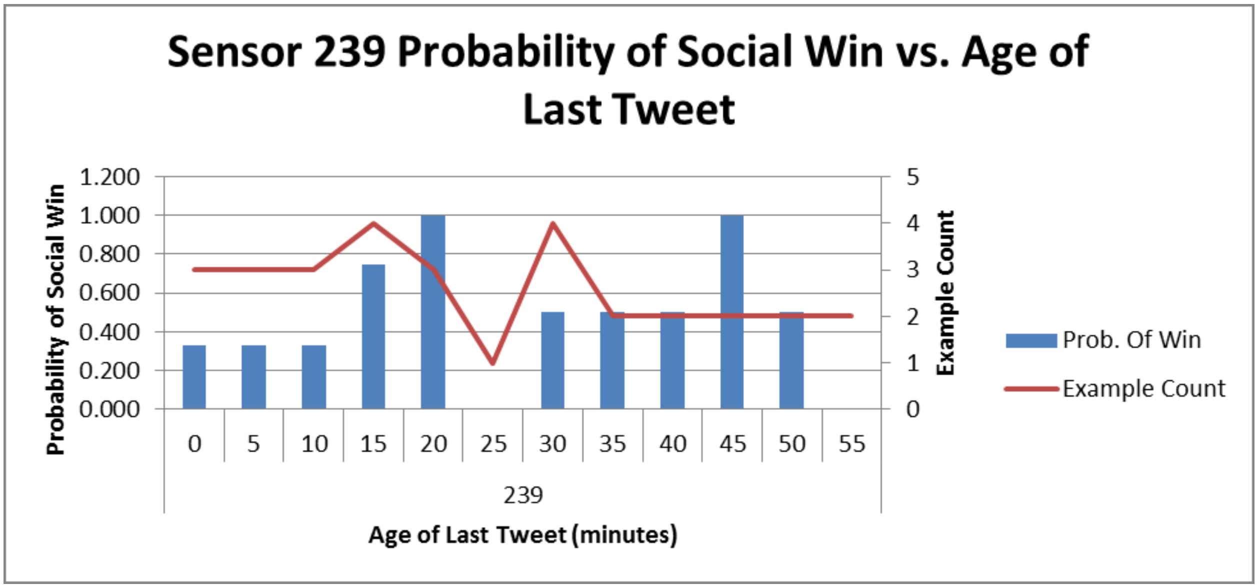
Row Labels	Prob. Of Win	Example Count
0	0.516	31



Row Labels	Prob. Of Win	Example Count
0	0.64	11
0.25	0.43	7
1	0.46	13

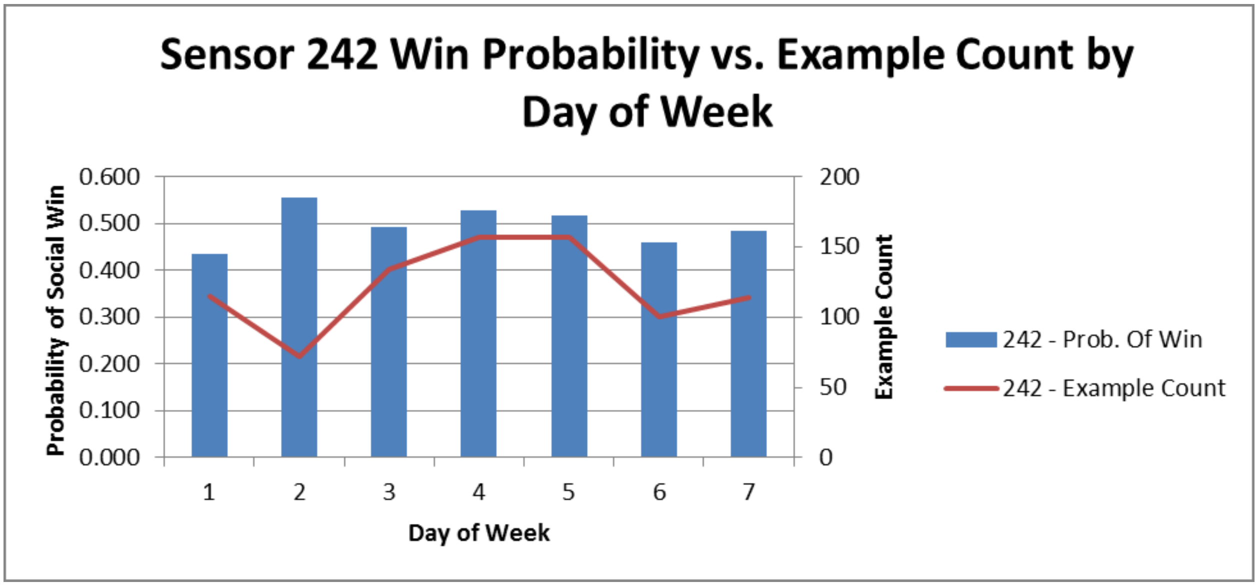


Row Labels	Prob. Of Win	Example Count
0.75	1.000	1
6.75	0.667	6
10.75	0.438	16
12.75	0.500	8

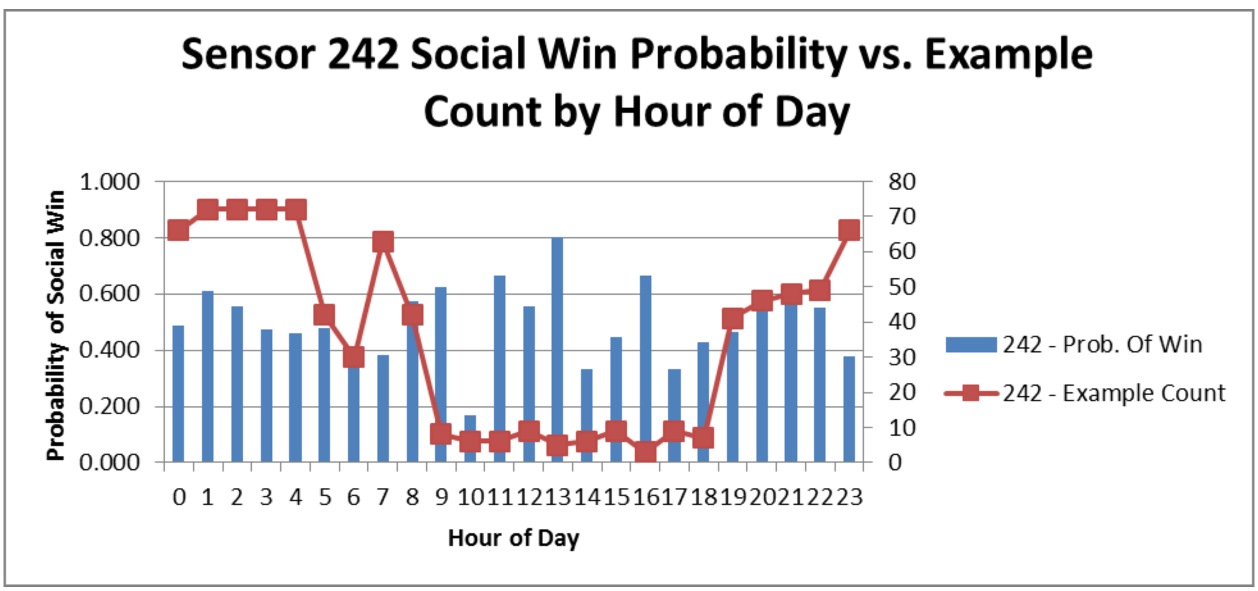


Row Labels	Prob. Of Win	Example Count
0	0.333	3
5	0.333	3
10	0.333	3
15	0.750	4
20	1.000	3
25	0.000	1
30	0.500	4
35	0.500	2
40	0.500	2
45	1.000	2
50	0.500	2
55	0.000	2

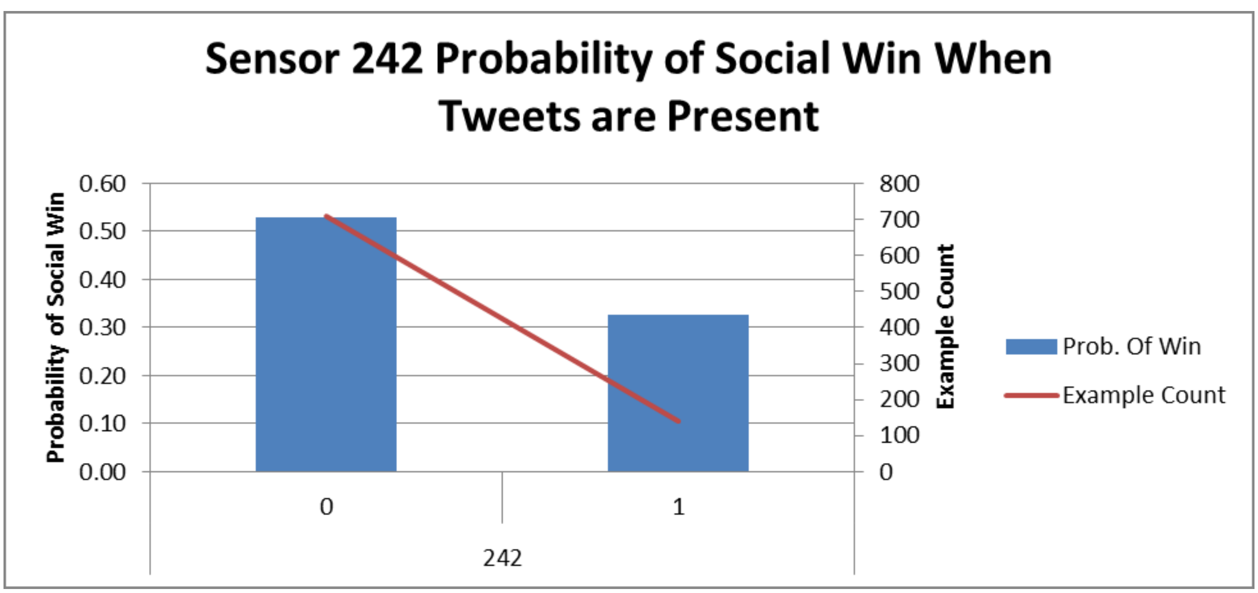
Sensor 242



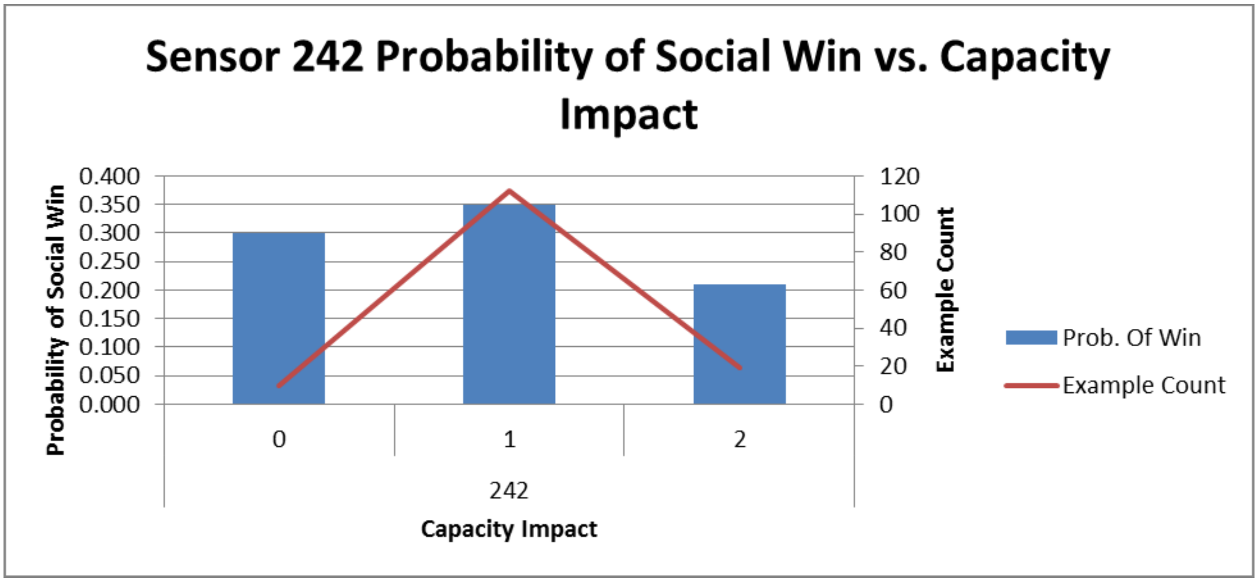
Row Labels	Prob. Of Win	Example Count
1	0.435	115
2	0.556	72
3	0.493	134
4	0.529	157
5	0.516	157
6	0.460	100
7	0.482	114



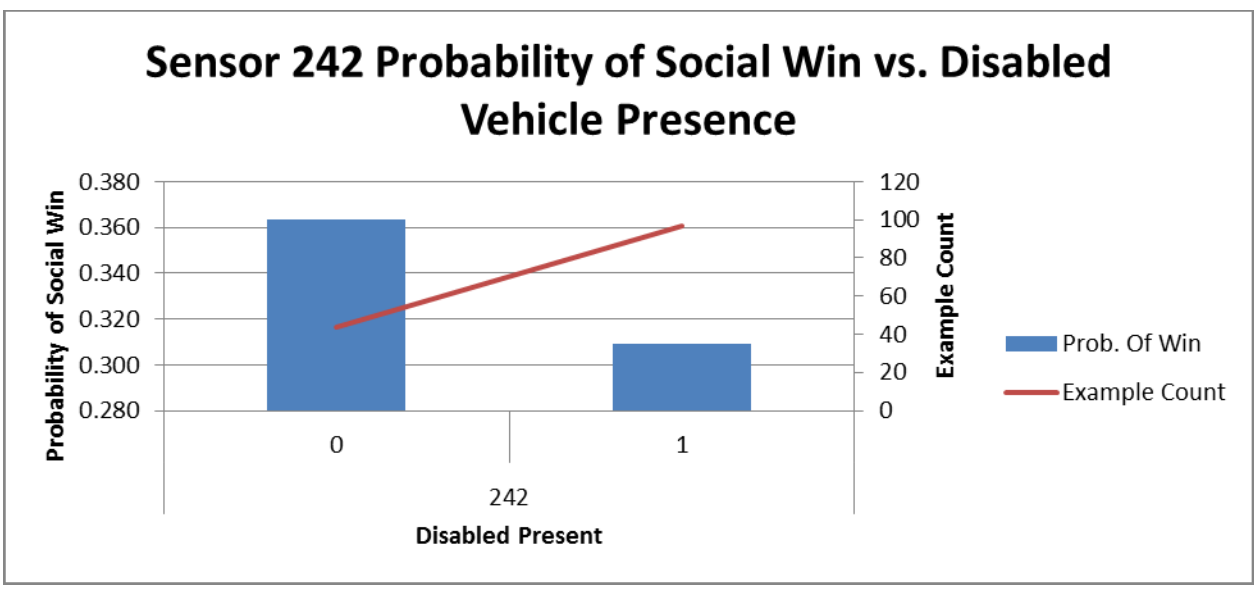
Row Labels	Prob. Of Win	Example Count
0	0.485	66
1	0.611	72
2	0.556	72
3	0.472	72
4	0.458	72
5	0.476	42
6	0.367	30
7	0.381	63
8	0.571	42
9	0.625	8
10	0.167	6
11	0.667	6
12	0.556	9
13	0.800	5
14	0.333	6
15	0.444	9
16	0.667	3
17	0.333	9
18	0.429	7
19	0.463	41
20	0.587	46
21	0.583	48
22	0.551	49
23	0.379	66



Row Labels	Prob. Of Win	Example Count
0	0.53	708
1	0.33	141

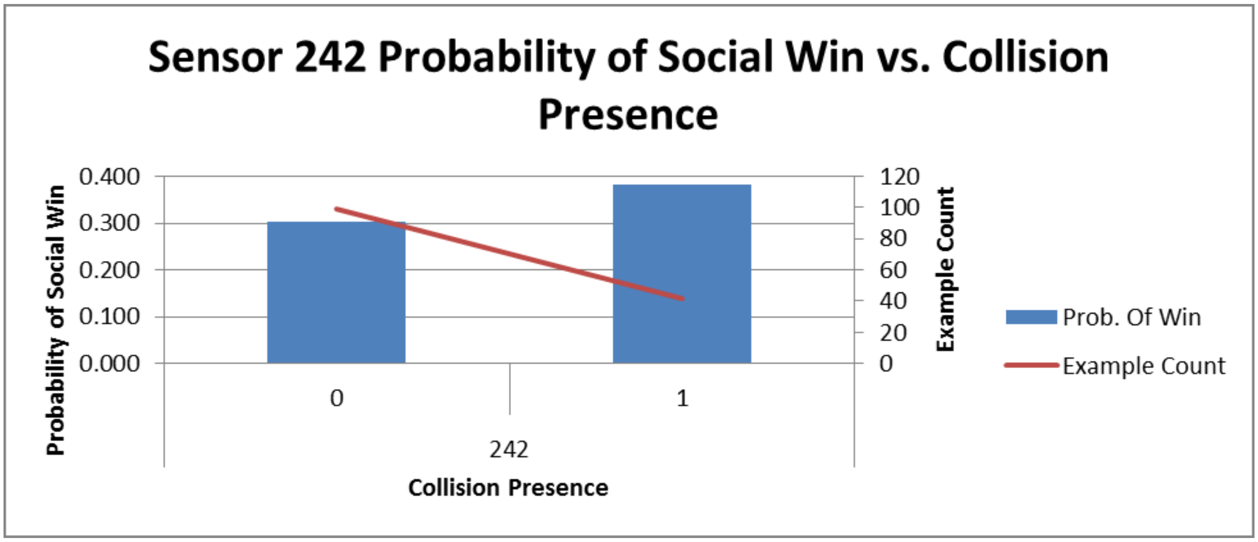


Row Labels	Prob. Of Win	Example Count
0	0.300	10
1	0.348	112
2	0.211	19

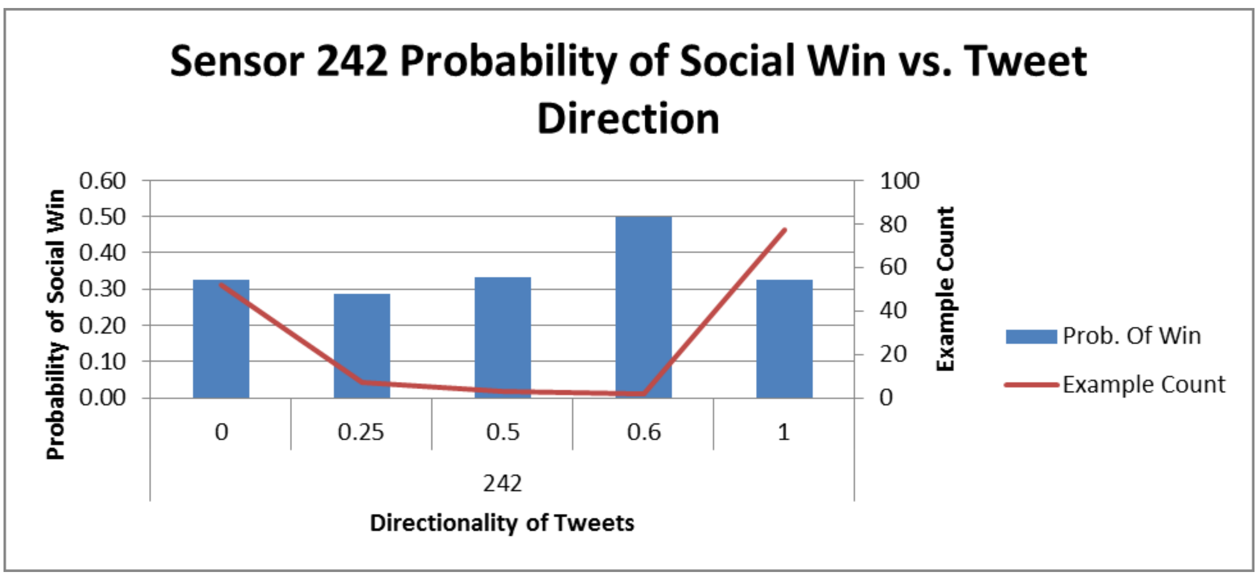


Row Labels	Prob. Of Win	Example Count

0	0.364	44
1	0.309	97

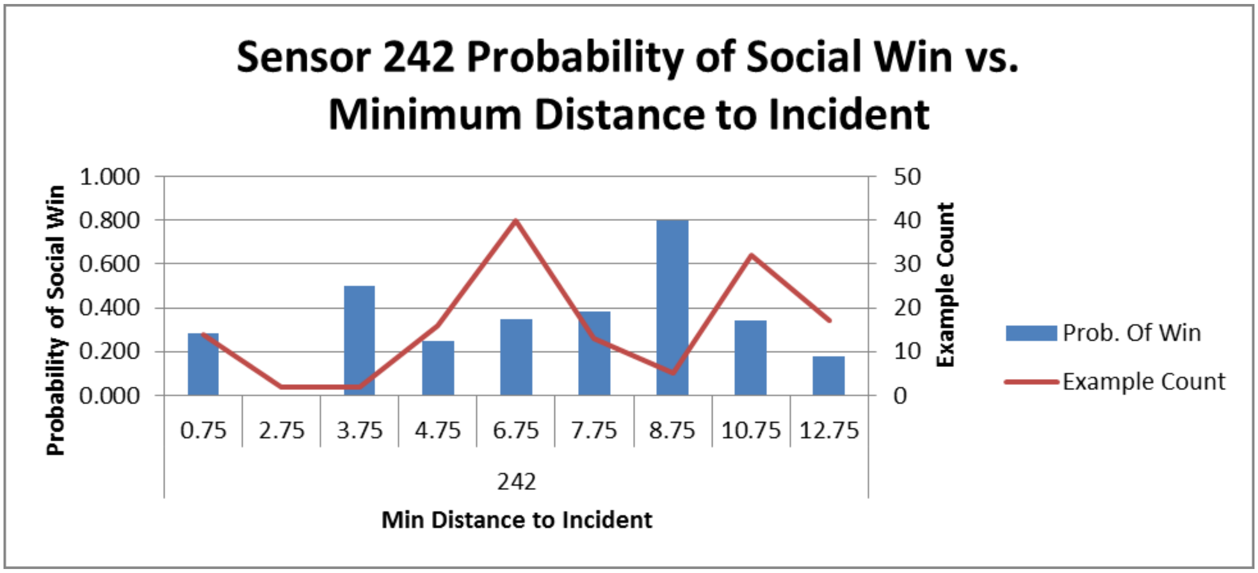


Row Labels	Prob. Of Win	Example Count
0	0.303	99
1	0.381	42

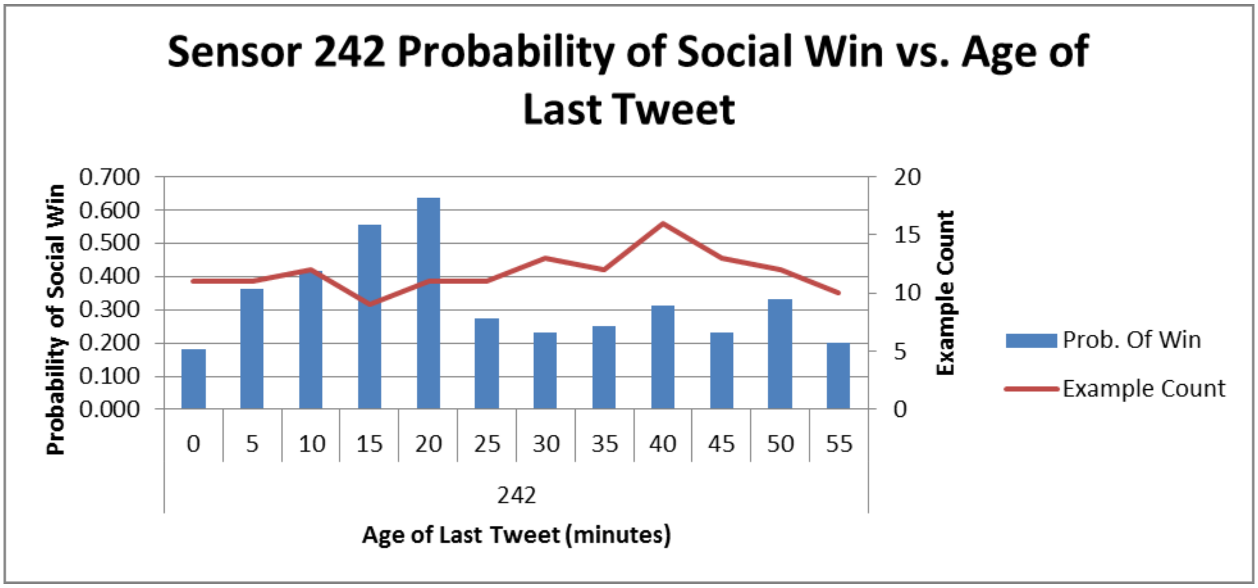


Row Labels	Prob. Of Win	Example Count
0	0.33	52
0.25	0.29	7
0.5	0.33	3

0.6	0.50	2
1	0.32	77



Row Labels	Prob. Of Win	Example Count
0.75	0.286	14
2.75	0.000	2
3.75	0.500	2
4.75	0.250	16
6.75	0.350	40
7.75	0.385	13
8.75	0.800	5
10.75	0.344	32
12.75	0.176	17



Row Labels	Prob. Of Win	Example Count
0	0.182	11
5	0.364	11
10	0.417	12
15	0.556	9
20	0.636	11
25	0.273	11
30	0.231	13
35	0.250	12
40	0.313	16
45	0.231	13
50	0.333	12
55	0.200	10

Bibliography

Twitter Blog. (2011, August 1). Retrieved from Twitter: http://blog.twitter.com/2011/08/your-world-more-connected.html

Backpropagation. (2012, March 10). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Backpropagation

Ensemble Learning. (2012, March 10). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Ensemble_learning

Supervised Learning. (2012, March 10). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Supervised_learning

Unsupervised Learning. (2012, March 10). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Unsupervised_learning

Artificial neural network. (n.d.). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/File:Artificial_neural_network.svg

Boyd, D., Golder, S., & Lotan, G. (2010). Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter . *Proceedings of the 43rd Annual Hawaii International Conference on System Studies* (pp. 1 - 10). Koloa, Kauai, Hawaii: IEEE.

Chen, L., & Chen, C. (2007). Ensemble Learning Approach for Freeway Short-Term Traffic Flow Prediction. *SoSE '07. IEEE International Conference on System of Systems Engineering* (pp. 1-6, 16-18). San Antonio, TX: IEEE.

Cheslow, M., Hatcher, S. G., & Patel, V. M. (1992). *An Initial Evaluation of Alternative Intelligent Vehicle Highway Systems Architectures.* McLean, VA: Mitre Corporation.

Davidov, D., Tsur, O., & Rappoport, A. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. *COLING '10 Proceedings of the 23rd International Conference on*

Computational Linguistics: Posters (pp. 241-249). Beijing, China: Association for Computational Linguistics.

de Moor, A. (2010). Conversations in context: a Twitter case for social media systems design. *I-SEMANTICS '10 Proceedings of the 6th International Conference on Semantic Systems* (p. Article 29). Graz, Austria: ACM.

Dieng, R. (1996). Comparison of Conceptual Graphs for Modeling Knowledge of Multiple Experts: Application to Traffic Incident Analysis. In Various, *Foundations of Intelligent Systems* (pp. 78-87). Berlin / Heidelberg: Springer.

Faghri, A., & Aneja, S. (2007). Artificial Neural Network-Based Approach to Modeling Trip Production. *Transportation Research Record: Journal of the Transportation Research Board*, 131-136.

Haahr, D. M. (2012, 03 24). *FAQ*. Retrieved from Random.Org: http://www.random.org/faq/#Q2.1

Huberman, B. A., Romero, D. M., & Wu, F. (2008, December 5). *Social Networks that Matter: Twitter Under the Microscope.* Retrieved from Social Science Research Network: http://ssrn.com/abstract=1313405 or http://dx.doi.org/10.2139/ssrn.1313405

Hwang, J. S. (2003, June 17). Ontology-based Spatial Clustering Method: Case Study of Traffic Accidents. Pacific Grove, California, USA: University Consortium of Geographic Information Science. Retrieved from University Consortium for Geographic Information Science: http://www.ucgis.org/summer03/studentpapers/juliehwang.pdf

Kotsiantis, S. B. (2007). Supervised Machine Learning : A Review of Classification Techniques. *Informatica* , 249-268.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? *WWW '10 Proceedings of the 19th international conference on World wide web* (pp. 591-600). Raleigh, NC USA: ACM.

Laniado, D., & Mika, P. (2010). Making Sense of Twitter. *The Semantic Web - ISWC 2010: 9th International Semantic Web Conference* (pp. 470-485). Shanghai, China: Springer.

Lawrence, R. (2011). *Social Media Analytics.* Yorktown Heights, NY: IBM Research.

Meliaa, S., Parkhurst, G., & Barton, H. (2011). The Paradox of Intensification. *Transport Policy*, 46-52.

Park, B., Messer, C. J., & Urbanik II, T. (2007). Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network. *Transportation Research Record: Journal of the Transportation Research Board*, 39-47.

Peters, A., van Klot, S., Heier, M., Trentinaglia, I., Hoermann, A., Wichmann, H. E., & Loewel, H. (2004). Exposure to Traffic and the Onset of Myocardial Infarction. *New England Journal of Medicine*, 1721-1730.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. *WWW '10 Proceedings of the 19th international conference on World wide web* (pp. 851-860). Raleign, NC USA: ACM.

Shawe-Taylor, J., De Bie, T., & Cristianini, N. (2006). Data Mining, Data Fusion and Information Management. *Intelligent Transportation Systems*, 221-229.

Shehata, M. S., Cai, J., Badawy, W. M., Burr, T. W., Pervez, M. S., Johannesson, R. J., & Radmanesh, A. (2008). Video-Based Automatic Incident Detection for Smart Roads: The Outdoor Environmental Challenges Regarding False Alarms. *IEEE Transactions on Intelligent Transportation Systems*, 349 - 360.

Traffic Congestion. (n.d.). Retrieved March 10, 2012, from Wikipedia: http://en.wikipedia.org/wiki/Traffic_congestion#United_States

Turner, S. (2004). Defining and Measuring Traffic Data Quality. *Transportation Research Record: Journal of the Transportation Research Board*, 62-69.

van Lint, J. (2004). *Reliable Travel Time Predictions for Freeways.* Delft, Netherlands: TRAIL Research School.

van Lint, J. (2006). Reliable Real-Time Framework for Short-Term Freeway Travel Time Prediction. *Journal of Transportation Engineering*, 921-932.

Wang, J., & Wang, X. (2011). An ontology-based traffic accident risk mapping framework. *SSTD 2011 12th International Symposium on Advances in Spatial and Temporal Databases* (pp. 21-38). Minneapolis, MN USA: Springer. Retrieved from http://sstd2011.cs.umn.edu/files/Slides/SSTD_Jing.ppt

Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., & Chen, C. (2011). Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 1624-1639.

Zheng, W., Lee, D.-H., & Shi, Q. (2006). Short-Term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach. *Journal of Transportation Engineering*, 114-121.