Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Master's Thesis

Estimating Confidence in Spatio-Temporal Models

Ethan Oswald Massey

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Erwin Prassler
Prof. Dr. Paul G. Plöger
M. Sc. Argentina Ortega Sáinz

April 2020

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

_____ _____

Date Ethan Oswald Massey

Abstract

Spatio-temporal models are often used in robotics to provide predictions about how and when dynamic changes occur in a given environment. They can be used in a variety of ways, but of specific interest are their uses in navigation, planning, and scheduling. While a diverse set of spatio-temporal modeling techniques exist, the majority of these models are unable to provide an indication of the confidence behind a given prediction once it has been made. These confidence values can be used to improve navigation by avoiding areas of high uncertainty during important tasks, provide travel time estimates for planners, and inform self-directed robotic exploration of an environment to improve environmental models.

This work presents a methodology for the creation of confidence estimates for pre-existing spatio-temporal models. The new methodology is then applied to Hypertime, a state-of-the-art spatio-temporal modeling technique, to create a total of eight confidence estimation models. These models have been designed for use with ROPOD, an autonomous multi-robot platform for the indoor transportation of goods. Multiple real-world datasets have been collected in association with ROPOD and are presented. Criteria for the evaluation of these new confidence estimating spatio-temporal models are outlined. These new models are then tested using the real-world datasets and their performance is analyzed with the new criteria. In the final experiment, a proof-of-concept is demonstrated, which combines the predictions of multiple world models in order to demonstrate a possible additional use for confidence estimates. Finally, recommendations for ROPOD are made using the results for the experimental section and an outline of possible future work is presented.

Acknowledgements

I would like to firstly thank my advising professors Prof. Dr. Erwin Prassler and Prof. Dr. Paul G. Plöger. Not only for their contributions to my education but for their support on this thesis. I would also to express my sincere gratitude to my advisor M. Sc. Argentina Ortega Sáinz. Her support, guidance, and encouragement made this project possible.

To my friends and family, I extended a warm and heartfelt thank you. Your emotional support was invaluable. Specifically, I'd like to mention my fianceé, Megan, for her love and continued understanding while pursuing this work. Lastly, a special thank you to Megan, Sam, and Dylan for your time spent proofreading and correcting the earlier revisions of this work.

Finally, I'd like to extend my thanks to all my educators, past and present, who have contributed to my academic and professional career. Thank you.

Contents

List of Figures

List of Tables

1

Introduction

As the field of automation has continued to grow, there has been a corresponding increase in the demand for autonomous robots to be integrated into environments that are shared with humans. These human-shared environments can often be quite hectic and challenging for robotic systems that have historically been rigidly designed to work in a very procedural manner. Due to an environments ever changing nature, this conflict is particularly acute in situations involving autonomous navigation. To successfully plan and navigate, whether moving goods around a factory or autonomously operating a vehicle, a robot must be able to effectively and efficiently deal with the dynamic changes that are an unavoidable part of operating in real-world environments. It is therefore desirable to design and implement tools to enable a robot to cope with the dynamic changes that are inherent to sharing an environment with humans.

1.1 Motivation

Spatio-temporal modeling is a way of describing dynamic changes that happen in an environment with respect to both location and time. These models are often used in environmental science to make predictions about that future state of a given climate or region. Spatio-temporal models have also been applied to the field of robotics to describe daily activities or changes in a human environment. The use of these models increases a robots ability to deal with the dynamic elements of

an environment by enabling them to make informed decisions and adapt to predicted changes. Naturally, the more accurate a spatio-temporal model, the better a robot can predict these changes. To that effect, the ability to accurately predict changes in turn dictates how effectively and efficiently a robot is able to operate and avoid encountering unexpected situations it may otherwise be unable to deal with.

Spatio-temporal models have been particularly useful in improving the ability of a robotic system to navigate dynamic real-world environments. Currently, there are a variety of potential methods for accomplishing such modeling, each with there own strengths and weaknesses. Regardless of implementation details, a models ability to make accurate predictions is predicated upon the quantity and quality of data collected and analyzed. With an insufficient amount of data, or with poor quality data, spatio-temporal models can often make inaccurate predictions that may ultimately cause more problems than they solve. It would therefore be highly advantageous if a model was able to also quantify the confidence associated with a given prediction. These confidence estimates would provide planning, scheduling, and navigation systems insights about a given environment that would enable them to make more informed and optimal decisions, resulting in more efficient and dependable robotic systems.

1.2 Problem Formulation

In order to improve the current state of spatio-temporal modeling development it is vital to establish a methodology for designing techniques that enable spatio-temporal models to provide confidence estimates alongside their predictions. Once this methodology has been established techniques for estimating confidence in predictions can be implemented. These confidence estimates can then be used in a variety of ways. In the most direct application, these confidence estimates could be used to inform planners or schedulers. If the confidence estimates provided by a model changed with respect to the time of day then a planner or scheduler could prioritize specific times in order to maximize the likelihood of success. Conversely, times during the day with low confidence could be target for exploration. Daily patrols could be done to improve times of the day where operation was common and uncertainty was

high. Moreover, the benefits of quantifying prediction confidence are evident at the operational or meta level. For example, if a robot could be made to maintain several spatio-temporal world models simultaneously, and if each of those models had an associated confidence factor, it would be possible for a robot to dynamically switch between these different methods. This would yield improved predictions and thus increase the system's operational efficiency.

With all of these benefits and possibilities in mind, the objective of this thesis is to design, implement, and test methods for quantifying confidence in spatio-temporal models. A methodology will be presented demonstrating how to approach this task. This methodology will then be applied to develop confidence estimating techniques for an existing spatio-temporal model. Metrics by which to judge these confidence estimating techniques will also be presented. Comparisons between different techniques will be made to determine their efficiency and effectiveness with respect to varying sets of real-world data. A proof-of-concept model that combines predictions from multiple models will also be demonstrated. This will provide insight into possible applications and will motivate future work.

2

Related Work

2.1 Spatio-Temporal Modeling Methods

2.1.1 FreMEn

Frequency Map Enhancement (FreMEn)[11] is a spatio-temporal modeling technique that was introduced in 2014 and specifically designed for use in long-term applications. These long-term applications, usually on the order of weeks or months, provide ideal conditions for a robot to collect ample information about the environment it is operating in. The driving principle behind the FreMEn algorithm is extracting periodic behavior from the information collected and leveraging it to make future predictions. This assumes, often correctly, that there are underlying periodic behaviors in the environment the robot is operating in. While not always the case, this is often a safe assumption when operating around or interacting with humans that have daily and weekly routines. The algorithm is only able to quantify the likelihood of a given event happening. This means that while the algorithm may be able to say that a door is likely open at a given location and time, it would be unable to predict how long it would take to get to the door from the robot's current location.

Implementation Details

Internally, FreMEn is based around converting observations from the time domain they were originally collected in, to the frequency domain for analysis, and then recreating a representation in the time domain for predictions. These transitions between the time and frequency domain connect conceptually to variations of the Fourier Transform. While the principles are applicable, the actual method of implementation varies from classical Fourier Transform implementations. These variations increase the flexibility of FreMEn. Although a Discrete Fast Fourier Transform (DFFT) was considered, it was ultimately decided against for two main reasons. First, any updates to the model would require the recovery of the entirety of the previously observed states. The authors note that this does not scale well and becomes increasingly computationally demanding. Second, a DFFT would require all samples to be equally spaced, which is not always possible when dealing with data obtained from robots operating in a real-world environment.[18] It is for these reasons that the authors developed a new method for storing and querying temporal information. This method displays compression ratios of 1:1000 while losing less than 5% of information.[18] The method implementation is as follows.

The desired function takes the form $P(t)$ which produces the probability of an object being in a given state, such as a door being opened or closed, at a given time t. $P(t)$ is composed of the sum of l triples, where l is known as the order of the function. These l triples each consist of relevant frequency data: frequency ω, time shift φ, and amplitude α. To calculate $P(t)$ at a given t, the l triples are summed up in Equation 2.1.

$$p(t) = \zeta(\alpha_0 + \sum_{j=1}^{l} \alpha_j cos(\omega_j t + \varphi_j))$$

(2.1)

Where additionally, α_0 is the static, or mean, probability of a given event and ζ is a clamping function that serves to somewhat limit the predictive range of FreMEn to be between 5% and 95%. The l best triples are chosen from k candidate ω frequencies know collectively as Ω. The number and frequency of these candidate frequencies are a tuneable parameter but are often set with Ω being 24 and ω being driven by Ω. The ω are often equally spaced such that the smallest ω would represent behaviors that happen roughly every hour and the largest ω representing behaviors that happen every day. The ω amplitude is determined by doing a series of summations after every observation of a given object at a time t. The state of that object at that time is given by $s(t)$ where $s(t)$ lies between 0 and 1. The updated equations can be seen in Equation 2.2.

$$\begin{aligned} \gamma_0 &\rightarrow \frac{1}{n+1}(n\gamma_0 + s(t)) \\ \gamma_k &\rightarrow \frac{1}{n+1}(n\gamma_k + (s(t) - \gamma_0)e^{-jt\omega_k}) \forall \omega_k \in \Omega \\ n &\rightarrow n+1 \end{aligned} \tag{2.2}$$

The absolute value of $|\gamma_k|$ represents the amplitude α_k. The l largest γ_k are then selected. This determines the three values of each l triple where $\alpha_j = |\gamma_k|$, $\omega_j = \omega_k$, and $\varphi_j = arg(\gamma_k)$. Further details on the implementation, limits and applications can be found in [11, 18, 9, 10].

Relevance

FreMEn is of particular interest for a number of reasons. Importantly, it has been shown to perform well in both computational resource usage and predictive power [6]. Additionally, although it is only able to make predictions about the likelihood of an event occurring, it is the only method known at this time to also inherently include a method for estimating confidence. This confidence comes in the form of its probabilistic prediction. Applications of this value are shown in [15] and discussed later in this section. Finally, the concepts and ideas of FreMEn can be applied to build more complex spatio-temporal models, as will be seen below.

2.1.2 Hypertime

Hypertime[19], originally introduced in 2018, is a spatio-temporal modeling technique built on top work done with FreMEn. Thus, it is also designed for use in long-term applications of weeks or months and relies solely on the data collected by the robot during run-time. Additionally, like FreMEn, it makes the assumption that environmental and behavior changes exhibit some sort of periodic behavior that has an underlying frequency and is thus predictable. Where it differs from FreMEn is its predictive capability. Hypertime is able to provide predictions in the form of real numbers. That is, Hypertime can make predictions such as how many people will be in a given area[20] or roughly how long it will take to travel the edge of a graph. However, this predictive capability does come at a cost and Hypertime, unlike FreMEn, is not able to make any confidence estimations about the predictions that it makes in its current implementation.

Implementation Details

When looking at the internal implementation of Hypertime, it is clear why it shares so many similarities with FreMEn. At its core, Hypertime can be conceptualized as being built from a collection of FreMEn models. The ability to use a collection of FreMEn models to make non-binary predictions lies in the novel use of data clustering to characterize a location's temporal behavior. Take, for example, a system tasked with characterizing the amount of people in a section of hallway as either being

completely free, partly busy, or completely obstructed. It would then be possible to create a FreMEn model for each one of these states and then take a sum of their predictions to yield a single model. In fact, this is very close to what Hypertime does. The end result is a function that provides the predicted state of a given location at an arbitrary time. While this has proven powerful, the current implementation is unable to provide probability estimates or bounds on non-binary values.

While an in-depth analysis of the individual, implementation-specific details of Hypertime are outside the scope of this work, an overview of the method's internal workings and necessary background information are covered in this section. At its core, Hypertime can be viewed as a collection of multi-dimensional Gaussian clusters. These clusters are created using an iterative five-step process:

1. Initialization
2. Spatio-temporal clustering
3. Model error estimation
4. Identification of periodicities
5. Hypertime space extension

Initialization is a one-time operation that consists of preparing internal data structures and loading the training data. An individual piece of training data consists of two parts, the time t, and the spatial location it was recorded at x. Note that while in the simplest case x must be at least 1-dimensional, in more advanced applications it may be an array representing a 2- or 3-dimensional location.

After initialization, the spatio-temporal training data is clustered. The coordinates used for the clustering and visualization of the spatio-temporal data are determined using their spatial location and a projection of their temporal data from a 1-dimensional linear space into a 2-dimensional circle. This warped time is determined by the Equation 2.3 where t is the original time of observation, i is detonates a specific observation, T is a given temporal period, and t_{coord} are the

resulting temporal coordinates.

$$t_{coord} = cos(2\pi t_i/T), cos(2\pi t_i/T) \tag{2.3}$$

Once coordinates for every training point have been established, clustering can begin. Clustering methods vary, but K-Means and Expectation Maximization are both commonly used. An example of this clustering is shown in Figure 2.1. In this figure a simple 3-dimensional case is shown where the location on the temporal circle is a result of its warped time coordinates given by the desired time of a prediction t and the vertical axis represents the observed value, i.e. number of people. In the case of the Expectation Maximization variant of Hypertime, the technique used throughout this work, the number of clusters is a predetermined value given at training time. Additionally, Gaussian mixture models are used to represent these clusters.

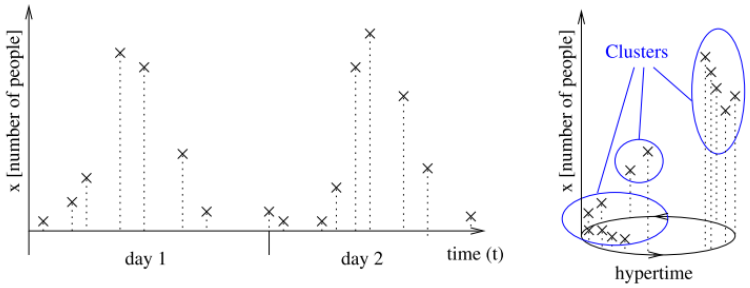

Figure 2.1: An example of spatio-temporal data clustering using Warped Hypertime[20]

With the clusters created, model error estimation can begin. This can be achieved in many ways but often sampling can be used to provide predictions at any given time t. This is done by sampling along a vertical axis at a time t. Each of the samples will have a likelihood of belonging to a given cluster. Predictions can then be made by either selecting the most likely value from the list of samples, or by combining the samples using their likelihoods as the weight in a weighted sum.

Regardless of method, Hypertime is then asked to make predictions for each entry in a training dataset. The results of the model's predictions are then compared against the training dataset and used to calculate the model's error over time.

FreMEn is then used to analyze the model's error over time to find the dominate frequency of error. The new dominant error frequency is used as the next temporal period T_{i+1} to extend Hypertime space using Equation 2.3. Essentially, two new coordinates are added to each and every training data point. One way to visualize this is to imagine multiple cylinders like those seen in Figure 2.1 superimposed on top of one another. This is beneficial not only as a way to represent complex temporal behavior, but also as a way to analyze and visualize temporally sparse training data.

It is at this point that the iterative process beings anew. With the new frequency added, new clusters are created, predictions are made, and the error is analyzed. This process continues until the model error produced by adding another frequency is larger than the previous model's error. Further implementation details along with variants of Hypertime can be found in [19, 20].

Relevance

Hypertime improves and expands upon FreMEn to create a powerful spatio-temporal model. This was shown in [6] when it was compared with other state-of-the-art spatio-temporal models. Despite its performance with both simulated and real-world data however, the authors note in [19] that future work remains to determine when and where to collect data to improve the spatio-temporal model. One way this could be accomplished is by using the confidence value of a given area at a given time to inform when and where to collect data. The specific details of this are discussed in the "Applications of Confidence Values and Data Collection" section.

2.2 Confidence and Types of Uncertainty

The topics of confidence and uncertainty are wide-reaching and often have similar but important differences in academic fields of study. The field of uncertainty estimation, in particular, is vast and its intricacies in their entirety are outside the scope

of any one single work. Success while working on problems related to confidence estimates has been achieved by setting a specific definition of what uncertainty and confidence mean with respect to the work presented. Previous work in the field of machine learning provides insight into what assumptions can be safely made about confidence estimates and uncertainty such as the work of [22]. In this particular work they establish that "uncertainty can be viewed as negative confidence and vice versa". This inclusive definition allows them more freedom in approaching their task of uncertainty estimation while also simplifying communication with the reader. The work of [22] also provides examples of using classification margins, or the distance from a decision boundary, to estimate confidence. This concept could be applied to spatio-temporal models where the classification [error] margins provided by a model could be used to quantify confidence in a given prediction.

Further exploration on the topic of uncertainty reveals two sub-categories of uncertainty that can be useful when analyzing and explaining model performance. The two types of uncertainty are "aleatoric" and "epistemic" The work in [7] provides working definitions for these two categories. Aleatoric uncertainty is defined as inherent to a given behavior and thus cannot be reduced beyond a certain minimum amount. Conversely, Epistemic uncertainty originates from incomplete knowledge about a behavior. In theory it is possible to reduce this error through additional observation and data collection, although it is noted that this may not always be possible or feasible due to restrictions on resources such as time or other physical restrictions inherent to the means of data collection. With respect to spatio-temporal models, often large amounts of temporally and spatially dense data are needed for accurate predictions. As a result of these relatively high requirements, datasets are often unable to provide complete description of the underlying behavior in a given environment. This leads to a high amount of epistemic uncertainty.

Further machine learning specific information on aleatoric and epistemic uncertainty is provided in [2]. In this work, the authors attempted to define and model uncertainties that had the largest impact on a computer vision classification problem. Through this work they demonstrate a link between the two types of uncertainty. By designing models that attempted to only capture one of the two types of uncertainty

it was discovered that the models will attempt to compensate for the other type of uncertainty that is not being modeled. That is to say that when graphed, the resulting models shared a high degree of similarity. This indicates the difficulty involved in attempting to specify between the two types of error in a given model.

While it may be difficult to differentiate between the two types of uncertainty, it is still possible to reduce some of the epistemic uncertainty. By collecting and introducing more data [2] demonstrated that it is possible to reduce this type of uncertainty. Importantly, they noted that while more data can reduce epistemic error, it is vital that the new data provide novel information not already provided in the test set. This reinforces the notion of epistemic error being connected to the sparsity of data collected and the importance of crafting training and test datasets with respect to the behaviors present vs those that are desired to be modeled.

2.2.1 Confidence Intervals

One of the most common tools applied by statisticians when analyzing data is the confidence interval. In general, the confidence interval can be thought of as a way to quantify the likelihood that a given value or parameter will lie within a certain range. One of the most common forms the confidence interval takes is the standard interval as seen in Equation 2.4 where $\hat{\theta}$ is an estimate of θ with an interval driven by α, the percentile of normal deviation, and where the estimated standard deviation is $\hat{\sigma}$ [21]. Since these values are not inherently known, bootstrapping must be used for estimation. Where "bootstrap is a computer-based method for assigning measures of accuracy to sample estimates" [4].

$$\hat{\theta} \pm z^{\alpha}\hat{\sigma} \quad (2.4)$$

Indeed, in the field of machine learning, confidence intervals or similar predic-

tions intervals are often used to put a bound on classification accuracy or error. Additionally, they have been used to estimate uncertainty [5] which, for certain domains of works, can be seen as being inversely correlated with confidence [22]. While very good at estimating and placing bounds on an estimate of an unknown population parameter, it is important to note that confidence estimates do not provide information on the underlying distribution of individual values.

2.2.2 Applications of Confidence Values

The STRANDS Project: Long-Term Autonomy in Everyday Environments[15]

STRANDS stands for Spatiotemporal Representations and Activities for Cognitive Control in Long-Term Scenarios. The project's stated goal is to increase robotic performance in Long-Term Autonomy (LTA) specifically related to public indoor environments such as offices and hospitals. At the date of the paper's publication, the authors state that their robots had operated a cumulative 104 days and traveled over 116km while operating autonomously for weeks to months at a time. The STRANDS project consists of state-of-the-art work from a variety of fields including scheduling, task planning, and local and topological navigation.

FreMEn, the spatio-temporal modeling method, was one of these state-of-the-art works included in the project. It was used primarily for adaptive navigation where it predicted the traverseability of a given edge of a topological graph at a given time, and for predicting human-robot interaction. The human interaction predictions are of particular interest, however, as a type of active learning was applied using the confidence values produced by FreMEn to improve human-robot interaction over time. As the ideal locations for human interaction were unknown at the beginning of the robot's deployment, it was necessary for the robot to explore in order to find new and potentially better areas for more human interaction. This resulted in a trade-off between exploration, finding new areas of interaction, and exploitation (that is, actually using the spatio-temporal knowledge provided by FreMEn to maximize the number of interactions). Schedules for spatio-temporal exploration were generated using predictions provided by FreMEn in tandem with an application of Monte Carlo sampling.

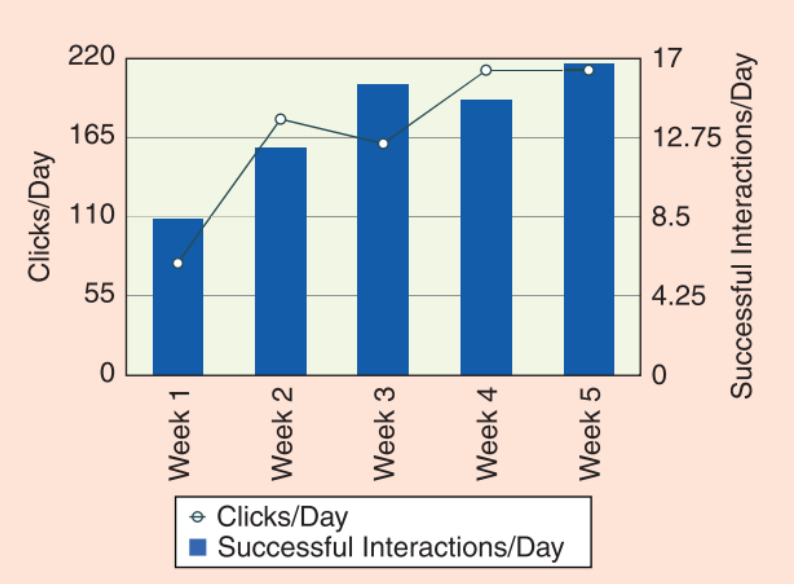

Figure 2.2: Increasing number of successful human interactions using confidence estimation [15]

Specifically, [8] details the methodology behind maximizing information gain by utilizing the uncertainty FreMEn provides. The information collected at a given time is defined as $I(t)$. Each cell of a grid or each edge of a graph is said to have its own $I(t)$ and its own FreMEn model. $I(t)$ is then calculated for each cell or edge by taking the difference of its a-priori entropy, $E(t)$, and its a-posteriori entropy, E_r. Thus, $I(t) = E(t) - E_r$ which represents the difference between the probability of occupation/traverseability before and after observation. The authors also provided an expanded version of this equation using $p(t)$ where $p(t)$ is the probability of occupation/traverseability, denoted by the subscript c, at a given time. This can be seen below in Equation 2.5.

$$I(t) = -p(t)log_2P(t) - (1 - p(t))log_2(1 - p(t)) + p_clog_2p_c + (1 - p_c)log_2(1 - p_c)$$

(2.5)

Given the positive results seen in STRANDS, it stands to reason that something similar could be applied to other active learning strategies and with other spatio-temporal models if a method for determining confidence is developed. Specifically, if Hypertime were able to provide confidence estimates, it would be possible to make informed decisions about the best location and time for data collection to improve the spatio-temporal mode. Furthermore, it may be of interest to compare how a model-independent confidence value would compare to that of a confidence value provided by a model itself or one produced specifically in connection with that method.

2.2.3 Temporal Planning

Another area of potential application for spatio-temporal models with confidence estimates is temporal planning. Temporal planning is an area of active research that revolves around finding optimal ways to reach an end state given an initial state, all possible relevant actions, and temporal bounds on those possible actions. These actions can then be thought of as intervals, as opposed to classical planning where actions are often thought of happening instantly. This type of planning is often used with systems that depend on, and can execute, more than one action or task simultaneously as is often common in multi-robot systems. [3]

In some approaches to temporal planning, the mean and standard deviation of an event duration are used to devise appropriate solutions. This can be seen in methods such as Static Robust Execution Algorithm (SREA)[14] and Dynamic

Robust Execution Algorithm with Mitigation (DREAM)[13]. If spatio-temporal models were able to provide confidence estimates in the form of a mean and variance, or similar, they could be used in connection with these temporal planners. Higher-accuracy and higher-resolution spatio-temporal model confidence estimates would allow for more informed and precise predictions to be made by these temporal planners.

3

Methodology

The term "confidence" is used in many fields in a variety of subtly different ways. Thus, before beginning to develop or assess methods of producing confidence estimates for spatio-temporal models it is critical to define what exactly is meant by "confidence". For the purposes of this work, confidence will be defined in accordance with the colloquial understanding of the word. That is to say, confidence is commonly defined as "The feeling or belief that one can have faith in or rely on someone or something."[17]. Using this definition as a reference, in this work confidence will be treated as a numerical representation that attempts to quantify the amount of faith a model puts into a given prediction, or set of predictions, that is produced by that model. This definition was specifically crafted in order to encompass a litany of methods for estimating confidence values. For example, while classic and rigorous statistical methods such as confidence intervals may be used to estimate confidence, other, less traditional methods may also be used to produce similar or better results.

3.1 Confidence & Uncertainty

When attempting to quantify confidence, the concept of uncertainty often comes up. Although occasionally differentiated in some academic work, these terms can typically be viewed as two sides of the same coin. Using the above definition of confidence as the modus operandi, for the purpose of this work, uncertainty will be treated as the inverse or negative of confidence. This is similar to the work of [22]. Connecting these two concepts expands the pool of related work available to

be drawn upon, while also simplifying the communication and understanding of the core concepts presented herein.

3.2 Inherent Uncertainty

When investigating and developing models to describe confidence or uncertainty it is important to evaluate where the sources of uncertainty may originate. Uncertainty, and thus lack of confidence, can be classified as one of two types of uncertainty, aleatoric or epistemic. Epistemic uncertainty can often take the form of an incomplete or noisy dataset that obscures the underlying behavior. In theory, this uncertainty can be minimized by obtaining a less noisy and more complete dataset. Ideally, a good model should be able to indicate when and where data should be collected to minimize this type of uncertainty. Aleatoric uncertainty is much harder to manage. This uncertainty is often the result of unknown or unknowable variables that result in variations in data and therefore in predictions. Aleatoric uncertainty is also known as statistical uncertainty. Unlike epistemic uncertainty, aleatoric uncertainty cannot be removed by data collection, at least when continuing to use the same methods for gathering data.

Assumptions made about the underlying behavior being modeled also contribute to these sources of uncertainty. In the specific case of spatio-temporal models, it is assumed that there exists one or more underlying behaviors that are both spatially and temporally dependent. Furthermore, it is assumed that, through a significant, but yet undetermined amount of observation, these behaviors can be described accurately enough to make useful predictions. That is to say, even the best spatio-temporal model would be inaccurate if a provided dataset was completely random and does not contain any underlying behaviors. In such an instance, any problems in the dataset would ideally be detected via the confidence estimates provided by the model. Furthermore, assumptions about how to model underlying behaviors also introduces uncertainty. If, for example, travel time between two nodes is assumed to be, and is modeled as, a Gaussian distribution of times, uncertainty has already been introduced. Regardless of the root cause of the uncertainty, a given spatio-temporal model and a corresponding dataset contain numerous avenues from which uncertainty is introduced. Modeling and providing this uncertainty provides

important transparency and ultimately allows for more intelligent and responsible decisions to be made based on the predictions developed by spatio-temporal models.

3.3 Modeling Confidence

Having established exactly what is meant by confidence and where some inherent uncertainty will lie in both datasets and models, it is now possible to shift focus to the classification and construction of models for quantifying confidence. The three most critical aspects in this endeavor are the quantity of data provided to a given model, the quality of that data, and mostly importantly the level to which the inner workings of a spatio-temporal model are understood and accessible. Arguably, other factors such as how well a model can effectively represent a given spatio-temporal environment could also be included. However, these and any similar factors can effectively be captured under the grouping "quality of data" where the quality of data is analyzed with respect to the model. This once again points back to the importance of understanding why, or at the very least how, a given model makes predictions. Indeed, while the quantity and quality of data is directly related to the quality of the output provided by a given model, the ability for that model to make accurate predictions about the confidence or uncertainty associated with its prediction is directly predicated on knowledge about or obtainable from a given spatio-temporal model.

For example, consider the ability of humans to quantify their confidence in predictions about various activities or outcomes. When first watching a new sport, someone is unlikely to have confidence in their ability to predict various outcomes related to that sport. That is to say, one does not have the ability to watch for a few moments and then make an accurate prediction about what will or will not happen next. Conversely, a dedicated fan maybe be able to make may predictions with high confidence about what types of actions a team will take in a given situation, along with the respective likelihood of those actions' success. This is an example where the quantity of data is the largest determinant factor. Similarly, the importance of the quality of data that influences confidence can been be observed when two or more people attempt to have a conversation in a noisy environment. With a large amount of noise present, someone is more likely to have low confidence in their belief about

what was said to them. With low enough confidence, one may then ask for another to repeat what was just said in an attempt to increase their confidence by attaining higher quality data, or at the very least a higher quantity of data upon which to make a prediction. A similar analogy can be seen in the history of scientific advancement. As more refined models about the functioning of the physical world were developed in the fields of physics and chemistry, for example, so too were scientists able to make better and more confident predictions about the phenomena that they observed.

Thus, while anecdotal assertions about the factors influencing confidence predictions are self-evident, the forthcoming classifications, construction, experimental testing, and evaluation of methods for quantifying confidence in spatio-temporal models will put these assertions to the test. Because the quantity and quality of data collected are experimental variables, and not inherent to a given model, said variables will be examined later in the experimental section. This section instead will focus primarily on classifying and evaluating the different levels to which a given spatio-temporal model's inner workings can be considered to be understood or accessible.

3.3.1 Black Box

When nothing is known about the inner functions of a model, it is considered to be a Black Box. This is the simplest case from an outside observer's perspective. Input is given to the model, in the form of a time and location pair, and the model responds by outputting a value that corresponds to that time and location. The only additional piece of information that may be known is whether the model requires any training. In this case, a model may also be provided with any number of time, location, value tuples in order to prepare the model to make predictions. Since it is unknown in what manner the model uses the information provided for training, it is still considered to be a Black Box model.

Due to the limited amount of information known about a Black Box model, it is not possible to draw any conclusions or make any correlations between previous

and current predictions made by that model regardless of location. Doing so would require one to make potentially inaccurate assumptions about the workings of the model. For example, if the Black Box was using a type of multi-length, short-term memory system for predictions, similar to a modified version of a multi-map approach seen in [6] one could falsely assume a correlation with long-term behavior where one does not exist. Additionally, it is conceivable to imagine models that both may or may not draw correlations between different locations in the spatio-temporal model. For these reasons, it is not safe to make any assumptions about the output provided by such a model.

As a result of these rather restrictive assumptions, methods that can be used to quantify confidence for Black Box models are extremely versatile. They can be applied to any spatio-temporal model regardless of implementation. This makes them particularly attractive when it is desirable to compare a large number of models with a diverse range of implementations. Additionally, given the limited number of variables and lack of domain specific knowledge, a multitude of classic statistical models can easily be applied. This includes approaches such as confidence intervals[4, 5] or slightly more domain specific applications like the Nash Sutcliffe Model[1]. These approaches give a very good and broad understanding of the confidence one can have in the predictions provided by a method.

Pros

- Often relatively easy to implement
- Large amounts of existing research into suitable statistical methods
- Can be applied to any method (method independent)
- No prior knowledge must be known about the spatio-temporal model

Cons

- Predictions about confidence must inherently be very general
- Potential of oversimplification leading to poor performance

3.3.2 Grey Box

A Grey Box model is applicable whenever partial, but not complete knowledge is known about the spatio-temporal model in question. Given this broad definition this classification covers the most diverse set of spatio-temporal models. They are most commonly used when either the internal functionality of a given model is inaccessible or it would not be feasible to attempt to interpolate confidence from the internal state of a model. The former, inaccessibility, is most often encountered when attempting to develop a method for estimating confidence for a spatio-temporal model that is closed-source or for which the code is not readily accessible or modifiable. The latter, when it is not feasible to interpolate confidence, is a slightly rarer case though it can be encountered when dealing with some subsets of machine learning. In the example of a neural net, all of the edges and their corresponding weights may be accessible but, depending on the design of the model, it may not be possible to clearly or directly extract meaning from those weights.

Regardless of the reason, Grey Box models provide a litany of methods for quantifying confidence. In contrast to the Black Box approach, since one now has a certain amount of knowledge of the internal function of a given spatio-temporal world model, more assumptions can be made. For example, if it is known that predictions are based off data that is clustered together in weekly increments, similar to that of the work in [16], predictions about confidence could be made based off of the quantity of data captured. If there exists no data on Wednesdays, for example, confidence of predictions would be said to be low during that time. Conversely, if a large amount of data has been captured on Saturdays then the prediction confidence may be high. Additionally knowing that the model uses long-term data analysis to make its predictions, it would also be possible to analyze historical predictions given by the model, correlate the accuracy of those predictions with data density and perhaps data quality (with reference to model performance), and quantify confidence for future predictions based on these factors.

This example provides one of many ways that partial knowledge about how a

spatio-temporal model makes its predictions can be exploited to make estimates about the model's confidence. It is still possible to estimate confidence in this manner on multiple spatio-temporal models, but in comparison to the Black Box approach, these confidence estimating models can not be applied irrespective of implementation details. It is crucial that confidence estimating models that make assumptions about the underlying functionality of a spatio-temporal model only be applied to models that match the assumption made. Moreover, models may need to be tuned to match specific implementations. Using the previous example again, the confidence estimating model will need to be tuned to match the frequency at which data is analyzed, be it on a weekly or daily basis. Thus, while this approach on average may be able to make more specific and more precise predictions, it does require a certain amount of knowledge and tuning to function effectively. Finally, as knowledge of the internal functionality of the model may be incomplete, or worse yet inaccurate, there are still a certain number of unknown unknowns that may decrease the precision and accuracy of confidence estimates. As this incomplete knowledge is inherent to a Grey Box approach these issues unfortunately can not be compensated for with increased data quantity or quality.

Pros

- Confidence estimates can be made using experimental data or theoretical knowledge
- Able to be designed to work with any class or type of model
- Equally valid for closed and open source models

Cons

- Requires some amount of knowledge about existing model
- May need to be tuned to match implementation details
- A certain amount of unknown unknowns remain

3.3.3 White Box

When a given spatio-temporal model's inner workings are completely known, accessible, and comprehensible, a White Box model is the most fitting. Assuming these three requirements are met, it may be possible to modify the original model in such a way as to enable the model itself to provide confidence estimates. This is in contrast with the previously discussed Black and Grey Box models which are similar to an outsider looking in. With the internal functionality of the spatio-temporal model completely available, the White Box model can be viewed as a type self-reflection. In such a case the model can take into account it's current state, functionality, knowledge, and possibly past predictions to make the most informed confidence estimate possible. This in turn enables confidence estimates that are both, ideally, highly accurate and precise.

Despite the obvious benefits gained from complete internal knowledge, the White Box model is not without disadvantages. Since these implementations are often incorporated directly with the model they also contain any implicit bias or misinformed belief that the model may have. Thus, they are not immune to over confidence. Additionally, it is not always possible to meet all three of the aforementioned requirements with the latter two, accessibility and comprehensibility, being the most crucial. If the spatio-temporal model in question is publicly available as a library or binary file, but the source code is not accessible, it is highly unlikely that a White Box model can be implemented unless the internal workings of that implementation are exposed. Finally, given the direct connection between the confidence estimates and the inner workings of an implementation, these models are extremely specific and cannot be applied to a wide number of spatio-temporal models easily as is possible with the Black or Grey Box models.

Pros

- Complete knowledge of an entire system can be leveraged for improved confidence estimates

- Likely to provide predictions with the highest accuracy and precision

Cons

- Internals of a model must be accessible (open-source) and comprehensible
- A certain amount of inherent bias still remains
- Implementations are often unique and extremely specific to a certain model

4

Implementation Details

Having defined what is meant by "confidence" and established classifications that can be used to differentiate confidence estimating models, it is now possible to provide an example model for each of the three classifications. Context for these examples will be provided in the form of a real-world multi-robot scenario, ROPOD, and a matching spatio-temporal modeling method, Hypertime. The following sections will elaborate on the details and motivation behind the selection of ROPOD and Hypertime, as well as providing the implementation details for the different methods for estimating confidence in the predictions provide by Hypertime.

4.1 ROPOD

ROPOD is an active research project being worked on by a diverse group of academics and industry professionals. It is funded by the "European Union's Horizon 2020 research and innovation programme" and is focused on developing "ultra-flat, ultra-flexible, cost-effective robotic pods for handling legacy in logistics" [1]. The end goal is to provide a multi-robot system of small robotic pods that autonomously move supplies around a preexisting environment designed for and occupied by humans. Individual robots may be tasked with moving single smaller objects, while larger objects may require the coordination of multiple robots. Additionally, while each robot is responsible for localization and navigation, a central server is responsible

[1] http://www.ropod.org/

for planning and contains information about the status and performance of all the robots. The first deployment of this system is being targeted at a mid-size hospital in Frankfurt, Germany. This hospital, along with Hochschule Bonn-Rhein-Sieg, are being used as testbeds throughout the course of development to test new ideas and gather experimental data.

4.1.1 Why ROPOD?

ROPOD provides both common and unique technical challenges while simultaneously providing the tools and opportunities needed to overcome those challenges. The root cause for many of these challenges is the requirement that the robots be able to function in real-world human shared environments. The introduction and subsequent interactions with humans result in a deluge of complicated, but likely predictable, factors that are well-suited for confidence estimation. Additionally, since these factors are highly time and location dependent they are an excellent use case for spatio-temporal models. Finally, since ROPOD is a multi-robot system, it provides more opportunities for robots to collect data which facilitates improved confidence models.

For example, a common task for ROPOD is to move freshly arrived supplies from the loading dock in the basement to their final destinations at various rooms throughout the hospital. Another common task is moving large hospital beds between different rooms in the hospital that may be on separate floors. In both of these tasks, the robot(s) must traverse multiple hallways filled with a varying number of people. During peak hours, it may be better to take a longer route to avoid large groups of people that may otherwise slow down or hinder the movement of goods. Additionally, when the robot(s) must switch floors, they must call an elevator. If this elevator contains a large number of people, it may not be possible for the robot(s) to board the elevator with their goods. It is, therefore, conceivable that some locations and times may be better for moving certain goods. In order to be able to predict when and where to travel, it would therefore be beneficial to be able to model the fluctuation in arrival and travel time of elevators, the travel time between different

areas, and finally the number of obstacles encountered in a given area.

With respect to confidence estimates, having a confidence value associated with predictions can quantify and inform the planning made by the central server. If confidence estimates are extremely low at a given time or location, robots that are otherwise not doing anything can be tasked to roam about those areas to collect data in order to improve predictions. Alternatively, tasks that are extremely high priority could be scheduled at an earlier time than otherwise necessary in order to maximize the likelihood of success. This could be beneficial if, for example, weekly shipments of a certain medicine must always be delivered before 9:00 in the morning. If those shipments arrived a day before, instead of attempting to arrive exactly at 9:00 when there may be a morning rush of employees arriving, the delivery could be prioritized and completed a few hours before when fewer employees were around. This mix of situational and material support factors make ROPOD an ideal model case.

4.2 Hypertime

While an in-depth explanation of Hypertime has been provided in the Related Work section, for the sake of convenience, a short overview of the method will be provided here as well. Hypertime[19] is a spatio-temporal modeling method that relies on the assumption that changes in an environment happen in both periodic and predictable fashion. It was specifically designed for long-term use in human shared environments, as these behaviors are often present in these types of environments. Data for the models is gathered as the robots go about their daily tasks. This data takes the form of a location, time, and information about a behavior. This tracked behavior could be the number of people at a specific location and time, or the amount of time it took to traverse a specific hallway. A Hypertime model can then be trained using this spatio-temporal data.

With respect to the implementation used in this paper, Hypertime is provided with multiple tuning parameters at training time. Two of the most important tuning parameters are the number of clusters into which spatio-temporal data will be grouped and the maximum period for which the data will be analyzed for temporal variance. During training, Hypertime analyzes the training data for any temporal

patterns, the most dominant of which will be selected to perform a hyperspace extension. Essentially, training data is augmented with additional temporal information increasing the dimensionality of the training data. After each hyperspace extension data is clustered resulting in multivariate Gaussian distributions which are then used for predictions. Predictions can be made by either sampling all distributions at a given time and doing a weighted sum, or by selecting the prediction with the highest probability. Training Hypertime is an iterative process where hyperspace extensions are repeatedly added with new temporal periods until the error of the model created is greater than the previous model.

After training, these models can estimate a behavior at a given time and location. In a simple, 1-dimensional location example, a model could be trained using the length of time it took to traverse a hallway at various times. By providing Hypertime with a given time, it would return an estimate for how long it would take to traverse that hallway at that time.

4.2.1 Why Hypertime?

Hypertime provides an excellent use case for designing and comparing methods for estimating confidence. Not only does it obviously meet the qualifications for a Black Box model, but both Grey and White Box models can be designed for it as well. The internal functionality is both fully comprehensible and understandable, as it is based on a combination of classic statistical and mathematical methods, and fully accessible, as it is open-source and well-documented in various published academic papers. Given that Hypertime models represent behaviors of a periodic nature, predictions will begin to repeat after a known maximum period. With this knowledge, a Grey Box model can be designed. With respect to the White Box model, since data is grouped into one of a set number of predetermined clusters, Hypertime can be modified to provide the variance of those clusters or indicate which cluster a given prediction belongs to for later statistical analysis.

Not only is Hypertime well-suited for demonstrating methods for estimating confidence, it is also ideal for use with ROPOD. As previously mentioned, Hypertime

has been designed for, and used in, a number of long-term indoor applications as seen in [19, 20]. Furthermore, previous studies of the suitability of various spatio-temporal modeling methods for use with ROPOD have shown Hypertime consistently outperforming other methods [6].

4.2.2 Modified Version of Hypertime

Since Hypertime is open-source, but does not currently produce confidence estimates, a method for individualized, prediction-specific, estimates was explored. This would be of great value as, in theory, confidence estimates for each prediction could be unique to each spatio-temporal location. The original idea was to use the conditional distribution obtained from the multivariate Gaussian distributions, or the clusters, present inside of Hypertime. With the knowledge that the Gaussian distributions are N dimensional and $N-1$ of those dimensions are defined by time t when requesting a prediction, the final result would be a 1-dimensional Gaussian with a single mean and variance. The mean would therefore represent the prediction at time t and the variance its confidence. The mathematics behind this operation can be found in Equation 4.1 and more information can be found in [12].

Given a 1-dimensional matrix:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \text{ with respective length } \begin{bmatrix} 1 \\ (N-1) \end{bmatrix}$$

The resulting means will be:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \text{ with respective length } \begin{bmatrix} 1 \\ (N-1) \end{bmatrix}$$

and a covariance matrix given by:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \text{ with sizes } \begin{bmatrix} 1 & 1 \times (N-1) \\ (N-1) \times 1 & (N-1) \times (N-1) \end{bmatrix}$$

then with $\mathbf{x_1}$ conditioned on $\mathbf{x_2} = \mathbf{a}$ the resulting mean and covariance matrix are:

$$\overline{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}_2\right)$$

$$\overline{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$$

Conditional Distribution of a Multivariate Gaussian Distribution for use with Hypertime (4.1)

This modified version offers multiple benefits compared to the previous method of prediction. The previous methods for prediction relied on sampling the clusters a given number of times. The sampling method both forces a minimum and maximum prediction value based on the bounds of the samples taken, and is either slow and inefficient if the sampling is high-resolution, or tends towards possible inaccuracies if low resolution samples are taken. In contrast, this new method produces a single mean and its range and accuracy is determined entirely on the Gaussian distributions present in the model.

Despite the numerous benefits of this modification, one issue persists. While the resulting 1-dimensional distribution has a mean that varies with respect to the values used to drive the conditional distribution, the variance does not. Pursuant to the inherent implications of the math presented in [12], while conditioning $x_2 = a$ results in a different variance, the actual value of a does not affect the new variance. It is merely the act of conditioning a that results in a different variance. That is all to say, while the mean will change per prediction, the variance will always be the same for a given cluster. Regardless, in the modified version of Hypertime used in this work, both the variance and the mean are provided when making predictions. While it is not possible to provide confidence estimates unique to every prediction, it is possible to use the variance directly, or to use it to identify which cluster was used to make that specific prediction. In the latter case, additional statistical analysis

can be done after training to create the desired confidence estimates.

4.3 Confidence Models

With the knowledge that Hypertime will be used to model behaviors present in long-term indoor environments for ROPOD, specifically those of a hospital or university, it is now possible to develop confidence models for the corresponding predictions. The behaviors that will be modeled in the later experimental section are the wait time/ride duration of an elevator and the travel time for various hallways. These behaviors are dissimilar to some of the previous work covered, specifically that of FreMEn or machine learning classification problems. Indeed, FreMEn was used to predict the likelihood of an event happening, making it essentially a two class classifier. While the number of possible predictions for classification problems are limited, or at the very least discrete, the aforementioned behaviors are continuous and thus pose an interesting problem when attempting to develop models to encapsulate confidence. It is for this reason that these confidence models should not produce predictions that quantify how likely a given prediction is to be correct, but rather an likely upper and lower bound on duration of a given behavior.

The lower and upper bounds for predictions will be obtained by grouping the results of training data into subcategories and analyzing Hypertime's predictive performance on the training data using various statistical methods. The number and type of subcategories the training data will be divided into will be determined by the assumptions made about how Hypertime makes predictions. This corresponds to the methods covered in the Methodology section. The driving factor for dividing predictions into subcategories is that there is an assumed correlation between both the amount and quality of information known about a specific spatio-temporal location and the respective quality and confidence of its corresponding prediction. In this way, by dividing the predictions provided by Hypertime in to subcategories and analyzing their results it will be possible to provide spatio-temporally unique confidence estimates.

In the simplest case, all the results obtained from training can be divided into

predictions that either over- or underestimate the true value. By doing statistical analysis on these two categories, it is possible to provide a generic upper and lower bound for all predictions. Models can then further divide these results into subcategories. The division will be made based on the information that model assumes to know about a prediction provided by Hypertime. Each one of these subcategories will contain its own upper and lower bounds to estimate the confidence of a given prediction for any future prediction that corresponds with that properties of that subcategory. Details on the number of subcategories and the manner in which results are divided will be covered later in this section. The statistical methods used to provide the upper and lower prediction bounds are as follows.

Two methods for providing prediction bounds have been selected, a variant of confidence intervals and a range provided by the standard deviation that will be referred to as two sigma. For each method of dividing training data into subcategories, a confidence estimation model will provide two sets of predictive bounds. The equations for these bounds can be seen in Equation 4.2 and Equation 4.3. Both of these equations have a tuneable parameter, namely the Z value for the confidence interval and the number of sigmas, for two sigma. While a specific Z and sigma value are both important and application specific, values targeting roughly 95% have been selected for these experiments. 95% was selected to capture the majority of behavior while leaving room for expected noise and outliers. The predictive bounds will be generated using the results obtained from the training dataset. The error for each training data point, i.e. the distance between the given prediction and its true value, will be used to obtain both the average error and the standard deviation for a given model trained on a given dataset.

$$\begin{aligned} \mu_u &= \bar{x}_u + Z\frac{\sigma_u}{\sqrt{n}} \\ \mu_l &= \bar{x}_l + Z\frac{\sigma_l}{\sqrt{n}} \end{aligned} \tag{4.2}$$

Confidence Interval Bound Equation 4.2 - Where μ_u and μ_l respectively represent the upper/over and lower/under prediction groupings, $\bar{x}$ is the sample mean error, Z is the number of standard deviations, and $\frac{\sigma}{\sqrt{n}}$ is the standard error of the mean.

$$\begin{aligned} \mu_u &= \bar{x}_u + 2\sigma_u \\ \mu_l &= \bar{x}_l + 2\sigma_l \end{aligned} \tag{4.3}$$

Two Sigma Bound Equation 4.3 - Where μ_u and μ_l respectively represent the upper/over and lower/under prediction groupings, $\bar{x}$ is the sample mean error, and σ is the standard deviation of the error.

4.3.1 Black Box

As mentioned in Methodology section, the Black Box model trades accuracy for utility and ease of application. For this model, all predictions will be grouped together into one general classifier. This will result in a large sample size n and a single constant upper and lower bound for all predictions.

4.3.2 Grey Box

Using the knowledge that Hypertime assumes spatio-temporal behaviors to be cyclic and contains a maximum period for which it will attempt to represent those behaviors, it is possible to group predictions temporally. Predictions can be divided into n uniform groups of length $\frac{T}{n}$ seconds, where T is the length of the maximum period analyzed (in seconds). In this method, it is assumed since predictions are temporally dependent, and representation is cyclic/periodic, errors are also likely to be cyclic. For example, if the maximum period, T, was one day, n could be set to 24, thus dividing each day into 24 one-hour long sections. The working assumption would then be that times of higher variability, e.g. lunch time, would exhibit a larger confidence bounds than those of less variability e.g. late at night or early morning. The exact size of both n and T are explored in the benchmarking part of the experimental section.

4.3.3 White Box

As the inner workings of Hypertime are completely known, accessible, and comprehensible, it is possible to use the internal knowledge of the method to produce

less generalized and more informed confidence estimates. Specifically of note is the fact that Hypertime uses multivariate Gaussian mixture models to cluster together similar spatio-temporal data points. It is feasible that each one of these clusters may directly correlate to prediction error. For this reason, predictions can be grouped by which cluster was responsible for the corresponding prediction. The number of groups c directly correlates with the number of clusters Hypertime uses, which is a configurable parameter. The original authors found the number of clusters to be dependent on the topology of the environment and corresponding data, but used three clusters for benchmarking and found this to be acceptable for their use case. [20] The number of clusters desired for the ROPOD use case will be explored further in the experimental section, but regardless, it is unlikely that the number of clusters will be of any significant size and are unlikely to be greater than four or five.

White X Grey Box

With a relatively limited number of clusters by which to group predictions, it may be desirable to delineate the predictions further. In an effort to increase prediction accuracy, grouping by clusters can be combined with the temporal clustering used in the Grey Box method. This will produce a maximum of $c \times n$ subcategories. While this may produce more accurate prediction estimates, it also runs the risk of being overly specific and may face issues of small sample sizes resulting in oversized or invalid prediction estimates. These problems are highly correlated with both the size of the training set and the spatio-temporal distribution of the data contained in the training set.

5

Experimental Setup

Having established the various methods for quantifying confidence in spatio-temporal models in the previous section, it is necessary to design experiments to test their performance. Datasets for these experiments will be comprised of real-world data that has been either collected by a ROPOD or that is directly relevant to the ROPOD project. In either case, the two datasets that are used provide benefits and unique challenges emblematic of those faced when attempting to quantify confidence in spatio-temporal models. The datasets used for these experiments, as well as other related datasets, can be found on GitHub[1,2].

5.1 Real-World Datasets

5.1.1 Agaplesion Hospital Elevator Dataset

The number and duration of calls made to a given elevator was tracked over a period of roughly two months. Data was obtained by connecting directly to the CAM interface of the elevator, and thus the logging is both complete and accurate. The specific elevator referenced herein is located in Agaplesion Hospital in Frankfurt, Germany and provided access from the basement of the hospital to the top floor. This elevator will be used for the ROPOD project as it is the closest

[1]https://github.com/emassey2/ropod-stm-dataset
[2]https://github.com/anenriquez/ropod_rosbag_processing

elevator to where the supplies in the basement are stored. Furthermore, it serves as a direct route to the rest of the hospital for the ROPOD robots. While other routes through the hospital exist, this elevator is the only one with which the ROPOD robots can communicate and thus may act as a bottleneck during times of high traffic.

The main goal of this dataset is to provide training data for models which predict the best time to use the elevators in order (a) to minimize wait time at the elevator and (b) to provide a rough estimate of how long before its estimated arrival at the elevators a robot should issue a call such that upon arrival the robot does not wait for the elevator to arrive. As this data was collected directly from the elevator, it is both high-volume and has very little noise. Observations for this dataset were made between September 17th, 2018 and November 23rd, 2018, resulting in just over two months of data. An overview of the dataset can be seen in Figure 5.1.

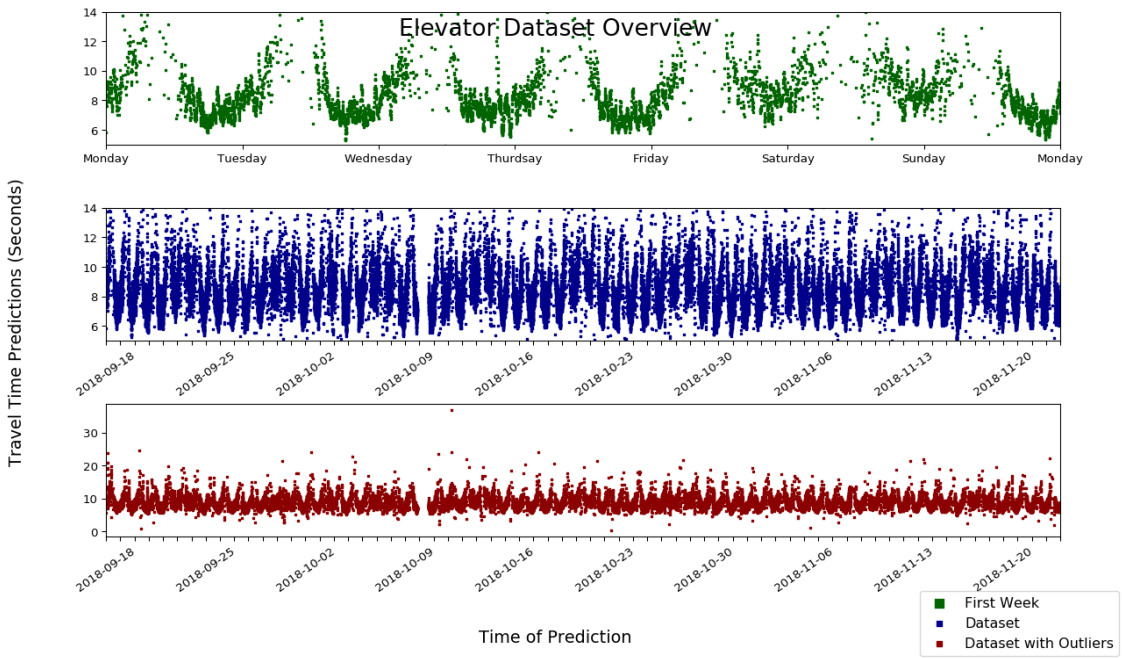

Figure 5.1: Agaplesion Hospital Elevator Dataset Overview

5.1.2 H-BRS Hallway Dataset

The second dataset used for evaluation contains travel times between various areas inside the Hochschule Bonn-Rhein-Sieg in Sankt Augustin. A ROPOD robot was tasked with traveling between seven nodes inside the university a few times per day, multiple days per week. The travel times between the various waypoints were calculated and stored, providing the data for the dataset. As the density of people, among other variables, changed throughout the day and week, so too did the travel time between the various waypoints. Data about the average and maximum number of obstacles encountered per edge was also tracked to provide comparison with travel times. The data was obtained from the start of September 2019 through the end of February 2020, providing data over approximately six months. Data collation began shortly before the semester began and continued through the holidays. Therefore, it represents both fluctuating crowds that would slow down the travel time of the robot and times of very low activity. Since the ROPOD robots are still under active development, the robot operation and data collection required human supervision. Because of this, that dataset can be quite sparse, especially in comparison with the Elevator dataset referenced previously. Additionally, data is predominantly available during working hours on weekdays as that is most frequent time that experiments with the robot were conducted. Figure 5.2 is a map of the relevant part of the university with the nodes labeled. Note, only four of the possible seven nodes are displayed. These four nodes result in four edges, each emblematic of different behaviors. The routes taken to collect the data and the edge-specific behavior is covered below.

Data Collection Routes

Multiple pre-planned routes were used to collect data for this dataset. For this reason, some hallways will have more information than others. The main pre-planned route involved traveling in order through the four hallways, creating a square. These four hallways will be the edges that will be used for evaluation.

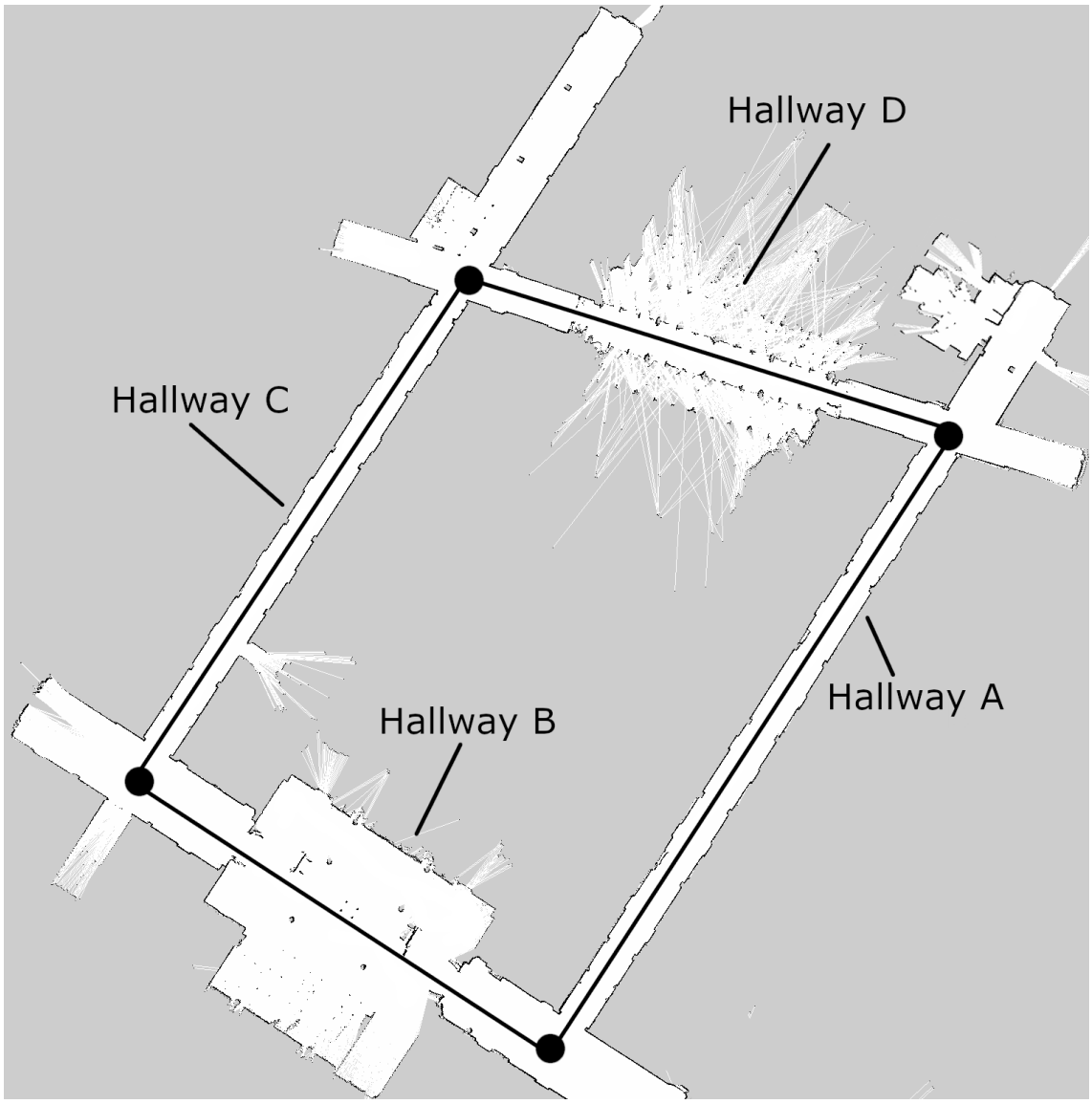

Figure 5.2: Waypoint nodes for the H-BRS Hallway Dataset

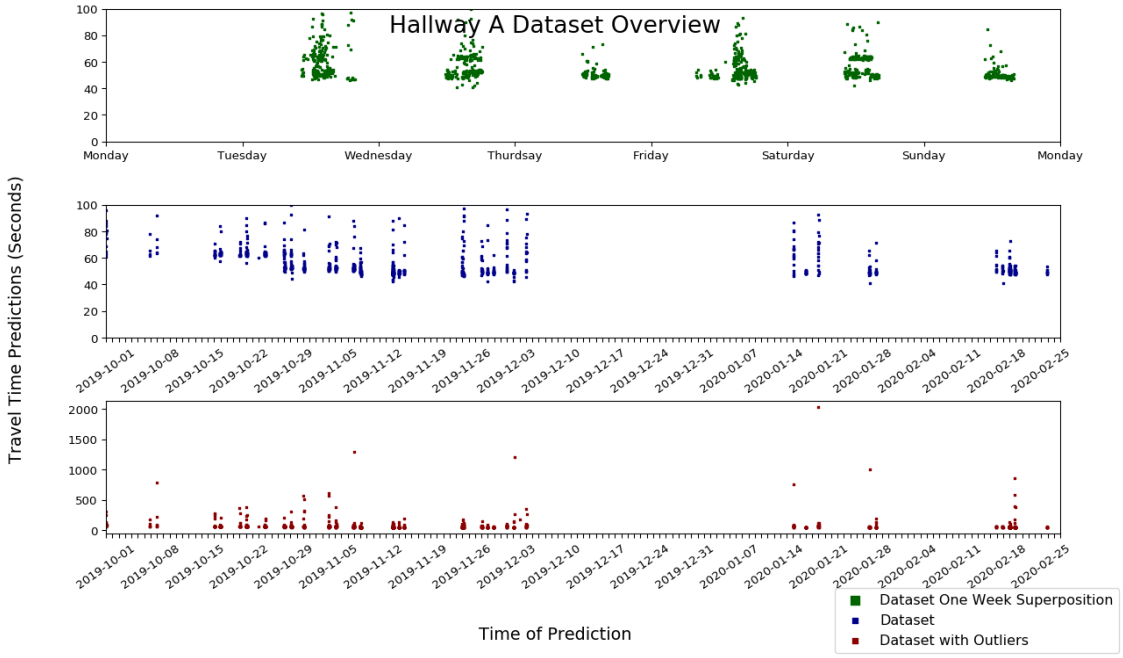

Figure 5.3: H-BRS Hallway A Dataset Overview

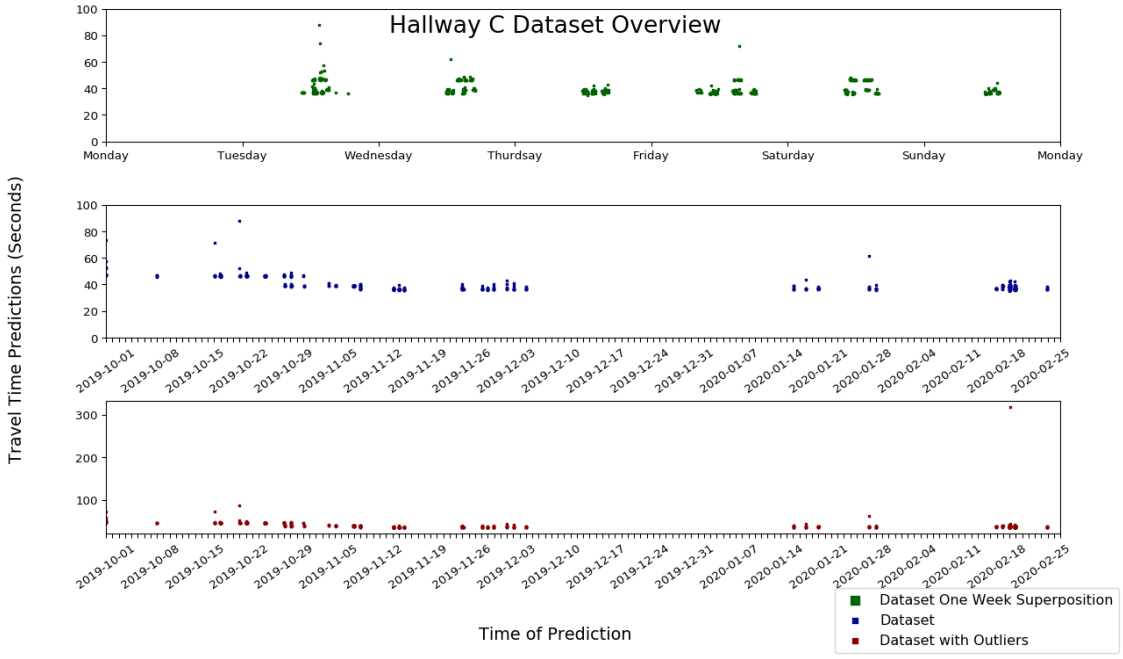

Figure 5.4: H-BRS Hallway C Dataset Overview

Hallway A & Hallway C

Both of these two edges represent medium-length hallways comprised mostly of university classrooms. Light traffic is to be expected during the majority of the dataset with occasionally heavier traffic in between classes. Looking at Figure 5.4, Hallway C in particular has little noise and variation. Hallway A, on the other hand, has more data relative to the other edges, but this came at a cost. Hallway A is used for some additional preprogrammed routes as well as occasional developer testing. These additional uses provided more data, but also introduced more noise. The extra noise and data can be clearly seen in Figure 5.3. These two edges will provide a clear contrast of how the volume and quality of data can affect predictive confidence.

Hallway B

This edge passes through a fairly open area filled with tables where students often congregate to study and work together. For this reason, higher traffic was to be expected due to the larger, more dynamic range of potential obstacles encountered. An overview of the data captured for this edge can be seen in Figure 5.5.

Hallway D

As demonstrated on the robot generated map, this intersection posed interesting problems for the robot. The problem of traversing this edge came not from dynamic obstacles in the form of people, but rather noise introduced by the robot's sensors. This hallway is comprised of glass walls which are challenging for the onboard sensors as they can produce false positives for object detection. As if this were not enough of a challenge, at the end of the hallway there is an automatic door that did not always detect the robots' presence and therefore occasionally had to be triggered by a human. This introduced variations in the amount of time it took to travel this edge. Some of the resulting outliers can be seen in Figure 5.6. These factors combined to create a novel and difficult edge to analyse and traverse.

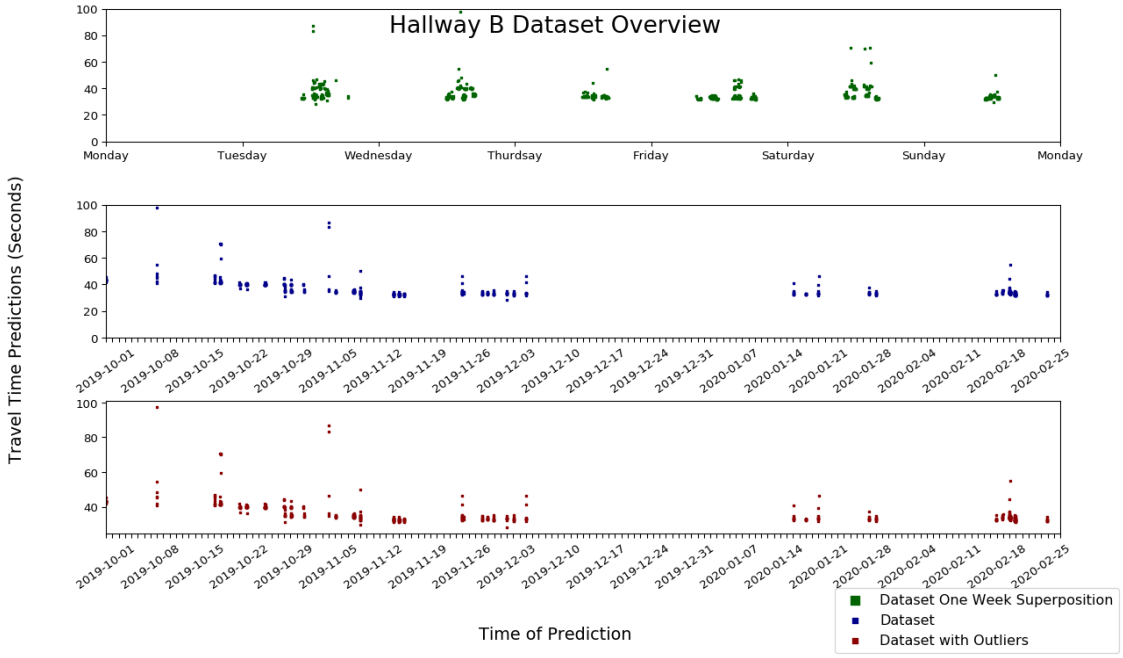

Figure 5.5: H-BRS Hallway B Dataset Overview

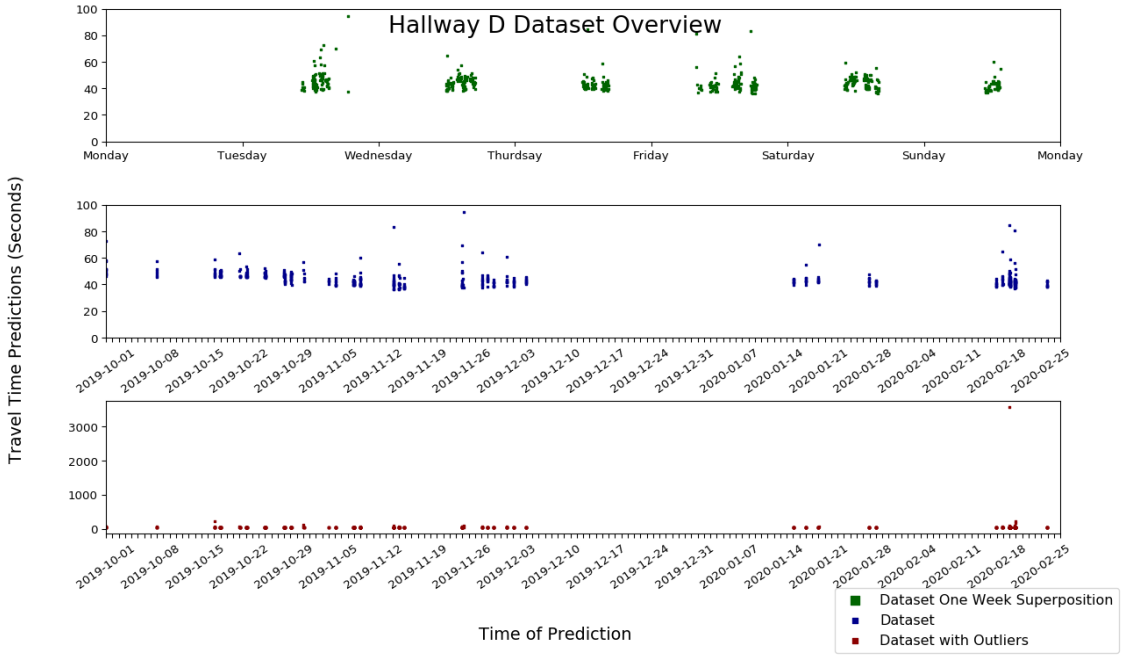

Figure 5.6: H-BRS Hallway D Dataset Overview

5.2 Evaluating Confidence

In order to evaluate the proposed methods for estimating confidence, performance metrics must be established. As it is the confidence or bounds of a prediction that is under test, and not the prediction itself, the performance metrics will be largely focused on said bounds.

5.2.1 Accuracy of Confidence Predictions

The most straightforward way to evaluate confidence predictions is to simply observe how often the observed data points were contained within the bounds of a prediction. This value is given by the number of correct predictions over the total number of predictions: $A_p = \frac{P_c}{P_t}$

5.2.2 Invalid Predictions

Given that the ability to provide confidence estimates is dependent on the performance of the training data, it may not always be possible for some models to provide confidence estimates. This is particularly the case for models that maintain many subdivisions of data and for datasets that are either sparse or that have a significant difference between training and test data. Regardless of the reason, when it is not possible to provide both a lower and upper bound for a confidence estimate, this will be considered an invalid prediction. The number of invalid predictions will be tracked as a means to compare how often different models failed to provide valid predictions.

5.2.3 Size of Bounds

The difference between the upper and lower limit of a prediction will be considered the size of the bounds of a prediction. The size of these bounds can be used to measure a model's confidence in its prediction. Models with smaller bounds are considered to be more confident, as they have less room for error. Looking at the average size of a model's bounds can provide insight into the general utility of a

model as well. For example, while by itself a model accuracy of 100% may seem impressive, if unreasonably large bounds are given it may not be as valuable as the accuracy may suggest. If the model bounds cover orders of magnitude more range than any observed behavior, they are not accurately describing reality and are poorly representing the underlying behavior that it is attempting to model. Conversely, bounds that are too small may equally result in estimates that do not accurately capture the underlying behaviors.

5.2.4 Magnitude of Inaccuracy

In the case of an inaccurate prediction, it can be useful to evaluate how incorrect the prediction was. Although the overall accuracy of the predictions provided by a given model encapsulates the general performance of a model, it is far from all-encompassing. In the event that two models have roughly similar accuracy, it may prove valuable to compare exactly how inaccurate they were when their predictions were incorrect. This metric will be referred to as the Magnitude of Inaccuracy or $|Inacc|$ and is obtained by taking the root mean square (RMS) of the distance an actual prediction is from the given prediction bounds. As this metric is only used to measure how inaccurate a prediction is when it is outside the prediction bounds, a measurement is only taken when a given prediction was inaccurate. The formula for this metric is given in Equation 5.1 where n is the total number of inaccurate predictions, i denotes an individual prediction, g is the ground truth/actual value, and μ_u & μ_l are the upper and lower prediction bounds, respectively.

$$|Inacc| = \sqrt{\frac{1}{n}\sum_{i=1}^{n} min(|\mu_{ui} - g_i|, |\mu_{li} - g_i|)^2} \tag{5.1}$$

5.2.5 Root Mean Square Error for Correct Predictions

While not the main focus of the experiments covered in this work, in the case that a prediction was correct, it may provide insight to know how correct it was. This is, in essence, the opposite of the magnitude of inaccuracy. While ideally these two metrics should have an inverse relationship, it may be interesting to note their exact

relationship to one another. When and how much the RMSE increases or decrease can also provide insight into the distribution of the underlying data. The equation for this metric is shown below in Equation 5.2 where g_i is the ground truth/actual value, and pc_i is a given correct prediction.

$$\text{RMSE of Correct Predictions} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(g_i - pc_i)^2} \tag{5.2}$$

5.3 Experimental Design

5.3.1 Experiment 1: Hypertime Parameter Selection

Experimental Motivation

The selection and tuning of various Hypertime parameters is necessary to achieve optimal performance. In addition to these parameters, there are multiple different methods for producing predictions. For this reason, it is desirable to run a preliminary experiment using different combinations of these parameters to select the best combination for ROPOD. Selecting the most optimal combination of parameters will decrease the noise, inaccuracy, and other variables for the forthcoming experiments and provide a useful guideline for using Hypertime for the ROPOD project going forward.

Experimental Details

The main parameters under test for this experiment are the number of clusters used, the maximum period of time for which periodicities can be modeled, and the three methods for making predictions: sampled mean, sampled mode, and conditional probability mean. Although a small number of other parameters exist for tuning, these three play the dominate role in tuning Hypertime. For benchmarking, the use of three clusters has previously been found to perform well [20]. Using three as a midpoint, benchmarks of Hypertime will be run with one to five clusters. Given the nature of the human-centric environments in which ROPODs will be operating,

behaviors are likely to be periodic on the order of hours, days, or weeks. For this reason, two maximum periods have been selected; 24 hours and one week. While it is also conceivable that behaviors of a larger period may be present, such as on the order of months or years, the datasets used for testing lean towards emulating temporal behaviors on the order of days and weeks. Results found with the selected maximum periods are assumed to extrapolate into longer time frames as the same underlying principles are at play. Finally, the sampled mean, sampled mode, and conditional probability mean (referred to hereafter as simply mean) will be the three prediction methods evaluated during these experiments.

5.3.2 Experiment 2: Elevator Dataset

Experimental Motivation

The Elevator Dataset will be used with the goal of establishing a baseline, best case scenario for the performance of the various confidence estimation methods. The high volume, density, and accuracy of the data present in the Elevator Dataset provides an optimal environment for the initial benchmarking of the confidence estimating methods under test. These tests will give an upper bound of performance for each method, ideally producing small confidence bands during times of low human activity and larger confidence bands during periods of high traffic. Regardless of the time, prediction accuracy should be relatively high and any inaccuracy should be of minimal distance from the bounds. This experiment will act as a reference point when analyzing the comparatively sparse and noisy hallway dataset in the next experiment. Additionally, comparisons can be made between the two experiments to determine the relative effect data density has on prediction quality.

Experimental Details

Two versions of this experiment will be run, with variation only in the training data. In both experiments the training data will be roughly the first two thirds of data, approximately 45 days worth, with the last third, approximately 26 days, being used

as test data. The difference between the two datasets lies in the amount of data provided for training. The first test will contain all the data available, roughly 30,000 data points, while the second will only contain 135 randomly selected data points. This corresponds to roughly three data points per day. This number was specifically selected to imitate the limited amount of data that could be obtained naturally by robots traversing the hospital. Both datasets will be used to test the performance of the Black Box, Grey Box, White Box, and White X Grey Box confidence estimate models using the Hypertime parameters determined in the previous experiment.

5.3.3 Experiment 3: Hallway Dataset

Experimental Motivation

Comparing the results of the various hallways in the Hallway Dataset obtained from H-BRS establishes a lower bound for the performance of the various confidence estimation methods. The imperfect, noisy, and low-density data present in this dataset serves as an excellent contrast to the previous dataset. The slight differences between the data density and noise for each of the respective edges also provides potentially interesting edge cases and areas for comparing and contrasting expected with observed results.

Experimental Details

This experiment will consist of training data provided between the months of September 2019 and the end of February 2020, with test data coming from the last third of the dataset. The main behavior under test will be the travel time along the edges, specifically the four hallways. In a similar fashion to the Elevator Dataset, the first two thirds of data will be used for training while the latter third will be used for test. Again, the four confidence estimate models, the Black Box, Grey Box, White Box, and White X Grey Box model, will be under test.

5.3.4 Experiment 4: Multi-Model Fusion Proof of Concept

Experimental Motivation

This experiment proves the feasibility of the combination or fusion of multiple spatio-temporal models or methods for confidence estimation. With the previous experiments focused on analysing the performance of various confidence estimating methods, it may become clear that certain methods excel under certain conditions while others may excel under different circumstances. Furthermore, it is possible to imagine other spatio-temporal models that also provide confidence estimates that may outperform Hypertime in some instances. It would therefore be beneficial to design a method to dynamically choose the "best" prediction during run-time. For this reason, this experiment has been designed as a proof-of-concept to demonstrate how this could be achieved and to hint at the type of benefits that could be expected from using such a method.

Experimental Details

The sparse version of the Elevator Dataset will be used for this proof of concept. The same test and training data will be used for ease of comparison. The four confidence estimating models used in the previous experiments will be reused for this section as well. Three methods for selecting the "best" prediction will be under test, two greedy methods and one hybrid. Of the two greedy methods, one will always select the model whose data grouping at the time had the best accuracy during training. The other greedy method will select for minimal bound size. The hybrid model will attempt to combine these two methods by multiplying the bound size by the inverse of the accuracy. In this way, a bound that was theoretically 100% accurate would remain the same size, and a bound that was accurate only 50% of the time would double in size. The hybrid model will then select between the methods based on minimal bound size. These methods will ideally provided a type of multi-model fusion that will outperform any other single model. The results of this test will then be compared with that of the previous Sparse Elevator Dataset experiment as a reference.

6

Experimental Results

6.1 Experiment 1: Hypertime Parameter Selection

The results of this experiment serve the dual purpose of selecting the proper parameters for Hypertime for use in later experiments and setting a baseline performance of Hypertime's prediction capabilities with an optimal dataset. While the dataset does have a somewhat wide range of possible values, ranging from near 0 to over 20, all variations perform well, with none having a RMSE over 2. This performance is attributed to the relatively low noise, high volume and density of training data, and easily discernible cyclic patterns present in this dataset. While all variations of Hypertime performed well, there are clear trends visible when viewing the average RMSE of different parameter categories. The results of each run are visible towards the end of this section in Figure 6.5 with categorical breakdowns covered in the following subsections.

6.1.1 Maximum Period

The strongest trend observed across all runs of the parameter experiment is the performance increase when using a longer maximum period. This is clearly observable in Figure 6.1. In fact, not a single shorter period model outperformed the longer variants. This behavior is consistent with the Elevator Dataset. This dataset

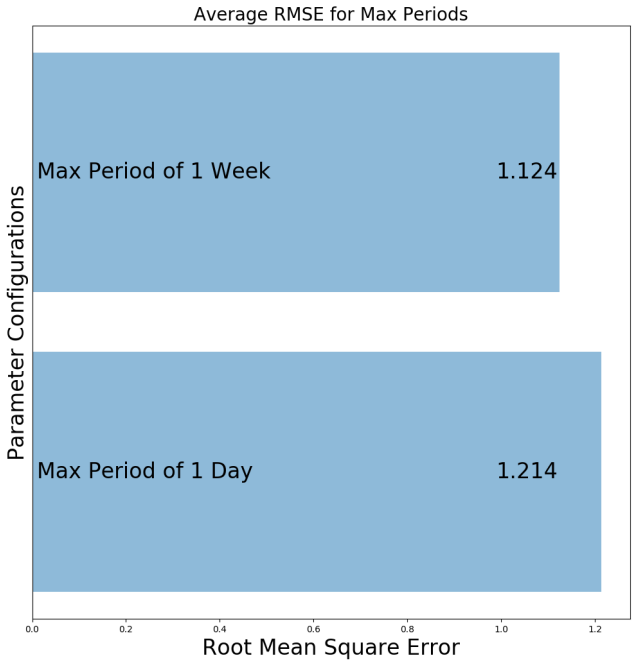

Figure 6.1: Combined Average RMSE for Varying Hypertime Maximum Period Configurations

strongly exhibits cyclic behaviors on the order of a week with there being a clear delineation between weekdays and weekends. This is observable both in the training data and in the predictions provided by the trained models as seen in Figure 6.4.

6.1.2 Prediction Methods

While not as stark as the results for maximum period, there are still clear trends present in the results of the average performance of the various prediction methods. As expected, the mean provided by conditional probability provides the most accurate predictions, followed closely by the sample mean. This makes sense, as they are close to the same estimate, but obtained through different methods. The main difference between the two is that the sampled mean accounts for values provided by multiple clusters while the conditional probability mean is only the mean provided by the most likely cluster. The mode, on the other hand, lags behind both of these two estimators as seen in Figure 6.2. Furthermore, looking again at Figure 6.5 the mode can be seen trailing the other methods, sitting last in both the week and day variants.

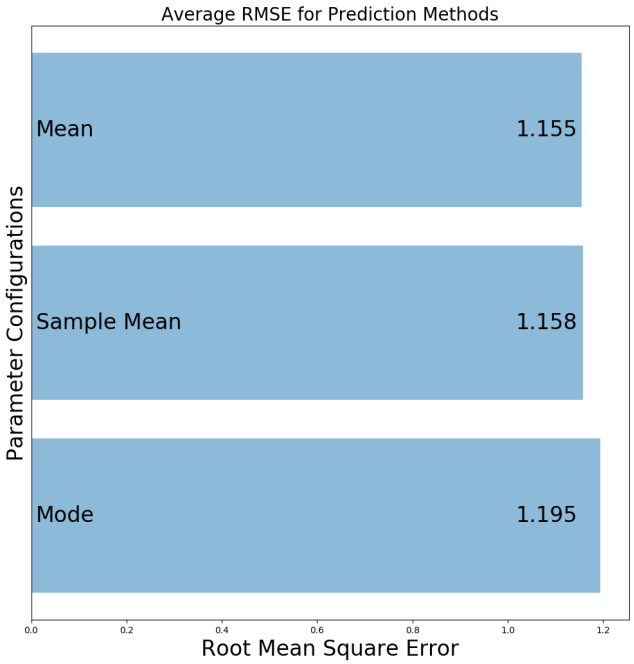

Figure 6.2: Combined Average RMSE for Varying Hypertime Prediction Methods

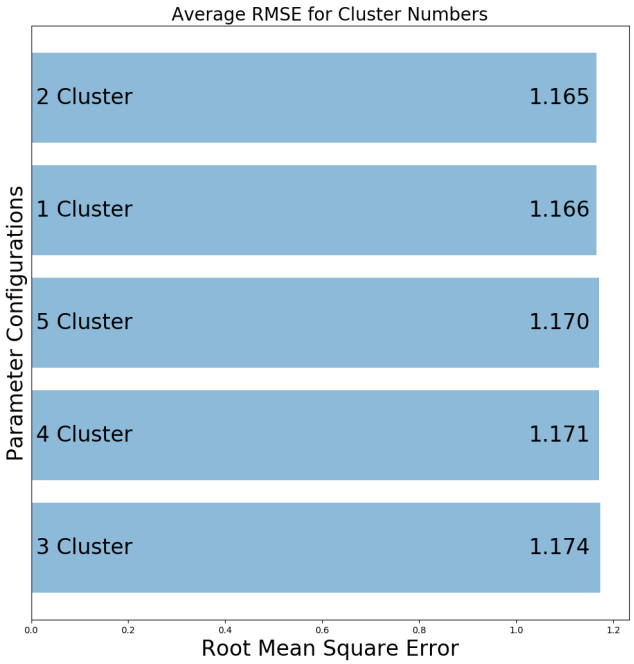

Figure 6.3: Combined Average RMSE for Varying Hypertime Cluster Numbers

6.1.3 Number of Clusters

The number of clusters used to group the spatio-temporal data appears to have had the least significant effect on performance. However, that is not to say that this parameter is irrelevant or does not provide insight. Figure 6.3 provides an overview of the performance with varying numbers of clusters used. While these average results reaffirm that there isn't a major impact on performance, Figure 6.4 clearly illustrates the differences between the methods. It contains a week of predictions from the conditional probability mean prediction method with a maximum period of one week. From this view, a few interesting behaviors can be observed. The 4- and 5-cluster variants contain the largest amount of noise or jitters in their predictions, which indicates that the models may have overfit. More interestingly, the 3-cluster variant also appears to have overfit but in a somewhat unexpected way. Its predictions are less continuous than the majority of other models. This behavior is particularly observable when looking at the transition periods when moving from periods of high traffic into periods of low traffic. It is suspected that these two sections of spatio-temporal data have been grouped into separate clusters, causing a sharp divide when transitioning between the two periods of two periods. For this reason, this behavior is present in all models with a cluster number greater than 1, but nowhere as stark as the 3-cluster variant. Finally, while the 1-cluster model performs surprisingly well, it does appear to have oversimplified. Its smooth sinusoidal curves with slight peaks during the weekend suggest that its likely using only two or three periods to make predictions. This is in contrast to the models with higher numbers of clusters, which suggests that models that use a higher number of clusters likely result in more behavioral periods being used to make predictions.

6.1.4 Chosen Hypertime Parameters

Combining the knowledge and observations made above, a 5-cluster Hypertime model using the mean prediction method with a maximum period of one week will be used. Note that while this is not strictly the best performing method with respect to its accuracy measured by the RMSE, it is still one of the top five best methods and has a difference of less than 0.001 seconds compared to the top performing method.

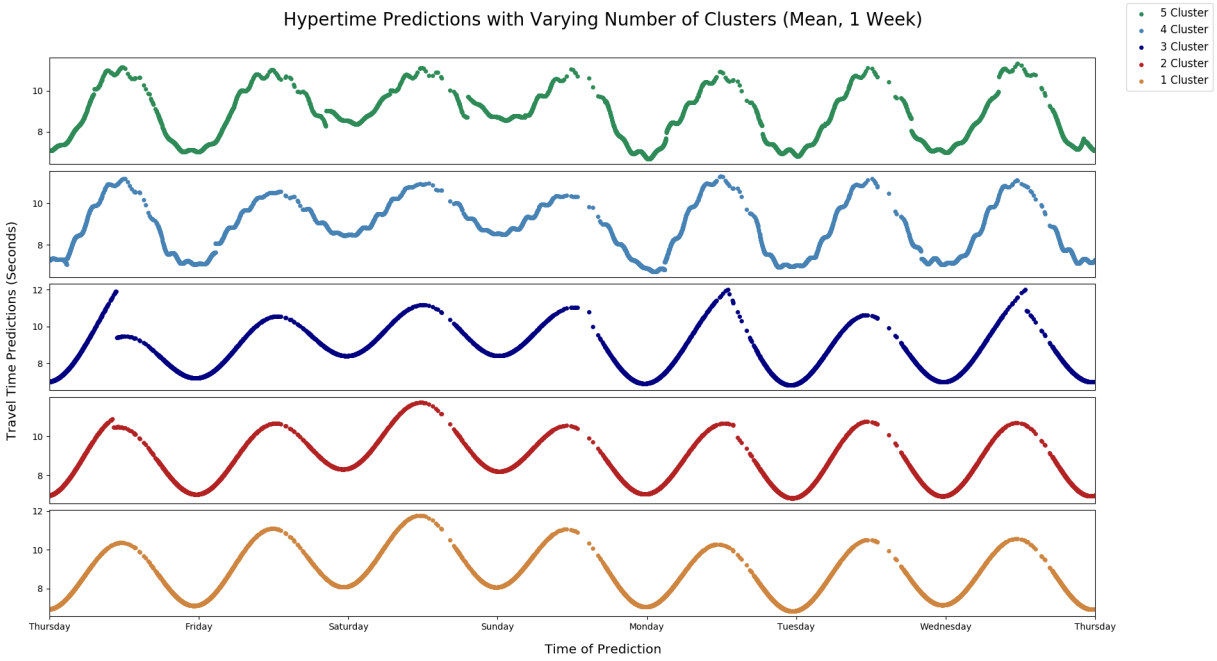

Figure 6.4: Comparison of Hypertime Predictions with Varying Number of Clusters

While the choice between the day and week variants was clear, the motivation behind choosing the mean prediction method and 5-cluster variant requires a more in-depth analysis.

While the sample mean variant was a contender for the chosen model configuration, it was not selected due to its inability to provide information on which cluster was used for prediction. This information is needed to create the White Box models. Unlike the mean and mode, the sample mean relies on combining estimates that can cross multiple clusters, thus making it an inappropriate fit for use with the White Box model. While this is unfortunate, as the results of this experiment show, the predictive power of the sample mean versus the mean obtained from the conditional probability are equivalent. For a similar reason, the 5-cluster variant was selected over the 2-cluster variant. The 5-cluster variant has more clusters, which provide ample opportunity for the White Box model to differentiate between predictions. Additionally, while the 5-cluster variant arguably overfits the data, producing minor, unnecessary fluctuations in predictions around roughly the same

time, these fluctuations should be negligible when bounded by confidence estimates. Thus, this specific combination of parameters will provide a favorable configuration for Hypertime to make accurate predictions while simultaneously providing optimal conditions for confidence estimates to be made in the forthcoming experiments.

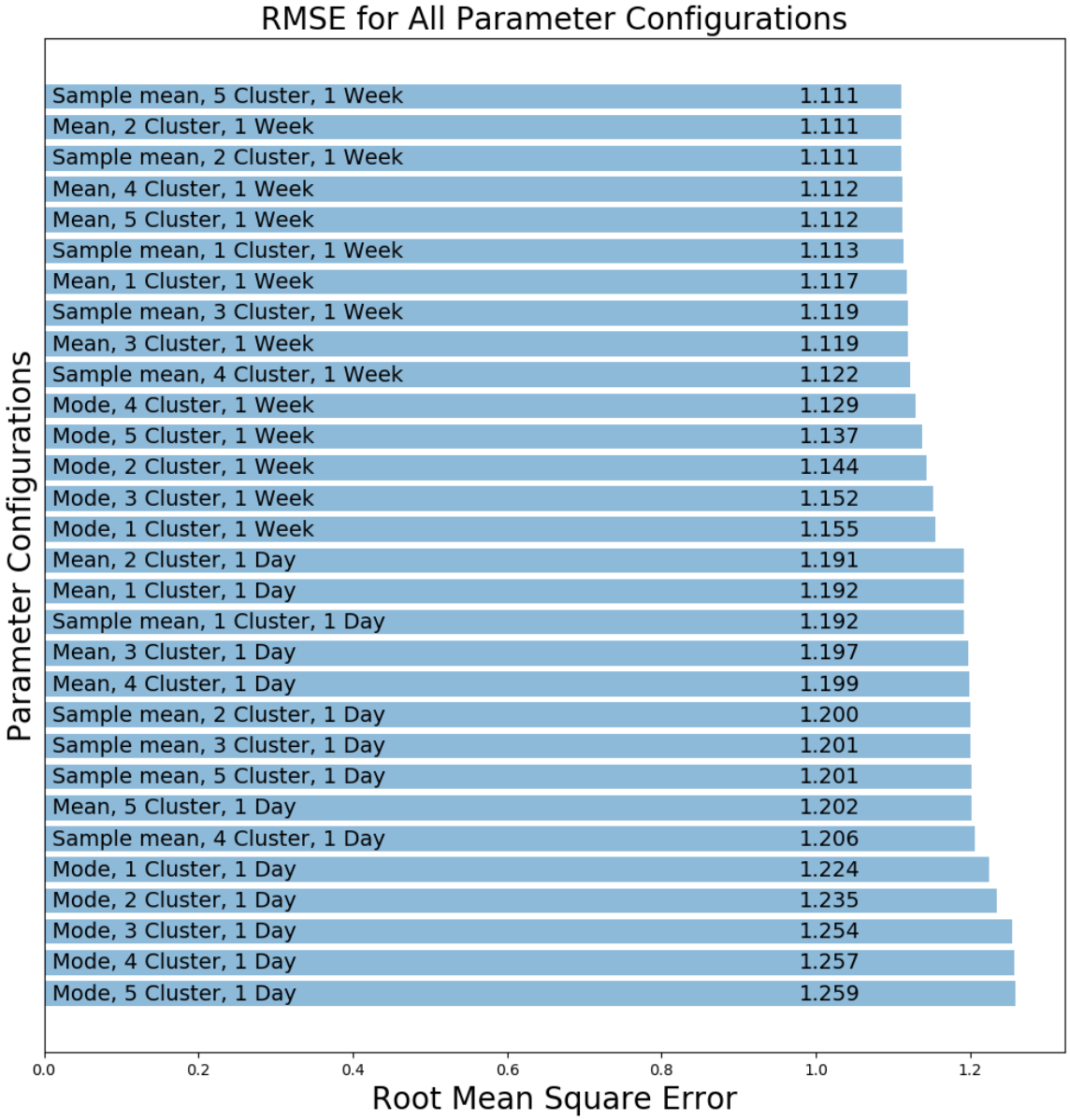

Figure 6.5: RMSE for All Tested Hypertime Parameter Configurations

6.2 Experiment 2: Confidence Estimation - Elevator Dataset

In a similar fashion to the results observed in the previous Hypertime Parameter Selection experiment, the results in this section can be used as a type of reference for later experiments. Given the high density and accuracy of the elevator dataset, the data used for training in this section represents the likely best case scenarios for these models, with a few minor exceptions. The dataset that is reviewed in the Complete Average Elevator Travel Time Dataset section contains complete and perfect knowledge of the elevator travel time, while the Sparse Average Elevator Travel Time Dataset section presents a likely scenario where knowledge about elevator travel times is limited to the individual experience of a fictional robot, or set of robots, with the elevator. In both cases, it is unlikely these models could be presented with better real-world data to train and predict on. This is in contrast with the results that will be analyzed in future sections.

6.2.1 Complete Average Elevator Travel Time Dataset

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	96.81%	0.839	0.350	5.077	0.00%
2σ Grey Box	96.14%	0.878	0.283	4.562	0.00%
2σ White Box	95.88%	0.885	0.265	4.426	0.00%
2σ White X Grey Box	95.65%	0.901	0.240	4.308	0.00%
CI Black Box	63.43%	0.337	0.699	1.601	0.00%
CI Grey Box	62.32%	0.382	0.627	1.691	0.00%
CI White Box	60.87%	0.368	0.634	1.608	0.00%
CI White X Grey Box	61.69%	0.403	0.582	1.701	0.00%

Table 6.1: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Elevator Dataset)

An overall summary of the results for this section can be found in Table 6.1. Of specific note for this particular dataset is the lack of any "Invalid" predictions. That

is, the models were never asked to make predictions about a group of data that they had never seen before in the training set. This further reinforces the completeness of this dataset. While the dataset may provide optimal training conditions, it is important to keep in mind that it has been sourced from real-world data and thus contains outliers and other noise that makes perfect predictions nearly impossible. Despite these challenges, the models tested performed well and displayed clear strengths, weaknesses, and trade-offs that will be analyzed further in the following two subsections, which have been grouped by the two main variables under test, the techniques used for creating the confidence estimates or bounds, and the methods for grouping data.

Bounding Techniques Comparison

Taking a look at the results in Table 6.1 a number of clear trends can be observed. First and foremost is the stark contrast in accuracy, an over 30% difference, between the more accurate 2σ and the more conservative confidence interval technique. However, this is somewhat expected given the difference in the philosophy behind these two methods. While both techniques assume that the error follows a Gaussian distribution, they differ in how bounds are determined. The confidence interval technique makes the assumption that there is a singular best value that represents the underlying error in the model for any given group/time, the mean in the equation. Furthermore, since the standard deviation's effect is limited by the square root of the number of observations, as the number of observations increases the predicted distance from the mean error observed decreases. This results in smaller bounds, but also results in fewer observed values being inside the predicted bounds. In contrast to the confidence interval, by bounding the mean error with two times the standard deviation, and not limiting its effect by the number of observations made, the bounds end up being much larger but encapsulate more observations as a result. In essence, the 2σ method assumes there is no singular "correct" error value and treats the underlying error of a group of data as a distribution.

This difference in assumptions continues to have implications when looking at

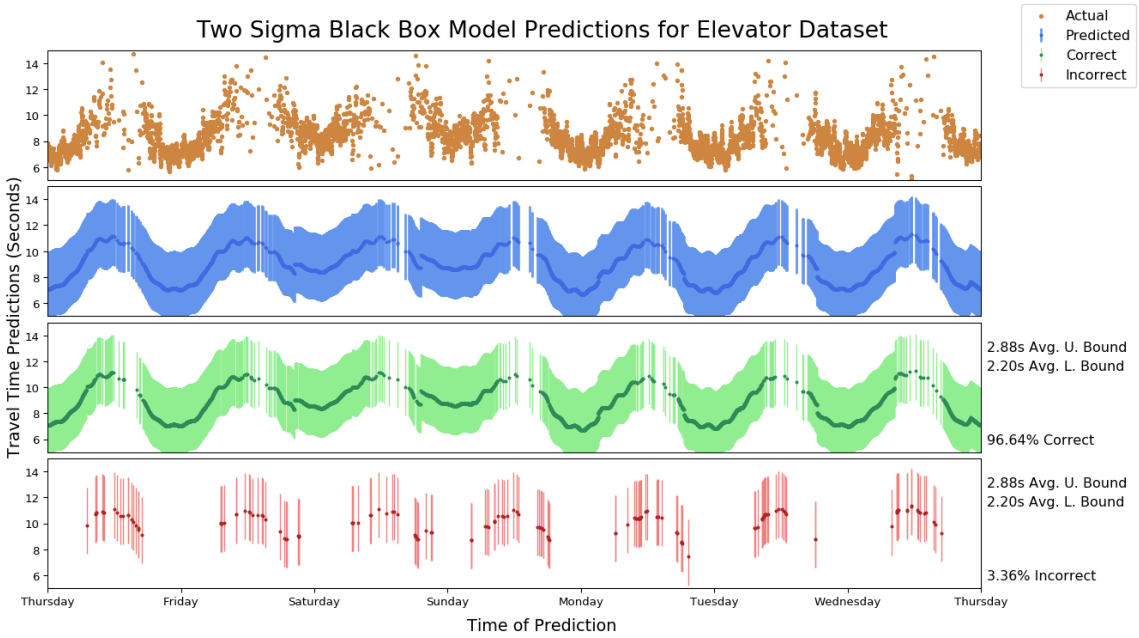

Figure 6.6: Two Sigma Black Box Model Predictions For Elevator Dataset

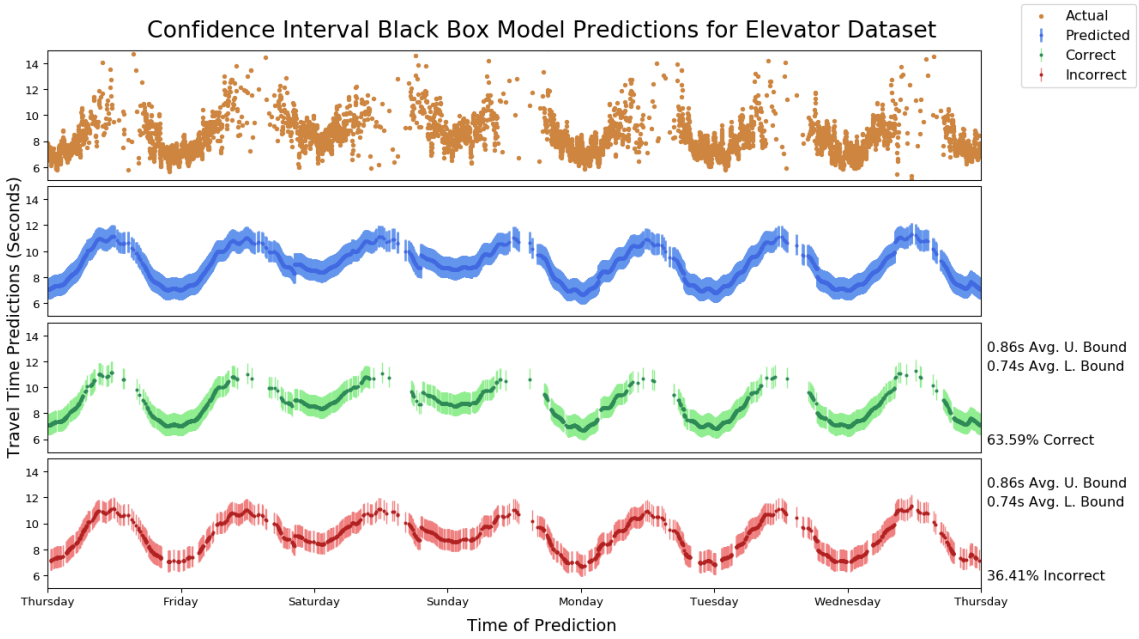

Figure 6.7: Confidence Interval Black Box Model Predictions For Elevator Dataset

other areas of the results. As mentioned above, the average size of the bounds is typically larger for the 2σ variants. This is clearly observable when comparing the most simplistic of the grouping methods, the Black Box, in Figures 6.6 and 6.7. While both methods struggle when there is large variance in the training data, such as during times of peak elevator use, the difference in the bounds is unmistakable. As a consequence of this, even during times of relatively low variance, the confidence interval variant struggles to consistently correctly bound the underlying behavior. Taking this analysis one final logical step further, the RMSE of the correct predictions and the magnitude of inaccuracy are inversely proportional. This is logically consistent since, for example, when the bounds are large, even outliers will not be much further outside the bounds.

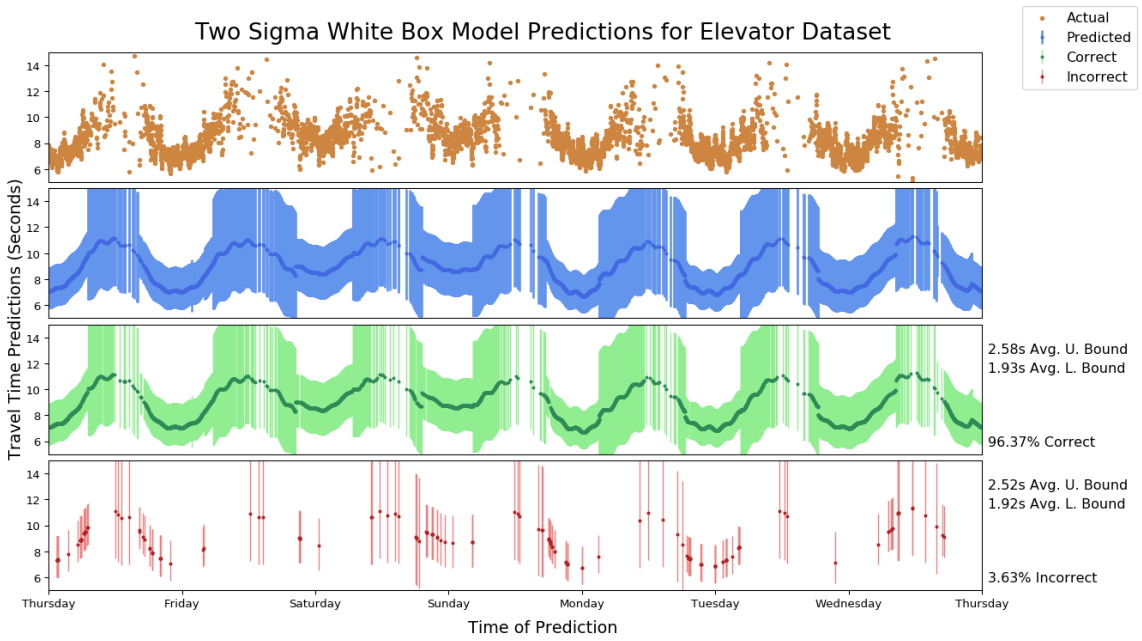

Figure 6.8: Two Sigma White Box Model Predictions For Elevator Dataset

In essence, if the desired behavior is to capture and be able to predict the likely range of time it would take to ride an elevator, or to do a given activity, then the 2σ method appears to be the preferred option. If, instead, it is desired to capture a majority of the range of a given behavior with less importance placed on capturing all

possible outcomes and more on summarizing the overall average while simultaneously minimizing the size of the bounds, the confidence interval appears to be a better option. Additionally, this selection may be influenced by the underlying error that is in question. If the error in the spatio-temporal model is likely to be a singular value that can be estimated more accurately with multiple observations the confidence interval would be preferred. The 2σ variant would be better suited for when the underlying error is expected not to be a singular value, but a range of values that may be variable and can be modeled by a normal distribution.

Data Grouping Method Comparison

While not as drastic as the difference in performance between bounding techniques when looking at the results in Table 6.1 the different grouping methods still have a profound effect on the confidence estimates provided. The results from the table are best analyzed in tandem with a graphical representation of the predictions. Looking at Figure 6.6 and Figure 6.8, the Black Box and White Box models respectively, perfectly demonstrates the difference between these two approaches. The Black Box grouping method is restricted and unable to make any assumptions about the underlying spatio-temporal model. It is only able to group prediction error by whether a prediction was above or below the actual value; therefore, it ends up being relatively simplistic. The White Box model, grouping data based on the Hypertime cluster responsible for a given prediction, is able to more effectively cluster errors. In this case, it results in two sets of distinct bounds, a set during high traffic and another during lower traffic.

While, in terms of accuracy, all of these models performed similarly, there appears to be a correlation between the number of possible groupings a model can make and the average size of its bounds. Models that contain more grouping on average have smaller bounds. Similarly to the comparison between the bounding techniques, decreased bound sizes correspond to increased RMSE and a decreased magnitude of inaccuracy. This type of distinction, the increased fidelity of bounds, and the decrease in average bound size could prove to be valuable for improved planning or similar post prediction processing, especially since it appears to maintain comparable

accuracy across the board.

6.2.2 Sparse Average Elevator Travel Time Dataset

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	93.88%	1.131	0.511	6.347	0.00%
2σ Grey Box	64.38%	0.808	0.913	2.988	17.36%
2σ White Box	89.51%	1.165	0.404	5.346	0.00%
2σ White X Grey Box	41.70%	0.603	1.201	1.531	52.99%
CI Black Box	63.05%	0.511	0.974	2.405	0.00%
CI Grey Box	52.70%	0.627	0.997	2.212	17.36%
CI White Box	65.35%	0.674	0.756	2.757	0.00%
CI White X Grey Box	37.99%	0.555	1.212	1.353	52.99%

Table 6.2: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Sparse Elevator Dataset)

The increased sparsity of this variant of the dataset was designed to approximate the amount of data a single robot may be able to obtain during normal operation and without specifically setting out to explore and collect spatio-temporal data. The contrast in performance between this version of the elevator travel time dataset and the one analyzed in the previous section specifically highlights the need for a sufficient quantity of training data. The ramifications of insufficient training data are explored in this section as well as in the Hallway experiment.

Perhaps the most obvious side effect of having less training data is presence of "Invalid" predictions. Whereas the complete experiment had no invalid predictions, some variations of this experiment had invalid or incomplete predictions on over half of the test dataset as visible in Table 6.2. As a reminder, this metric represents the percent of times where a confidence estimate prediction was requested from a model and the model was unable to provide both an upper and lower limit on the confidence estimate at that specific time. An example of this behavior can be seen in Figure 6.9, especially near Saturday where only upper bounds were provided. This

undesired behavior is a direct result of insufficient training data with respect to the type of data grouping method used. Data grouping methods, like the Grey Box and White X Grey Box, group data together strictly on temporal bounds. Since only a few data training points were available for training on Saturday morning, it is feasible that there was no training data where the trained model's prediction underestimated the actual behavior. As a result of this, the trained model never observed an underestimation and thus could not provide a lower bound during this time.

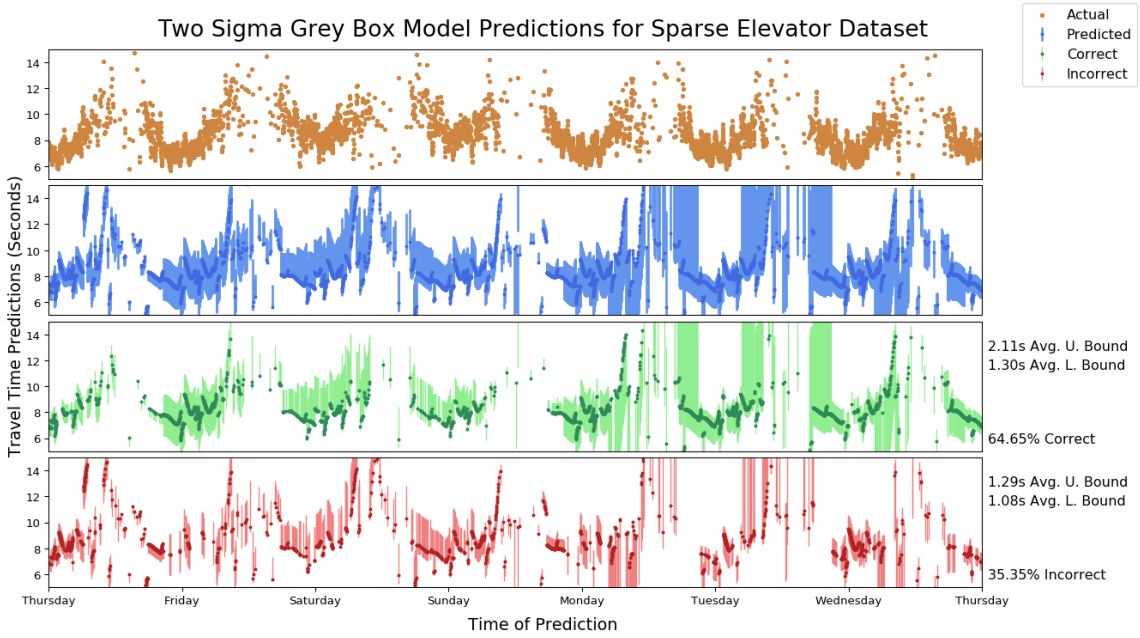

Figure 6.9: Two Sigma Grey Box Model Predictions For Sparse Elevator Dataset

While this behavior is undesired, it can be viewed as a symptom of an underlying issue. By analyzing when invalid predictions are made, it is possible to make recommendations on when and where to explore and collect more data. With this in mind, considering again the Grey Box model, it would beneficial to focus exploration on Saturdays and any other time with a significant amount of invalid predictions. Otherwise, if it is not possible or unreasonable to collect more data during certain times, it may be desirable to select a different method of data grouping. Since the Black Box model makes minimal assumptions about the data being analyzed and

the White Box model relies on the limited number of clusters Hypertime uses for prediction, they both are extremely unlikely to provide invalid prediction bounds, as verifiable in Table 6.2.

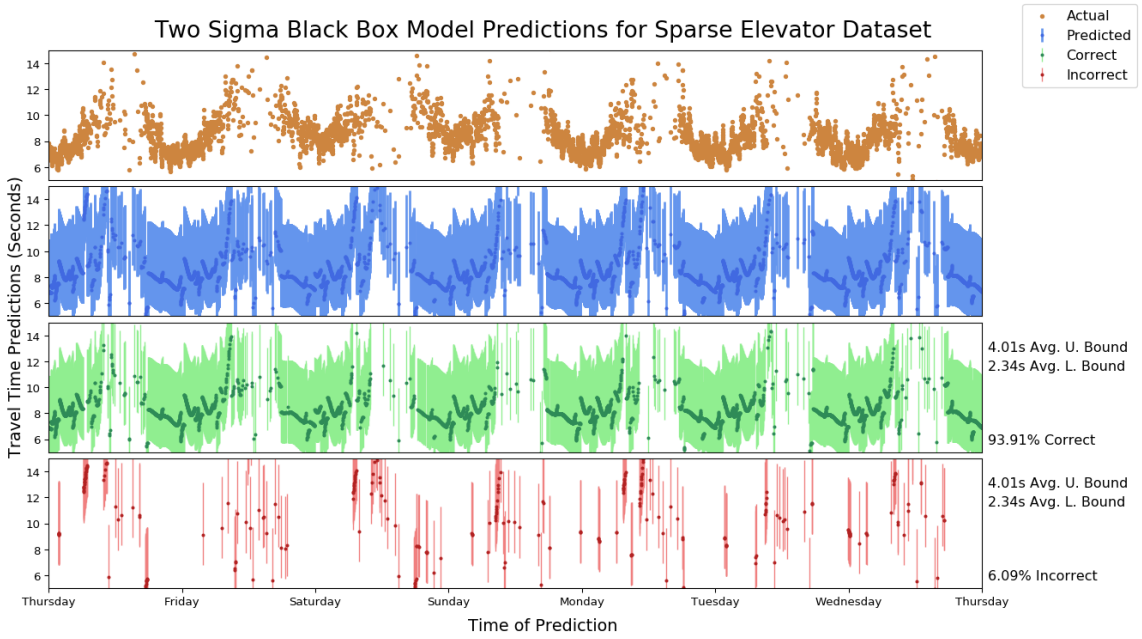

Figure 6.10: Two Sigma Black Box Model Predictions For Sparse Elevator Dataset

Finally, while the method of grouping data had a large effect on prediction quality and accuracy, the techniques for providing the bounds had limited effect when looking strictly at accuracy and accounting for changes due to grouping methods. Interestingly, the average bound across all methods increased proportionally with respect to the more complete training set. This is likely also due to the decrease in training data. Less training data means the method is less likely to get an accurate representation of the underlying data and outliers are more likely to skew the results. This increase in prediction bounds is easily seen when comparing the results in Figure 6.10 and Figure 6.6, specifically when comparing the average size of bounds on incorrect predictions.

6.3 Experiment 3: Confidence Estimation - H-BRS Hallway Dataset

While the previous elevator travel time experiments represented, in many ways, the best case scenarios for confidence predictions, the experiments with the Hallway Dataset in this section very much represent the opposite end of the spectrum. Perhaps the most obvious of these differences is the amount of data collected. Whereas the elevator travel time dataset had upwards of tens of thousands of data points for training, most of the hallways only have a couple hundred data points to be split between both testing and training. While this is still on the low end of data needed for a viable dataset, it is not unreasonable to expect an individual, or even group of robots, to collect more than a couple thousand runs[1] over a few months. For this reason, many of the results in this section share similarities with the results seen in the sparse Elevator Dataset, but this section's results additionally demonstrate when, where, and how these confidence estimating models begin to experience issues in a variety of different ways.

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	81.46%	2.089	4.753	25.996	0.00%
2σ Grey Box	30.34%	1.345	11.925	7.362	61.24%
2σ White Box	80.34%	1.594	11.898	12.909	12.36%
2σ White X Grey Box	12.36%	1.249	12.150	3.489	78.09%
CI Black Box	75.28%	1.081	10.100	8.183	0.00%
CI Grey Box	25.84%	0.704	11.963	3.684	61.24%
CI White Box	65.73%	0.933	12.080	4.449	12.36%
CI White X Grey Box	7.87%	0.496	12.177	1.747	78.09%

Table 6.3: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Hallway Dataset B)

As previously mentioned, the most common issue faced by these models during this experiment was a lack of training data. This is most directly visible when looking at the percentage of predictions that were made with partial or completely invalid

[1] A run, in the specific case of the Hallway Dataset, would consists of traversing all four hallways once

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	98.88%	4.383	20.089	25.284	0.00%
2σ Grey Box	23.60%	3.084	21.351	5.497	61.24%
2σ White Box	82.02%	4.363	20.030	14.423	0.00%
2σ White X Grey Box	15.73%	3.029	21.361	5.122	72.47%
CI Black Box	79.78%	1.597	20.998	7.605	0.00%
CI Grey Box	8.99%	0.589	21.387	2.410	61.24%
CI White Box	41.01%	0.735	20.844	5.383	0.00%
CI White X Grey Box	1.69%	0.119	21.376	2.504	72.47%

Table 6.4: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Hallway Dataset C)

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	71.35%	5.692	261.716	43.029	0.00%
2σ Grey Box	26.97%	2.572	265.412	5.803	61.24%
2σ White Box	66.29%	3.148	263.987	21.011	20.79%
2σ White X Grey Box	15.73%	2.569	265.496	3.258	83.71%
CI Black Box	49.44%	1.870	264.588	10.308	0.00%
CI Grey Box	23.03%	1.131	265.424	3.309	61.24%
CI White Box	46.63%	1.697	265.036	7.232	20.79%
CI White X Grey Box	13.48%	2.362	265.497	1.875	83.71%

Table 6.5: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Hallway Dataset D)

bounds. With respect to this metric, hallways B, C, and D, all perform similarly as seen in Tables 6.3, 6.4, and 6.5, respectively. Since all three of these hallways were part of the same preprogrammed route, they all exhibit similarly high numbers of invalid predictions due to lack of data collection. While the percentage of invalid predictions is higher than might be expected, this can be partially explained by the difference in time and days when data was collected early in the dataset versus later. This is particularly an issue for the Grey Box model that groups data strictly temporally as seen during Wednesdays and Thursdays in Figure 6.11[2]. Since little to no training data was captured during these times, it was not possible to put bounds on many of the data points. While this is certainly an issue if left unmanaged, there are ways to mitigate this issue. If certain times are know to be of interest ahead of time, it is possible to query Hypertime over the future range of time of interest. Any data points with invalid, or even extremely large bounds, could be explored further to reduce the issues.

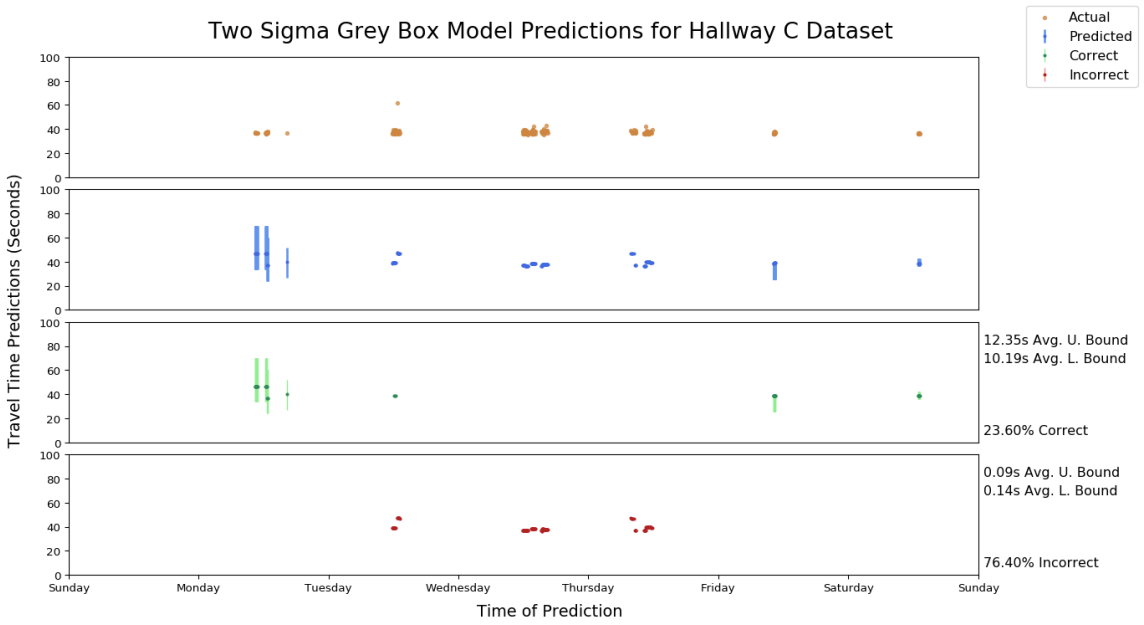

Figure 6.11: Two Sigma Grey Box Model Predictions For Hallway Dataset C

[2]To improve visualization, and due to the sparse nature of the data for this dataset, all graphical figures in this section display the entirety of their respective test dataset superimposed over a timespan of one week.

Additionally, while a significant number of invalid predictions has been observed previously in the Grey Box and associated Grey X White Box models, in this experiment the same behavior was also observed in the White Box models. Since the White Box models rely on the cluster responsible for a given prediction, this would mean that a given cluster that was never used to make a prediction in training was responsible for a prediction during test. While this was thought to be highly unlikely, it does appear to have been an issue. A deeper investigation of this issue reveals that in the case of datasets Hallway B and Hallway D, the unexpected cluster is centered around zero. Since the training dataset did not have any zero or single-digit travel times, it seems likely that this is an issue or side effect of Hypertime itself. While the specific conditions required to recreate this issue in Hypertime are not yet fully established, it is believed that a zero cluster is the result of a confluence of events. The number of clusters Hypertime uses for grouping spatio-temporal data is user-provided at training time and is thus immutable thereafter. Attempting to group data into five clusters when the dataset is already of limited quantity and spatio-temporal variance likely resulted in two to four potentially similar clusters and a fifth being placed at zero. Consequently, during the test, a specific time corresponded best to this zero cluster and resulted in prediction and confidence estimate issues. The zero cluster and corresponding predictions can be seen in Figure 6.12[3] on Tuesday and Thursday.

While this is certainly an issue and presents a unique challenge for models that used the White Box model, there are potential ways to minimize this issue for both the Hallway B and Hallway D datasets. The simplest way of mitigating would be to use fewer clusters when training Hypertime models. However, the immediate drawback to this countermeasure is the decreased number of groups White Box dependent models would have available for providing unique confidence estimates. Another option would be further investigation of the root cause of the issue to confirm the potential problem outlined in this paper. Armed with a deeper understanding of the core issue, active preventative measures could be taken by collecting additional

[3]Note that Figure 6.12 contains the predictions and confidence estimates of a Black Box model, and while it is affected by the zero clustering issue, the issue does not affect its ability to make confidence estimates. It is for this reason that the model was selected as a reference, as the zero-clustered predictions are clearly visualized and highlighted by their bounds.

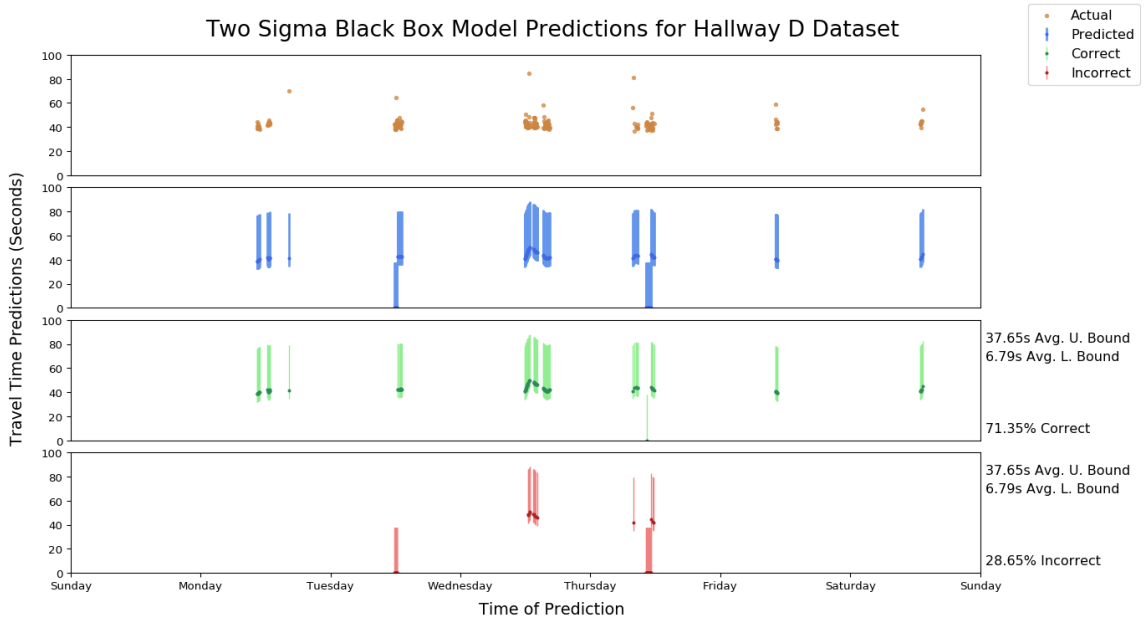

Figure 6.12: Two Sigma Black Box Model Predictions For Hallway Dataset D

data when possible to avoid this issue entirely.

While the results for Hallways B, C, and D all had an elevated percentage of invalid predictions, the results for Hallway A did not. In fact, Hallway A is the only hallway that did not contain any predictions with invalid bounds for the White Box related models. This result can be attributed to the fact that Hallway A was used not only when running the pre-programmed route that includes the other hallways in the dataset, but also for additional test runs for the ROPOD project. While these additional runs provided more data for training, and thus resulted in less invalid predictions, they also caused issues. Namely, since room C022 marks the start of Hallway A and C022 is the lab where the majority of new software development for ROPOD occurs, a large number and variety of tests are included in the Hallway A dataset. This includes runs where the robot had issues, long periods of waiting while developers made changes to the robot, and similar issues. Since a significant number of travel times recorded for Hallway A were abnormally long, sometimes upwards of tens of minutes, there were adverse affects on the predictive effectiveness of both Hypertime and the confidence estimate models under test.

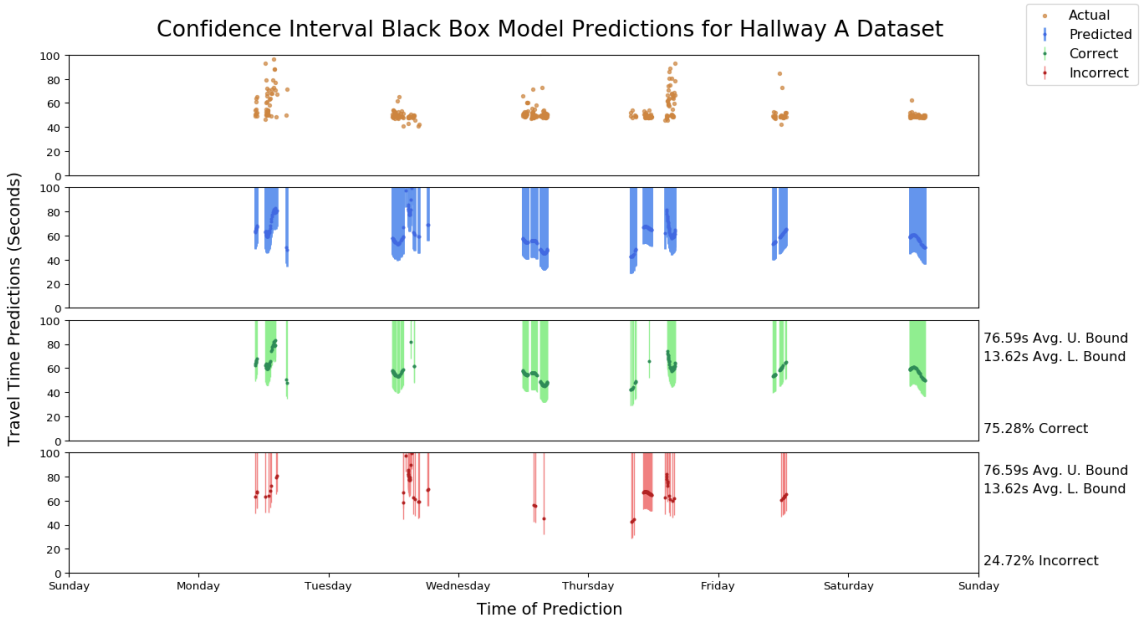

Figure 6.13: Confidence Interval White Box Model Predictions For Hallway Dataset A

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Black Box	95.79%	37.543	107.300	358.562	0.00%
2σ Grey Box	60.39%	24.307	123.142	170.956	34.55%
2σ White Box	91.29%	29.028	107.043	338.523	0.00%
2σ White X Grey Box	47.75%	22.175	124.168	165.829	47.47%
CI Black Box	75.28%	11.674	137.409	90.211	0.00%
CI Grey Box	43.82%	9.778	138.085	66.064	34.55%
CI White Box	63.48%	10.238	134.546	93.320	0.00%
CI White X Grey Box	33.15%	10.663	136.725	69.403	47.47%

Table 6.6: Test Data Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Predictions For Hallway Dataset A)

The effects of this noisy data can be seen in abnormally large upper bounds in Figure 6.13 and in the average bound size and magnitude of inaccuracy metrics in Table 6.6. While abnormally large upper bounds do increase increase the frequency at which observed events are within the bounds, this comes at the cost of unreasonably

large bounds which almost entirely negate their purpose. In order to avoid this issue, the training data must either be less noisy, or methods could be employed after data collection to remove noise. Something like a statistical model to remove outliers may suffice.

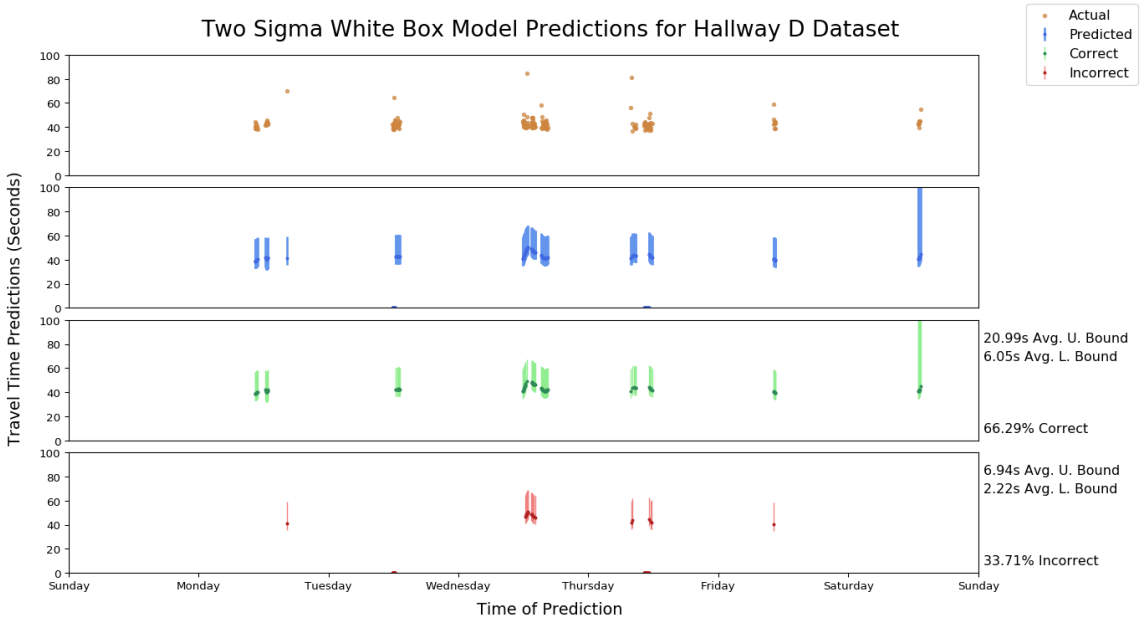

Figure 6.14: Two Sigma White Box Model Predictions For Hallway Dataset D

Indeed, while noisy training data can cause issues in confidence estimates, a small amount of noise is acceptable. At the end of Hallway D there is an automatic door that occasionally causes traversal issues for the robot used for data collection. It can occasionally become stuck until a human triggers, or otherwise opens, the door. However, in contrast to Hallway A, while Hallway D does have a non-zero amount of noise it manages to perform acceptably well under the given circumstances. Evidence of the slightly noisy training data can be seen in the large magnitude of inaccuracy as seen in Table 6.5. This large magnitude of inaccuracy is similar to that seen for Hallway A in Table 6.6. Despite this noisy data, the actual size of the bounds in Figure 6.14 and Table 6.5 is much more reasonable. This affirms the assertion that any amount of noise may have an impact, but small amounts of noise have negligible negative impact on the predictive quality of confidence estimates.

6.4 Experiment 4: Multi-Model Fusion Proof of Concept

Having established the relative strengths and weaknesses of the proposed models, this experiment serves as a demonstration of a number of potential methods for selecting between a multiple number of models. As mentioned previously, the benefit of this is twofold. First, as observed in the previous Hallway Dataset experiment, models that group data into smaller, more numerous groups often have difficulty providing bounds for all predictions when trained with on sparse datasets. By using multiple models, the more specific groupings and predictions can be used when possible, and more generic groupings and predictions, like those of the Black Box model can be used at other times. Second, while this experiment only uses variants of Hypertime, it is conceivable that multiple spatio-temporal models could be maintained simultaneously with their predictions being selected from or combined to produce a superior model. While, outside of Hypertime, the latter remains future work, the methods covered in this section prove the multi-model fusion concept while providing valid bounds for all requested times.

Model	**% Corr.**	**RMSE Corr.**	**Magnitue of Inac.**	**Avg. Bound**	**% Invalid**
2σ Best Accuracy Predictions	93.93%	1.202	0.471	6.584	0.00%
2σ Lowest Bound Predictions	65.43%	0.780	0.864	2.776	0.00%
2σ Best Hybrid Predictions	68.29%	0.792	0.793	2.939	0.00%
2σ Black Box	93.88%	1.131	0.511	6.347	0.00%
2σ Grey Box	64.38%	0.808	0.913	2.988	17.36%
2σ White Box	89.51%	1.165	0.404	5.346	0.00%
2σ White X Grey Box	41.70%	0.603	1.201	1.531	52.99%
CI Best Accuracy Predictions	68.20%	0.744	0.840	2.953	0.00%
CI Lowest Bound Predictions	49.07%	0.401	1.050	1.700	0.00%
CI Best Hybrid Predictions	51.33%	0.421	1.034	1.776	0.00%
CI Black Box	63.05%	0.511	0.974	2.405	0.00%
CI Grey Box	52.70%	0.627	0.997	2.212	17.36%
CI White Box	65.35%	0.674	0.756	2.757	0.00%
CI White X Grey Box	37.99%	0.555	1.212	1.353	52.99%

Table 6.7: Original and Multi-Model Fusion Confidence Estimate Results for Mean, 5 Cluster, 1 Week (Sparse Elevator Dataset)

In accordance with its greedy approach, the Best Accuracy models outperform all

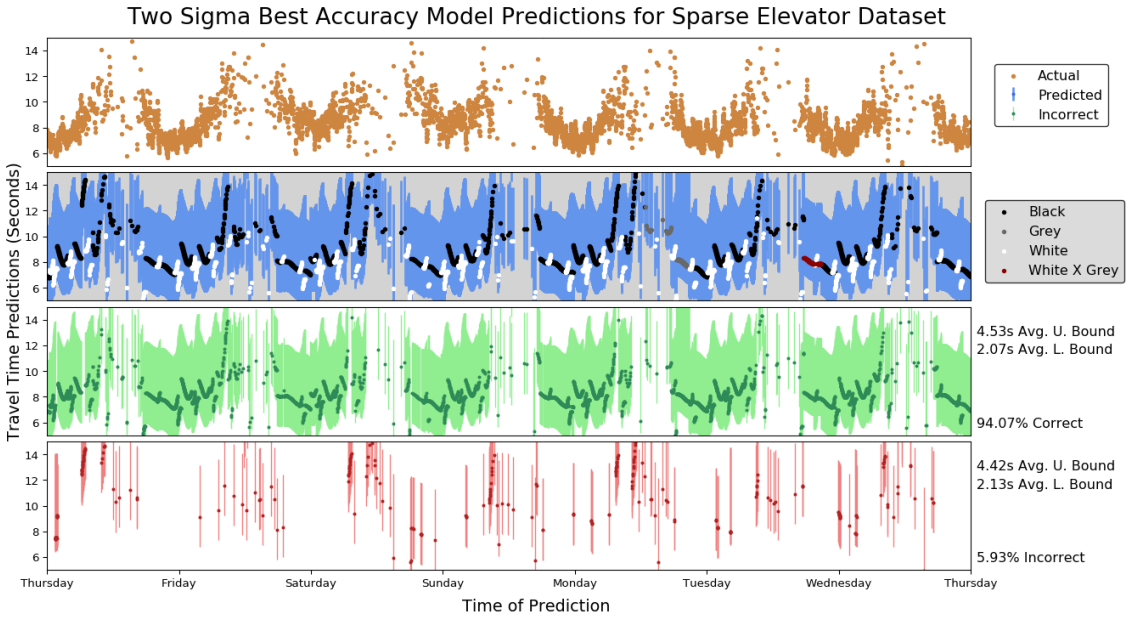

Figure 6.15: Two Sigma Best Accuracy Model Predictions for Sparse Elevator Dataset

other models with respect to their given confidence estimating technique, as seen in Table 6.7. While the performance increase is present, more so for the confidence interval variant, it is likely that the models are approaching diminishing returns. In the ideal case, both the confidence interval and two sigma approach should be achieving around 95% accuracy, of which the Best Accuracy model is within just over a percent.

Looking closer at Table 6.7 and comparing the Best Accuracy model with the performance of others, it appears most similar to the Black Box model. However, while a large portion of the predictions for the Best Accuracy model come from the Black Box model, looking at Figure 6.15, predictions from all four models can be seen. The Black Box model appears makes up the base predictions, especially during times of high traffic, while a cluster or two from the White Box model make up a large portion of predictions during non-peak hours. This provides an excellent example of the type of fusion that is desirable. It is likely that Hypertime has clustered a number of non-peak hour data that has a relatively low variation and thus makes accurate bounding more achievable during that time. An example of this can be seen in Figure 6.16, especially when comparing a relatively high traffic day such as

Saturday to a lower traffic day like Tuesday.

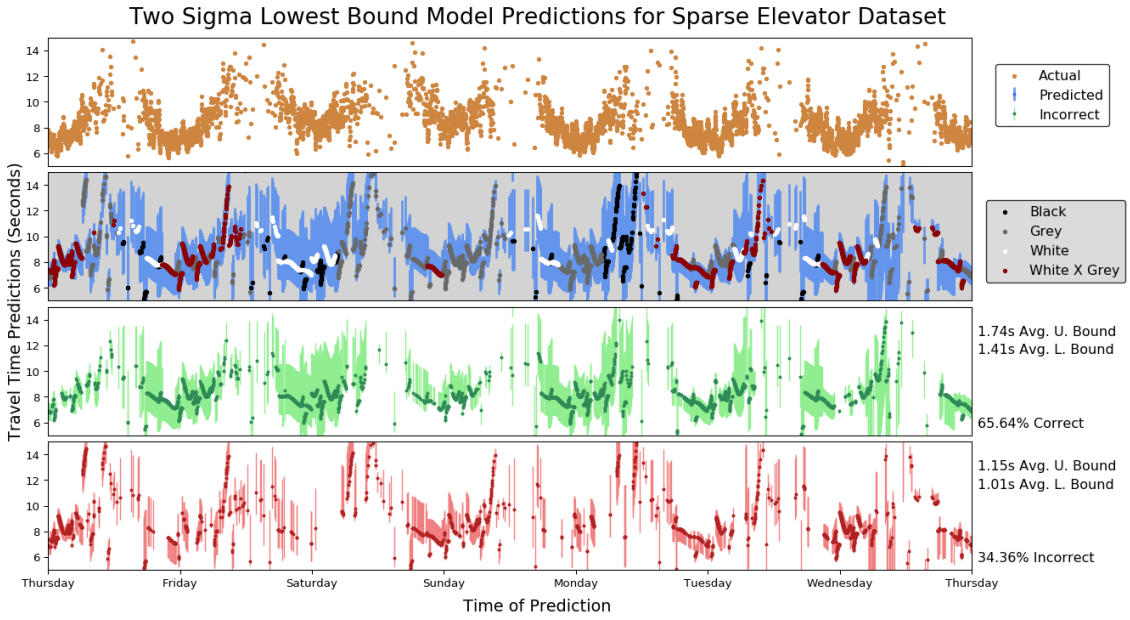

Figure 6.16: Two Sigma Lowest Bound Model Predictions for Sparse Elevator Dataset

The other greedy approach, the Lowest Bound models, act equally in accordance with their name. Only the original White X Grey Box models have lower bounds than this approach. This is likely only the case due to their high number of invalid predictions leaving only a few valid bounds likely of a small magnitude. While a smaller bound size may be desirable, unfortunately this naive approach performs relatively poorly in terms of accuracy, only beating models which had a number of invalid predictions. While the Lowest Bound greedy approach may be undesirable in terms of accuracy, the approach was still able to provide accurate predictions for roughly two thirds of all times requested for the two sigma variant. Additionally, only selecting for the smallest bounds over multiple models limits the effect perceived noise can have on bound sizes. While this has an obvious negative impact on accuracy, it may be desirable for some applications that do not require precise predictions but a rough visual representation of times that truly have high variance.

The final and only non-greedy approach of the section, the Best Hybrid models,

provided prediction bounds of size and accuracy that lie between the other two methods. While this isn't unexpected, as the hybrid model's metric is lowest bound size modified by accuracy, its performance is much closer to that of the Lowest Bound models. On average, predictions provided by this approach will have bounds of slightly larger size, with limited increase in accuracy when compared to the Lowest Bound model. This is likely due to this metric scaling linearly with respect to accuracy. While a non-greedy and/or hybrid approach likely has a higher performance ceiling, these results show that careful tuning must be done to devise an optimal metric. This will be especially important if multiple disparate spatio-temporal models are to be fused, as is expected for future work.

7

Conclusions

The work presented in this thesis focused on developing and applying frameworks to design and evaluate techniques for providing confidence estimates to existing spatio-temporal models. Many motivations, design decisions, and findings of this work, while applicable to the field as a whole, were developed with the ROPOD project in mind. For this reason, special care was directed towards these models' use in providing bounds for the time it takes to travel various sections of human shared environments. A framework for approaching the development of new confidence estimation techniques was presented that categorized techniques based on the amount of information known about the internal workings of a spatio-temporal model. Having selected Hypertime as the chosen spatio-temporal modeling technique, four confidence estimation methods were devised using the aforementioned framework. Confidence estimates were provided by grouping and analyzing the accuracy of predictions on various training datasets. Two statistical methods for analyzing the results and providing temporal bounds were presented.

Two new real-world datasets were presented, one related to the travel time of a hospital elevator and the other containing information on the travel time of multiple university hallways. Both datasets were recorded over the course of multiple months. The datasets were used to evaluate the new confidence estimation techniques. The results for the experiments showed a strong correlation between spatio-temporal data density and the ability to present accurate confidence estimates. A graphical

presentation of the results shows the ability for models to provide recommendations on when uncertainty is high, which can allow for improved targeted data collection. Models trained with datasets lacking sufficient data were found to be unable to frequently provide valid confidence estimates. A proof-of-concept for combining the predictions of multiple models to mitigate this issue was successfully demonstrated.

7.1 Recommendations for ROPOD

Using the results of this work as a base, it is possible to make recommendations for what type of confidence estimation technique ROPOD should use in combination with Hypertime. First and foremost, the importance of selecting the correct tuning parameters for Hypertime can not be overstated. It is recommended to use the tools created during the course of this work to benchmark any other datasets or environments for which Hypertime will be used as the modeling method. In this way, optimal predictions can be ensured. Once Hypertime is tuned, some amount of developer insight will be needed to select the desired methods for confidence estimates. While not always the best option, the two sigma bounding method combined with the White Box model, which uses the clusters provided by Hypertime, is an excellent baseline. Special care must be taken to ensure significant data is collected. From the results of the sparse datasets, a month of data with approximately one data point per hour (during times of interest) is the bare minimum needed. If sparse data is unavoidable, as may be the case for some environments, a fusion of multiple models may be desired. Combining the baseline and general predictions of the Black Box model with the more temporally specific predictions of the White Box model is recommended. Lastly, care should be taken to remove significant outliers or noise in training datasets to avoid erroneous prediction bounds.

While not a major focus of this work, some recommendations on the implementation-specific details for Hypertime with confidence estimates can be addressed. The spatio-temporal training data can be captured either directly from the robots via communication with a server or extracted from stored ROS bag files. This data should be stored in a database containing the location, time and observed behavior, which can all be of later use. As training a Hypertime model can take from as little as a minute to hours, depending on the amount of training data, the training of

the models should be done overnight or during other periods of low use. Trainings need not happen more than every other day but should happen at least every week. Docker can be used to enable flexibility with respect to targeted run environments. When a prediction is requested, a Docker container can be queried with a given Hypertime model, time, and location. Finally, while not of direct concern to the training or operation of the Hypertime models, it may be desirable to implement methods to visualize the models. This could take the form of heatmaps or other formats that would grant insight into the motivations and decisions made by the individual robots.

7.2 Contributions

Throughout this work a number a number of contributions to the field of spatio-temporal modeling were developed and presented, the most important of which are presented below.

- Presents a methodology for approaching the development of novel confidence estimation techniques for spatio-temporal models
- Demonstrates the use of this new methodology for developing spatio-temporal models with confidence estimates
- Provides multiple new real-world datasets with varying noise and data density
- Outlines several techniques for comparing and evaluating the confidence estimates provided by spatio-temporal models
- Introduces novel prediction method for Hypertime
- Evaluates and compares multiple confidence estimation methods used with Hypertime
- Recommends optimal confidence estimation technique for deployment with existing autonomous robotics project, ROPOD

7.3 Future work

In addition to the above contributions to the field, a number of possible avenues for future work can also be suggested. First, while the multi-model fusion

proof-of-concept presented in the final experiment provides a good outline on a possible method for combing multiple confidence estimate techniques, more work is required. Of specific interest is the development of new heuristics to aid in the selection of the "best" prediction. Current techniques, while successful, proved to be too naive in their assumptions resulting in slightly suboptimal method fusion. Additionally, further research is required to explore the possibility of combining multiple dissimilar spatio-temporal models, e.g. temporally long-term model and a temporally short-term model.

With respect to Hypertime, an odd behavior was observed during experiments where some predictions of zero time length were provided. It was hypothesised that these were likely due to an internal clustering issue as the result of a clustering parameter being too large, but further research is required to conclude this definitively. Additionally, while some of the best success was seen grouping Hypertime data using its internal cluster information, i.e. the White Box model, it is possible that other data clustering techniques may be viable and could result in superior confidence estimates. Likewise, while both the two sigma and confidence interval bounding methods provided analytically useful results, other methods may provided additional accuracy or statistical insight.

Finally, although mentioned but not directly addressed via the experiments in this work, additional uses of confidence estimates provided by spatio-temporal models are possible. Techniques for determining when prediction bounds are irregularly large are of specific note. With significantly advanced techniques it may be possible to differentiate between large bounds caused by an existing environmental behavior and an insufficient amount of spatio-temporal data. Furthermore, it may be of interest to provide a graphical interface for these types of observations. This would allow both developers and end-users alike to be more informed about a robot's perception and understanding of its environment.

A

Additional Agaplesion Hospital Elevator Dataset Results

A.1 Additional Results for Complete Travel Time Elevator Dataset

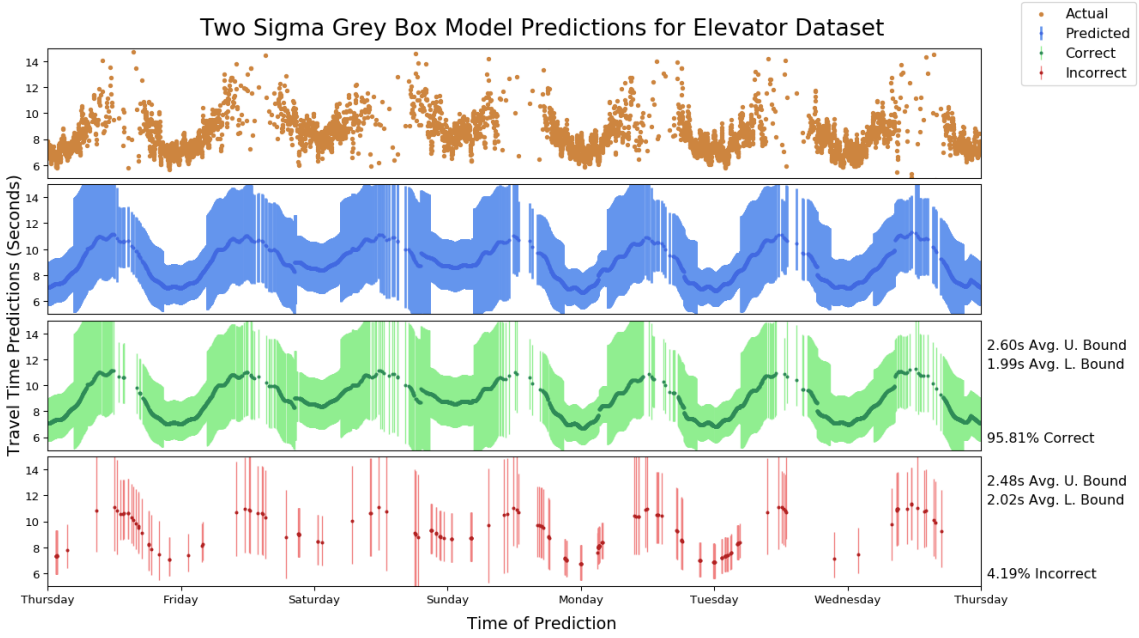

Figure A.1: Two Sigma Grey Box Model Predictions For Elevator Dataset

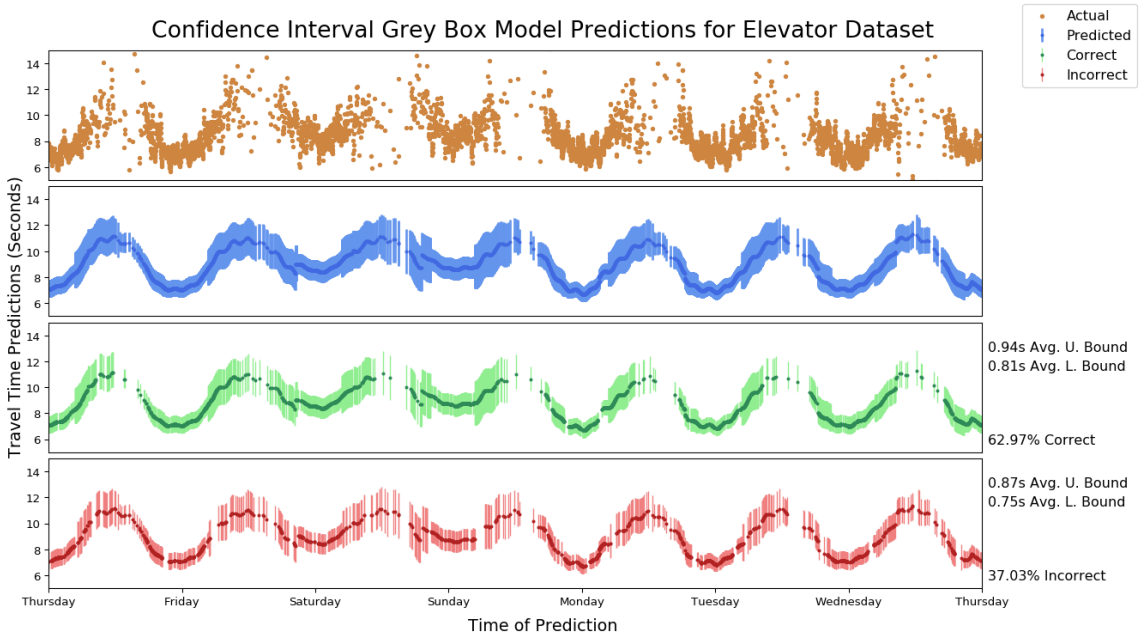

Figure A.2: Confidence Interval Grey Box Model Predictions For Elevator Dataset

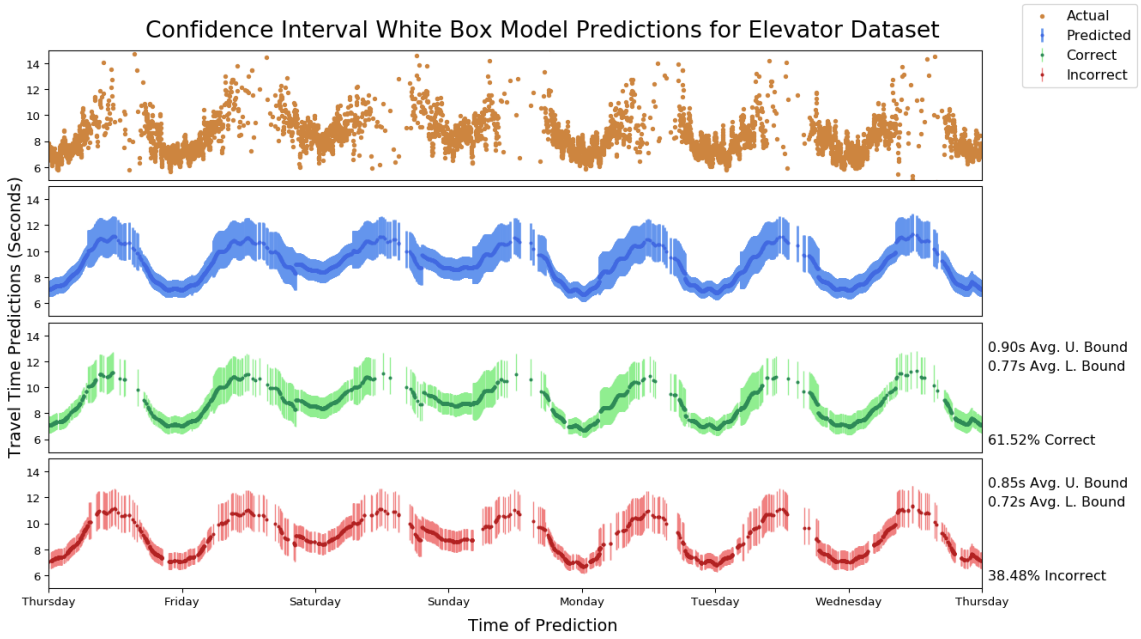

Figure A.3: Confidence Interval White Box Model Predictions For Elevator Dataset

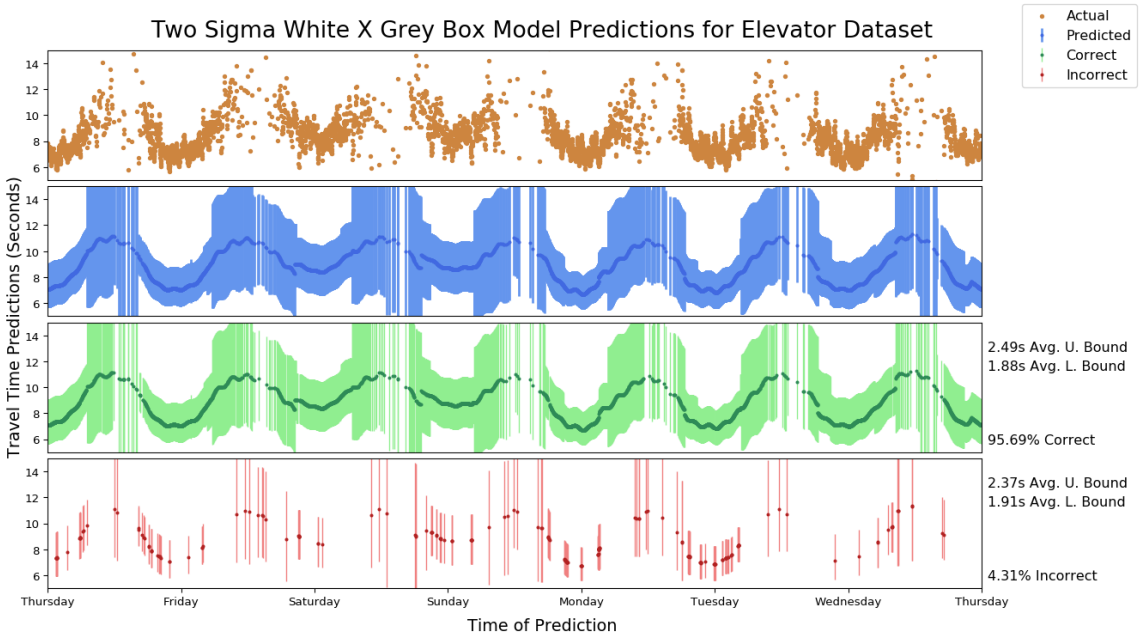

Figure A.4: Two Sigma White X Grey Box Model Predictions For Elevator Dataset

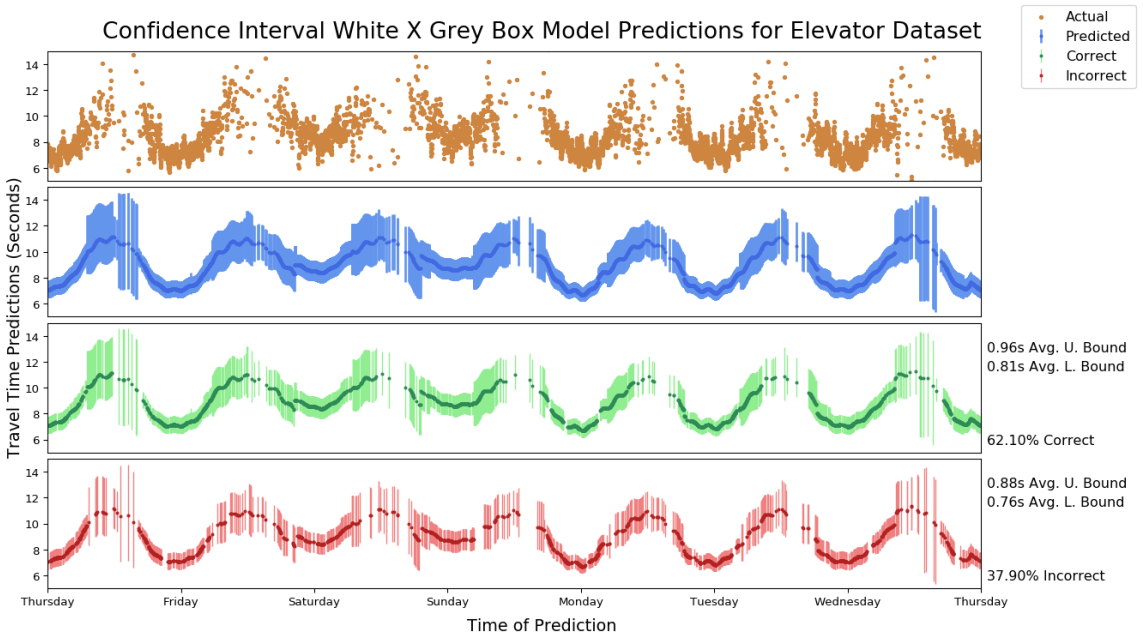

Figure A.5: Confidence Interval White X Grey Box Model Predictions For Elevator Dataset

A.2 Additional Results for Sparse Elevator Travel Time Dataset

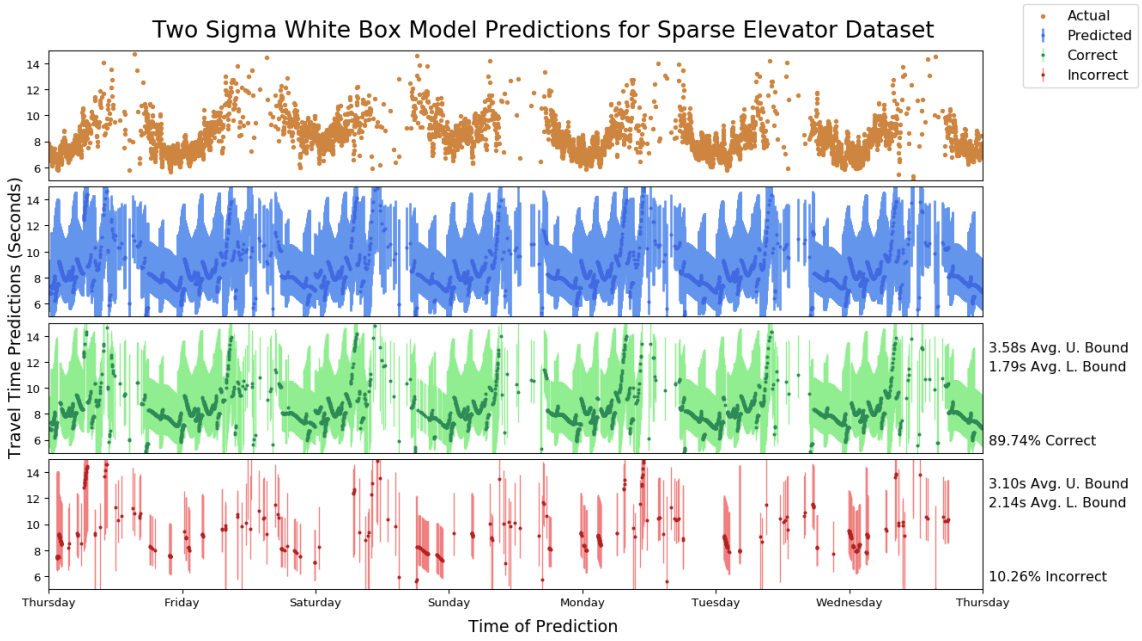

Figure A.6: Two Sigma White Box Model Predictions For Sparse Elevator Dataset

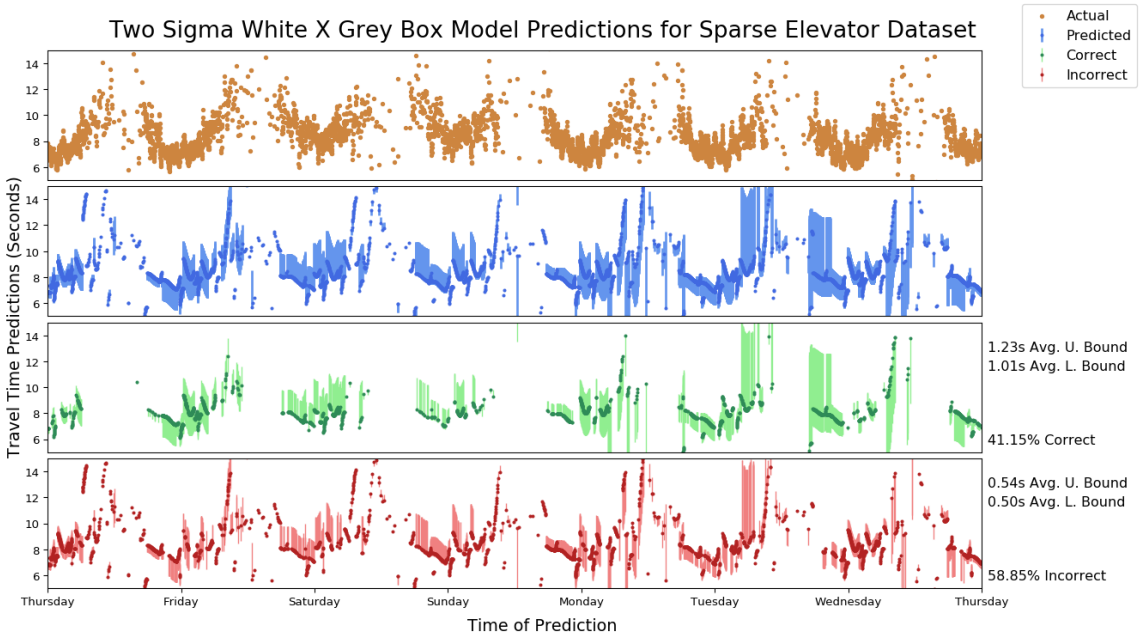

Figure A.7: Two Sigma White X Grey Box Model Predictions For Sparse Elevator Dataset

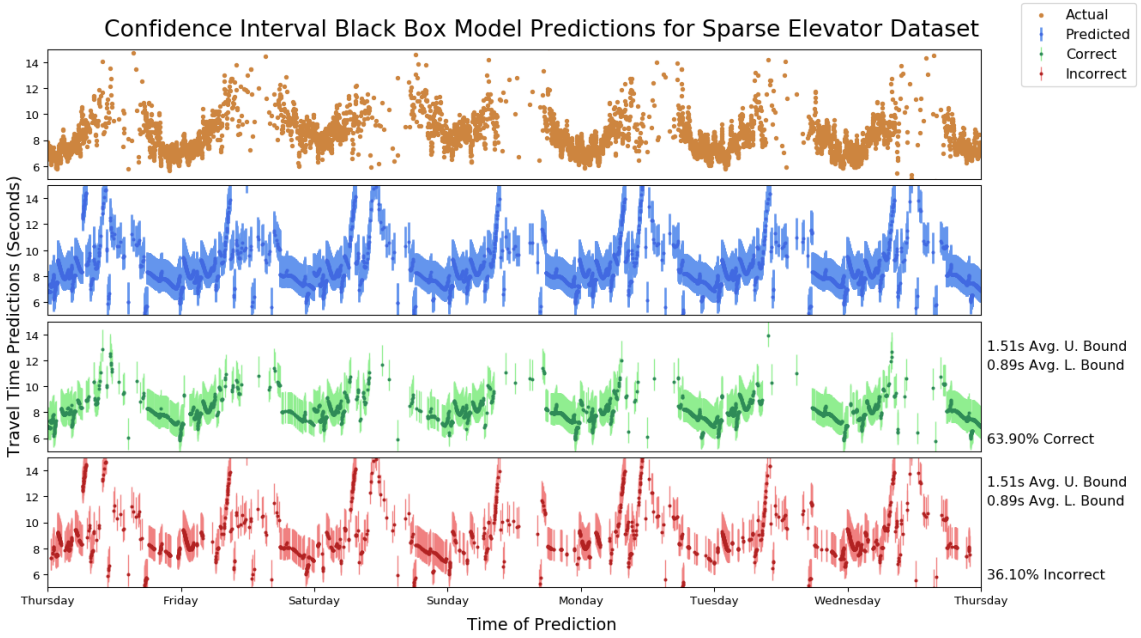

Figure A.8: Confidence Interval Black Box Model Predictions For Sparse Elevator Dataset

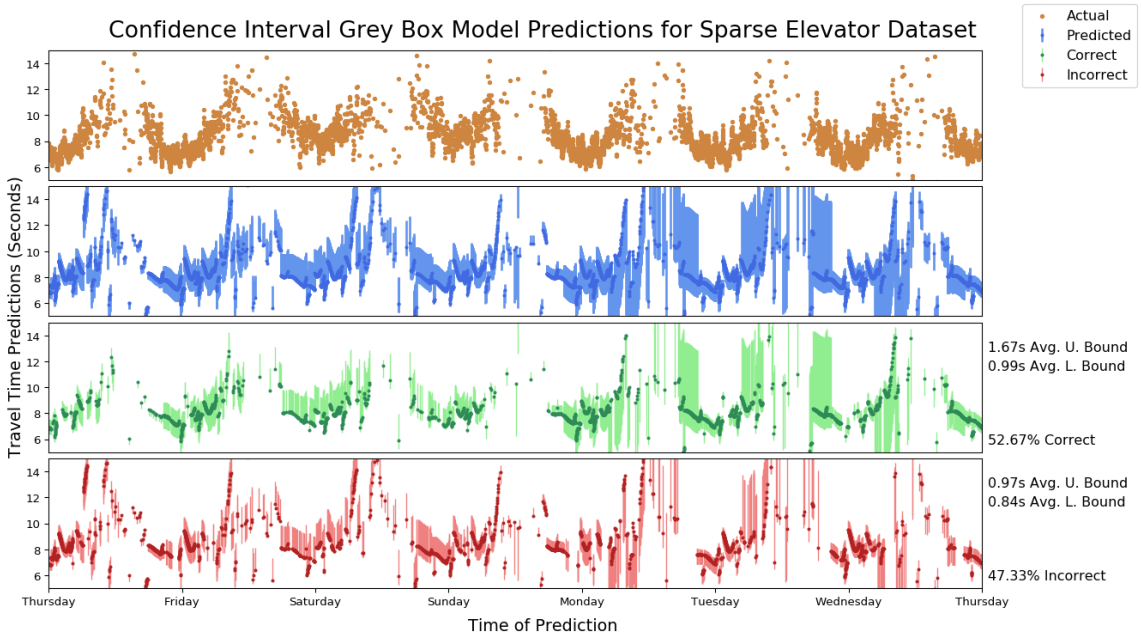

Figure A.9: Confidence Interval Grey Box Model Predictions For Sparse Elevator Dataset

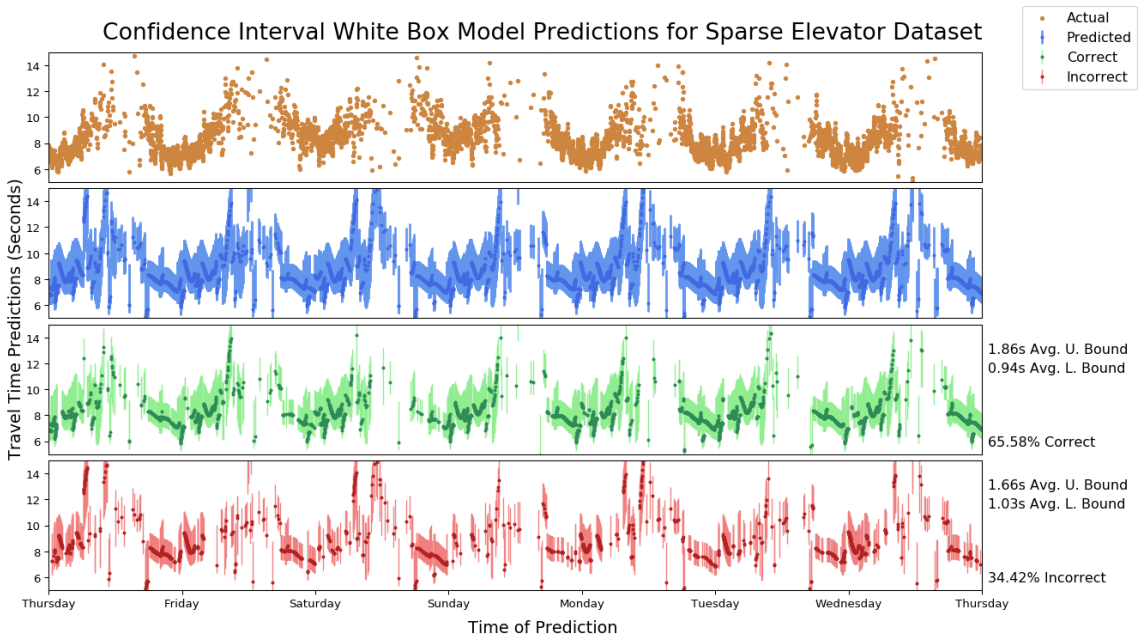

Figure A.10: Confidence Interval White Box Model Predictions For Sparse Elevator Dataset

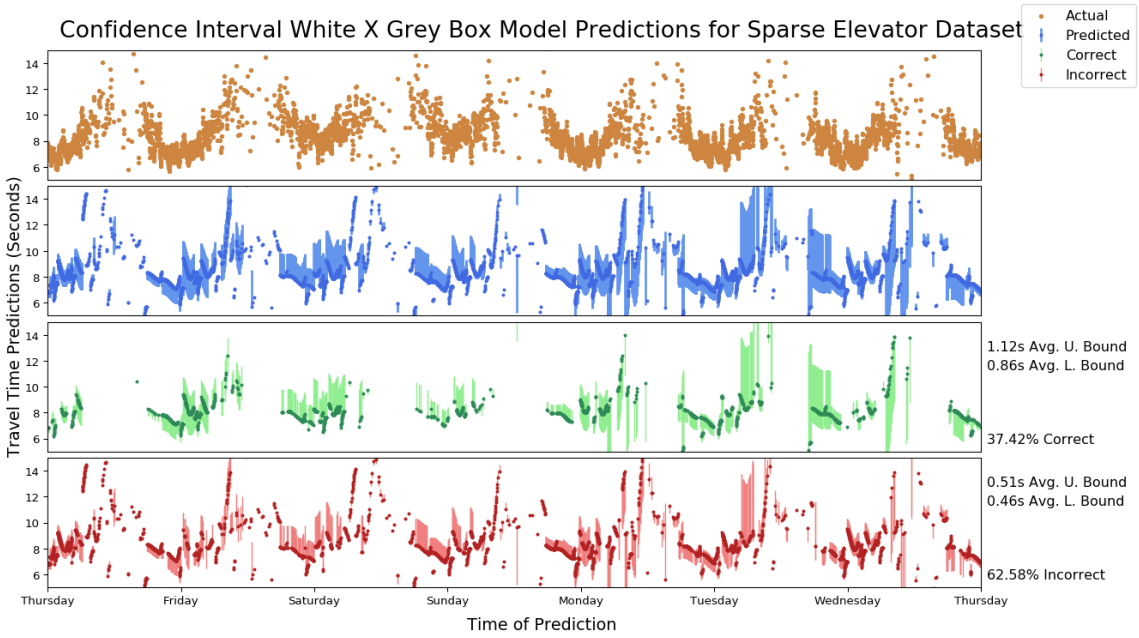

Figure A.11: Confidence Interval White X Grey Box Model Predictions For Sparse Elevator Dataset

B

Additional Results for H-BRS Hallway Travel Time Dataset

B.1 Hallway A

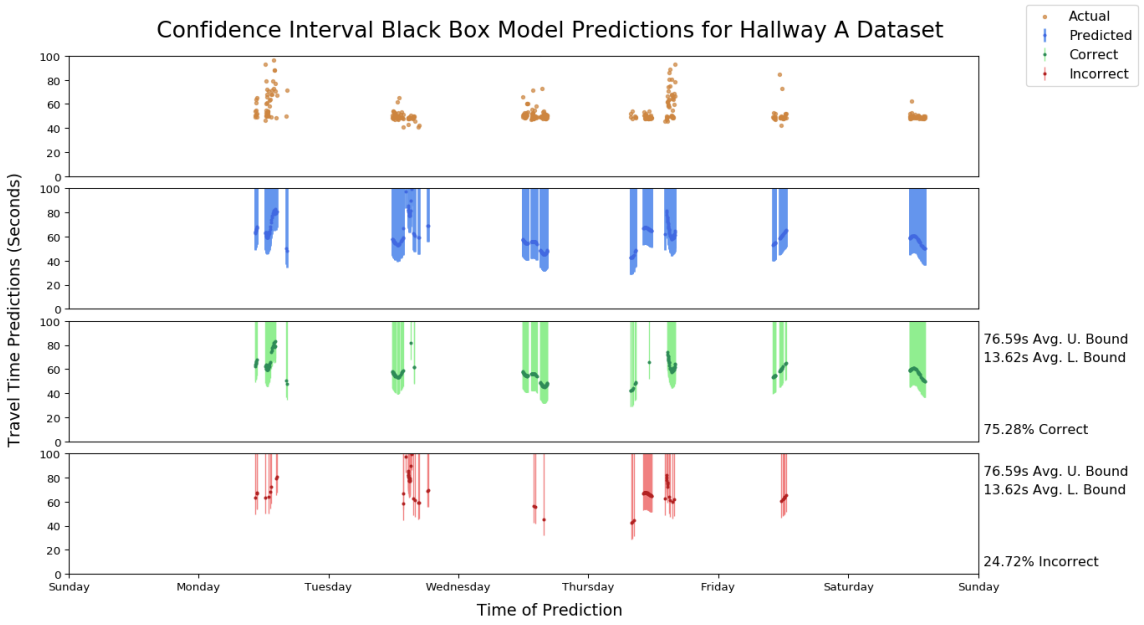

Figure B.1: Confidence Interval Black Box Model Predictions For Hallway Dataset A

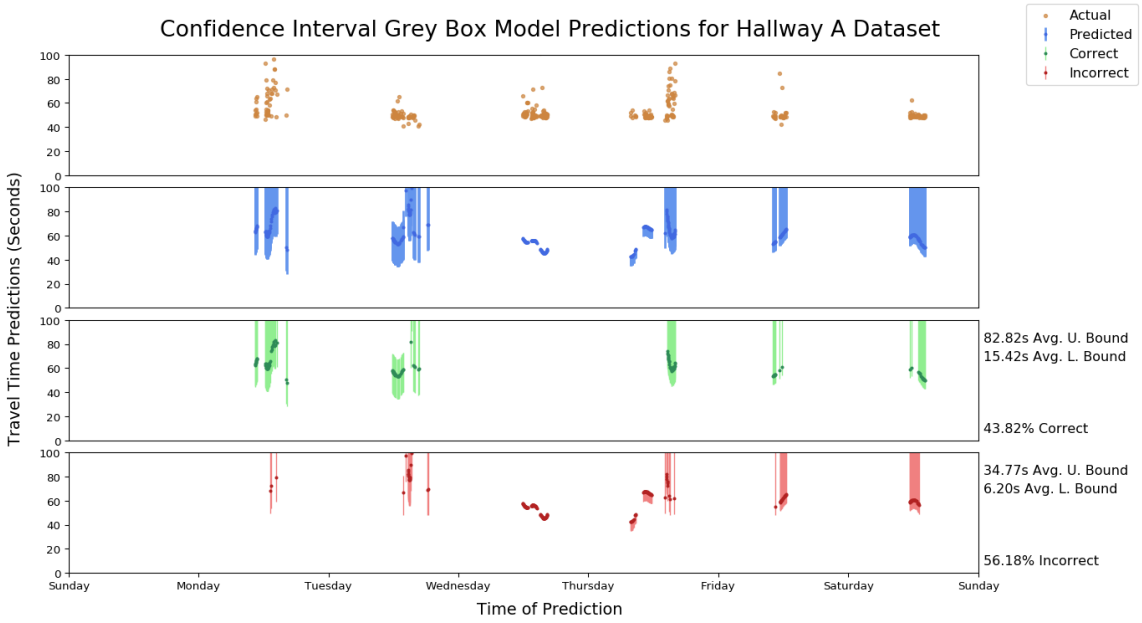

Figure B.2: Confidence Interval Grey Box Model Predictions For Hallway Dataset A

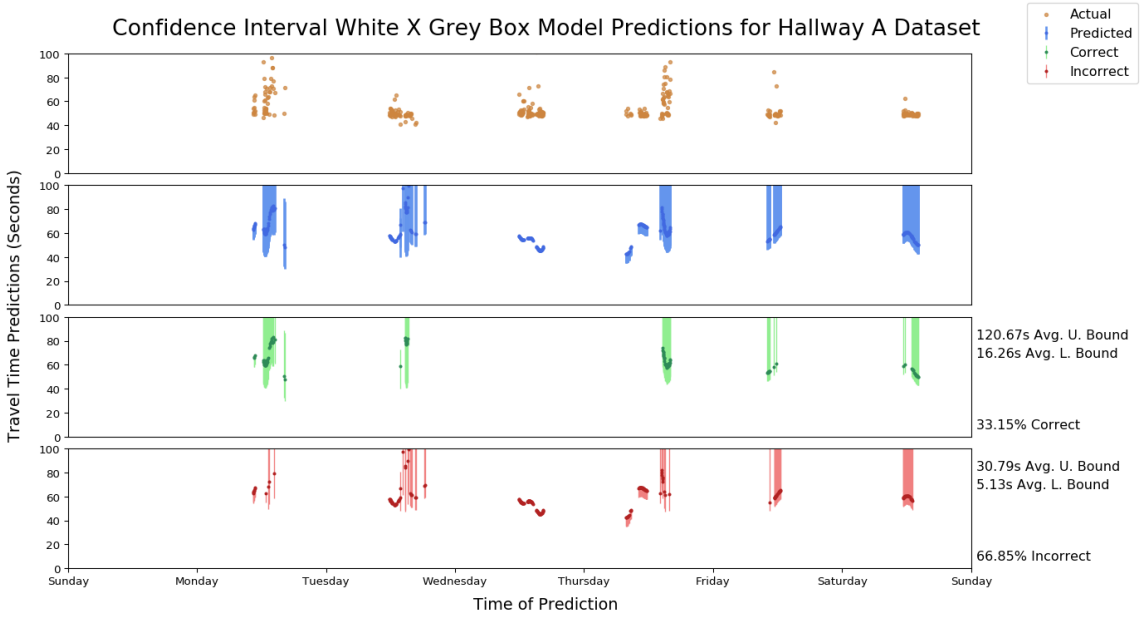

Figure B.3: Confidence Interval White X Grey Box Model Predictions For Hallway Dataset A

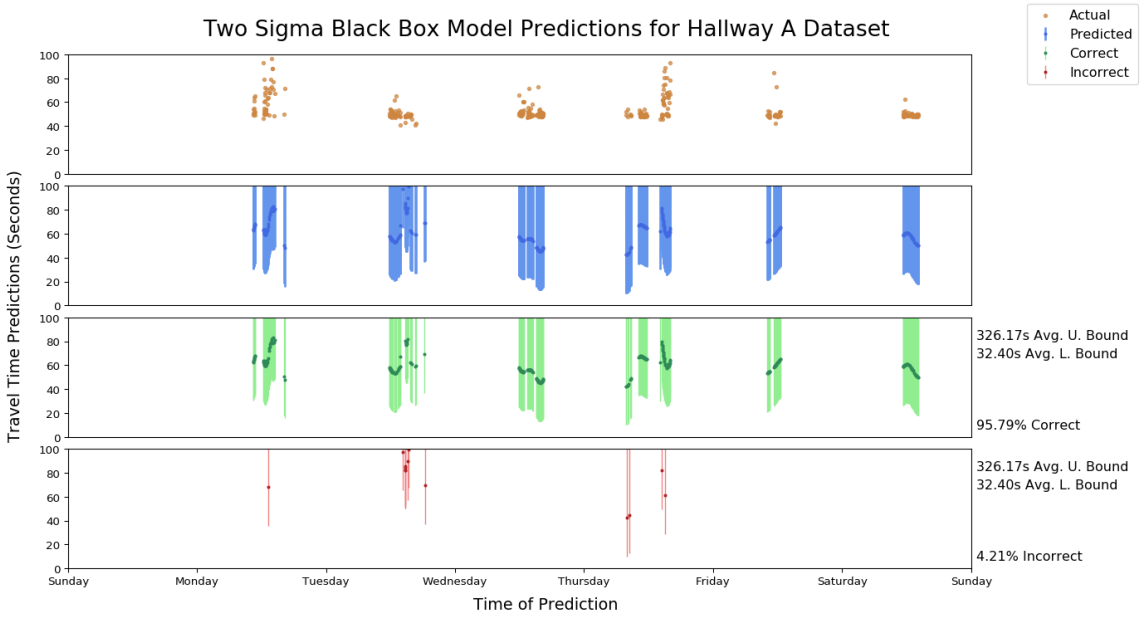

Figure B.4: Two Sigma Black Box Model Predictions For Hallway Dataset A

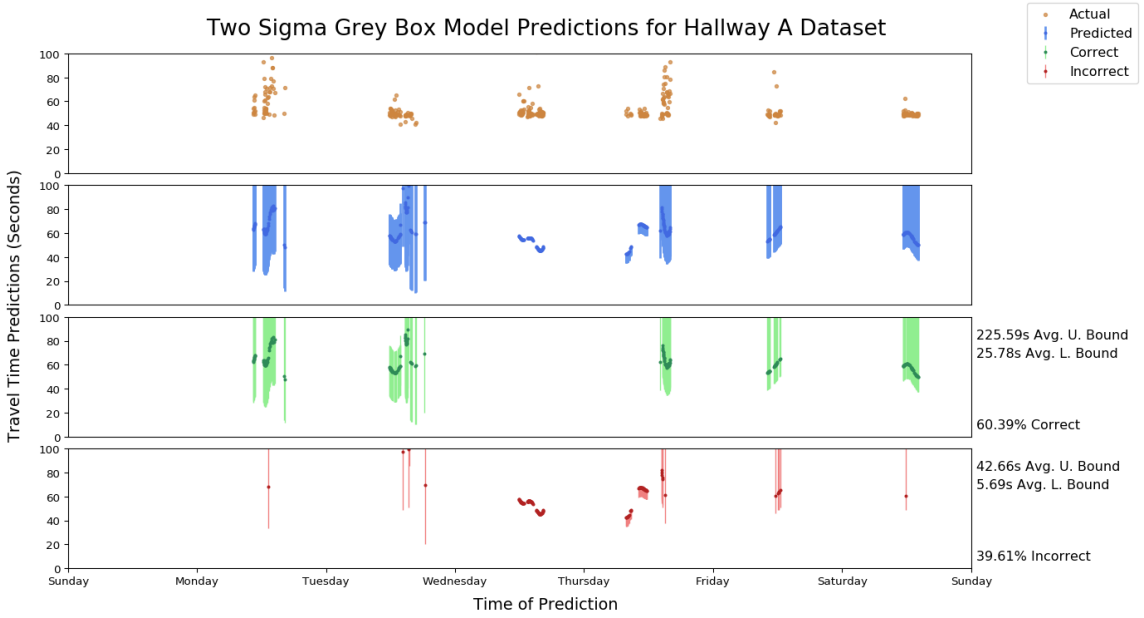

Figure B.5: Two Sigma Grey Box Model Predictions For Hallway Dataset A

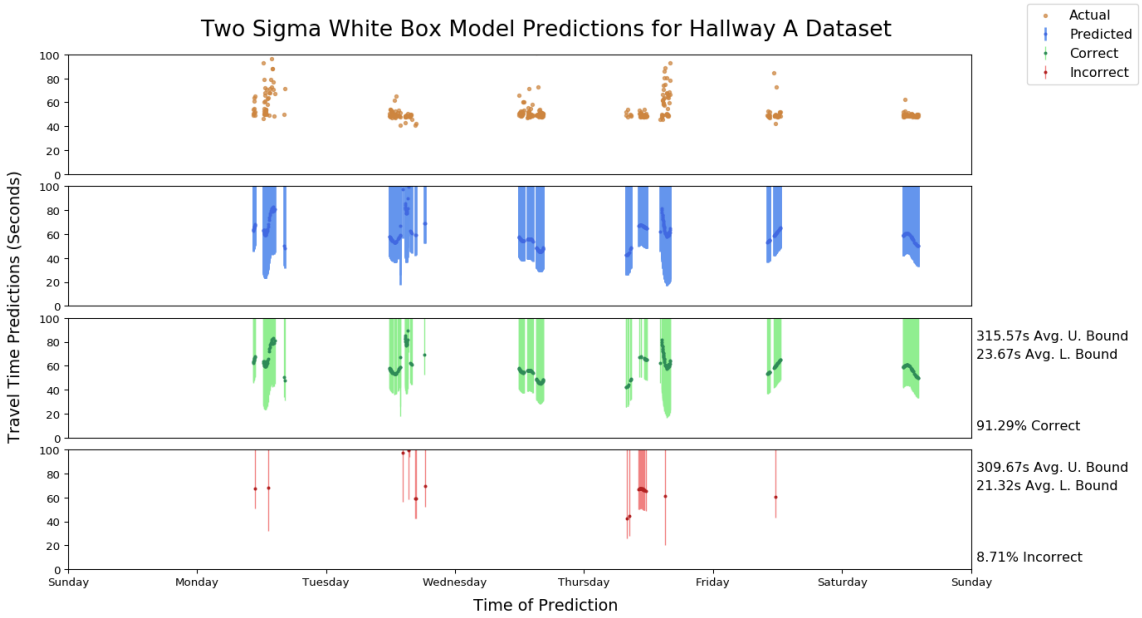

Figure B.6: Two Sigma White Box Model Predictions For Hallway Dataset A

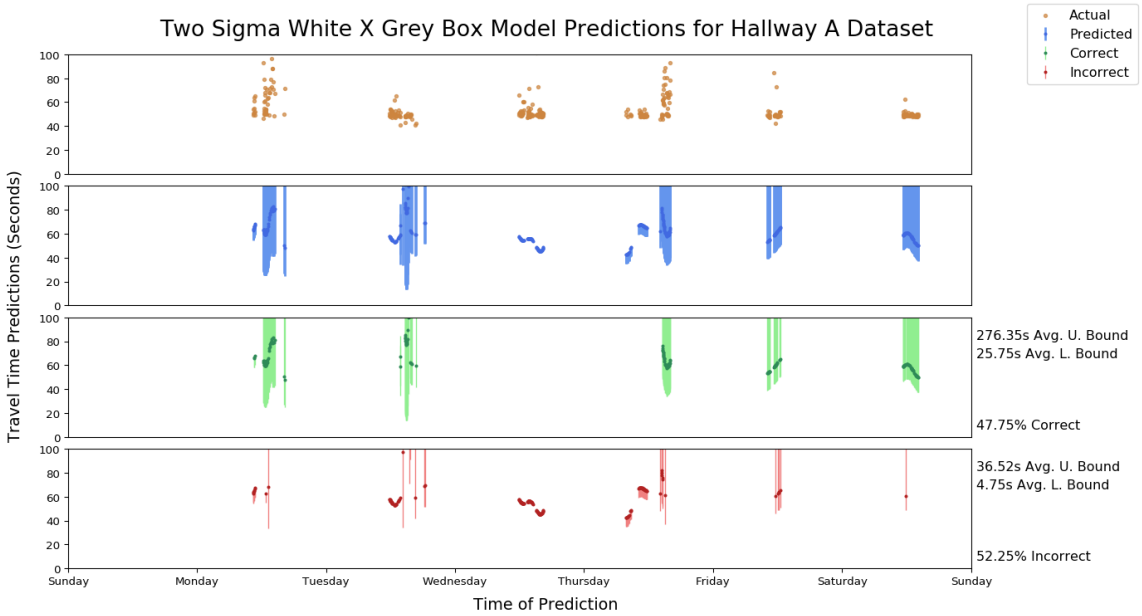

Figure B.7: Two Sigma White X Grey Box Model Predictions For Hallway Dataset A

B.2 Hallway B

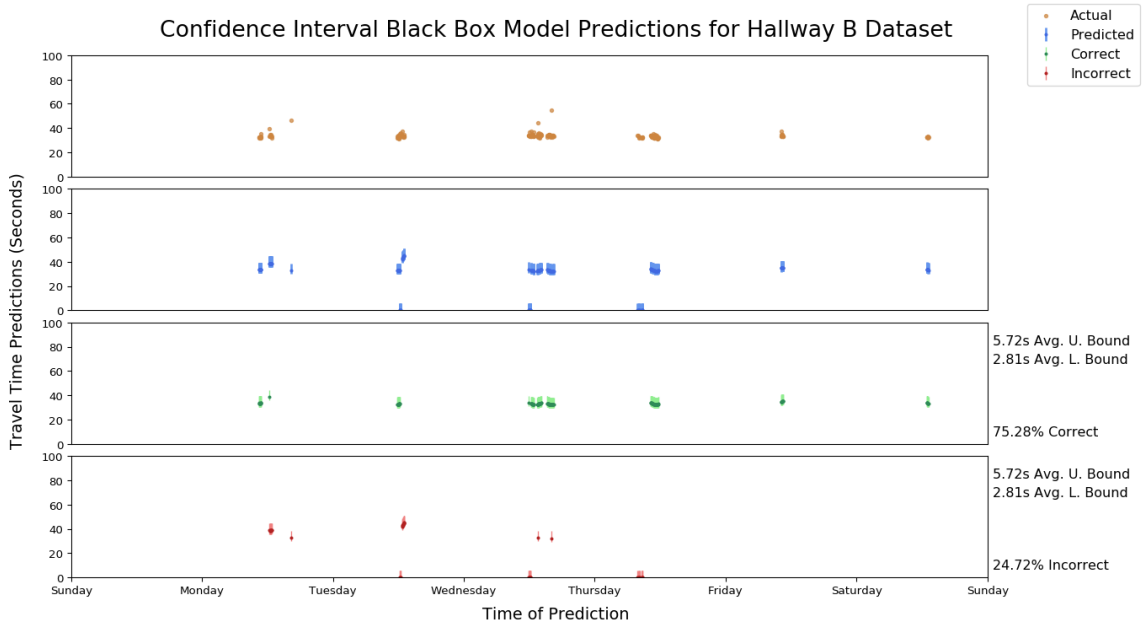

Figure B.8: Confidence Interval Black Box Model Predictions For Hallway Dataset B

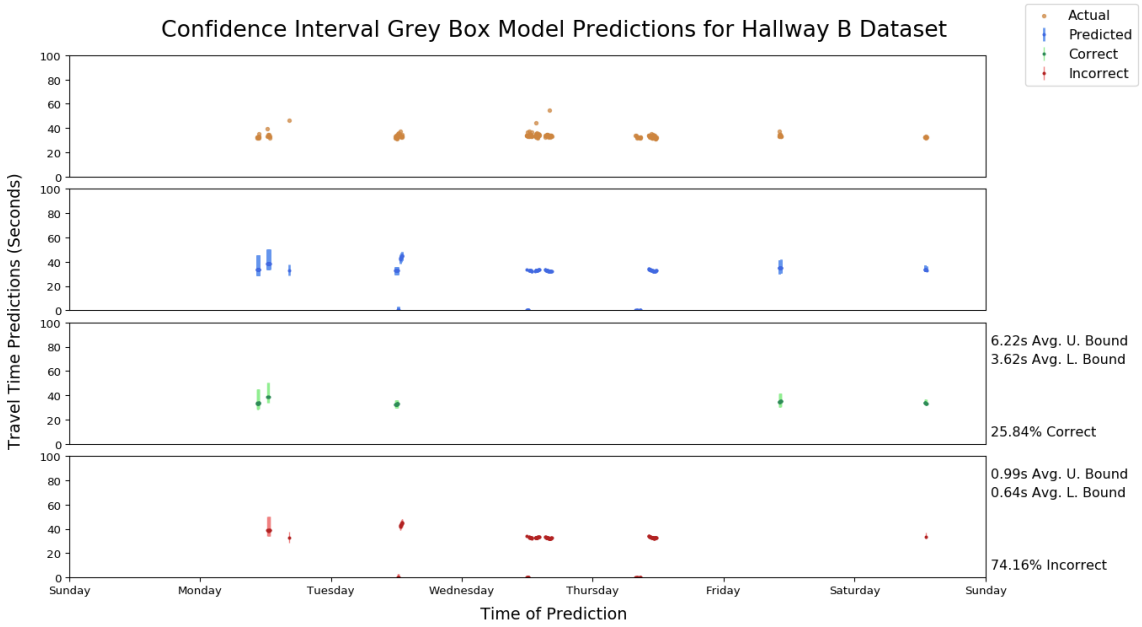

Figure B.9: Confidence Interval Grey Box Model Predictions For Hallway Dataset B

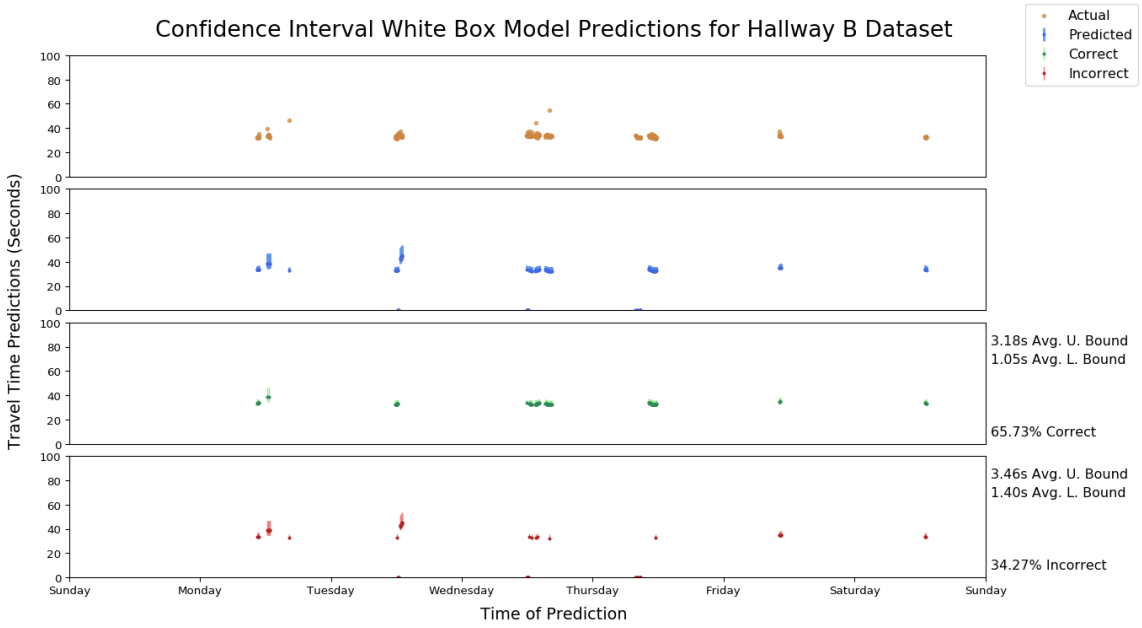

Figure B.10: Confidence Interval White Box Model Predictions For Hallway Dataset B

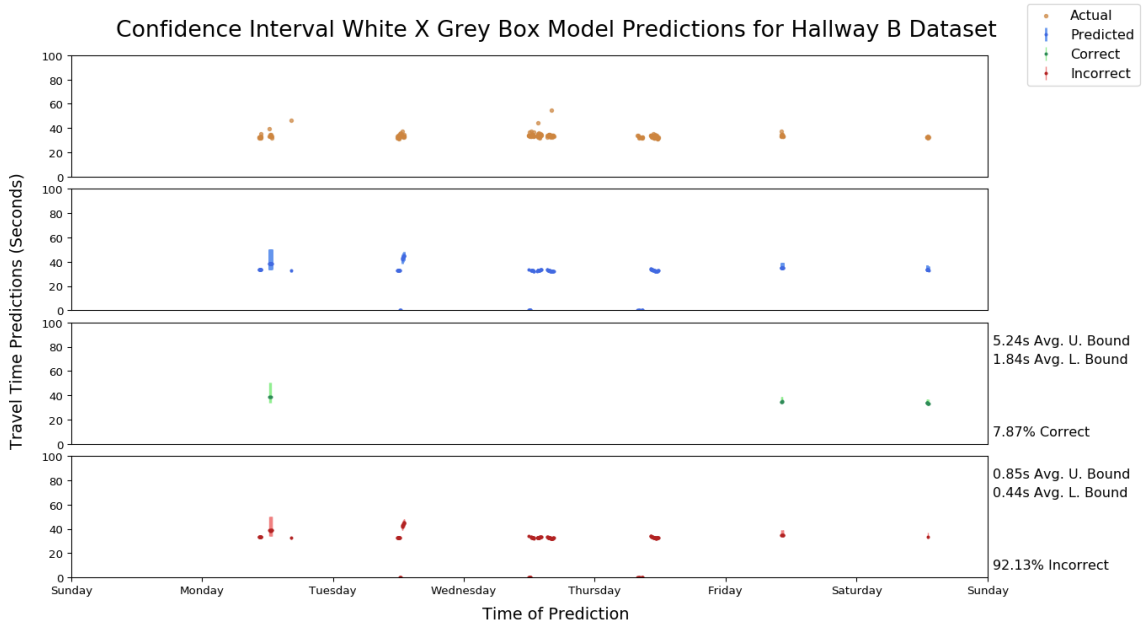

Figure B.11: Confidence Interval White X Grey Box Model Predictions For Hallway Dataset B

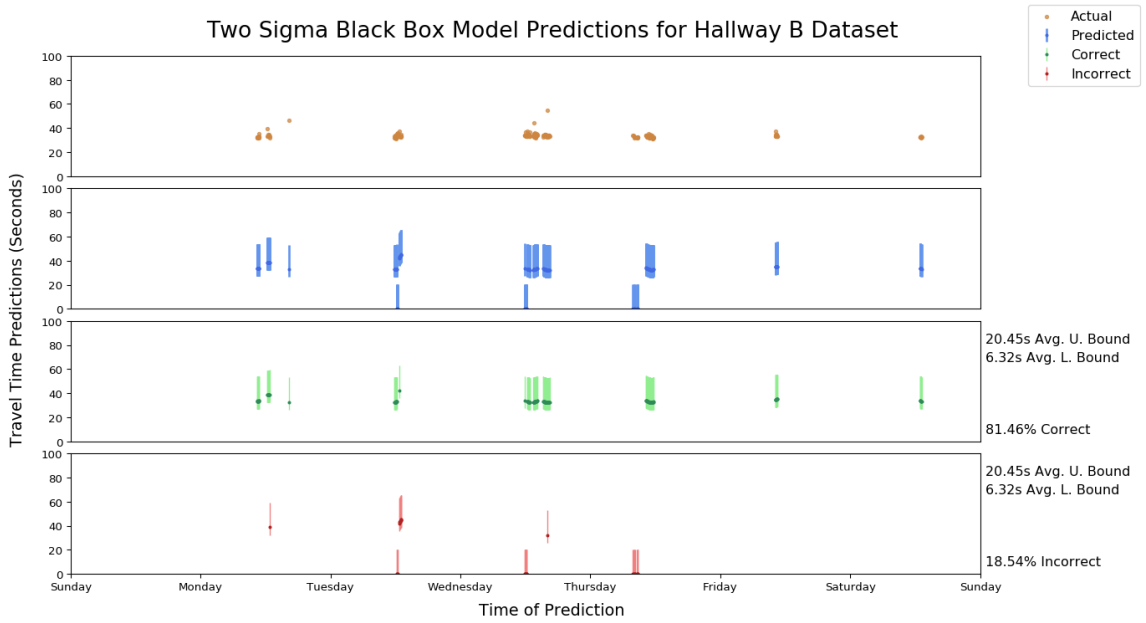

Figure B.12: Two Sigma Black Box Model Predictions For Hallway Dataset B

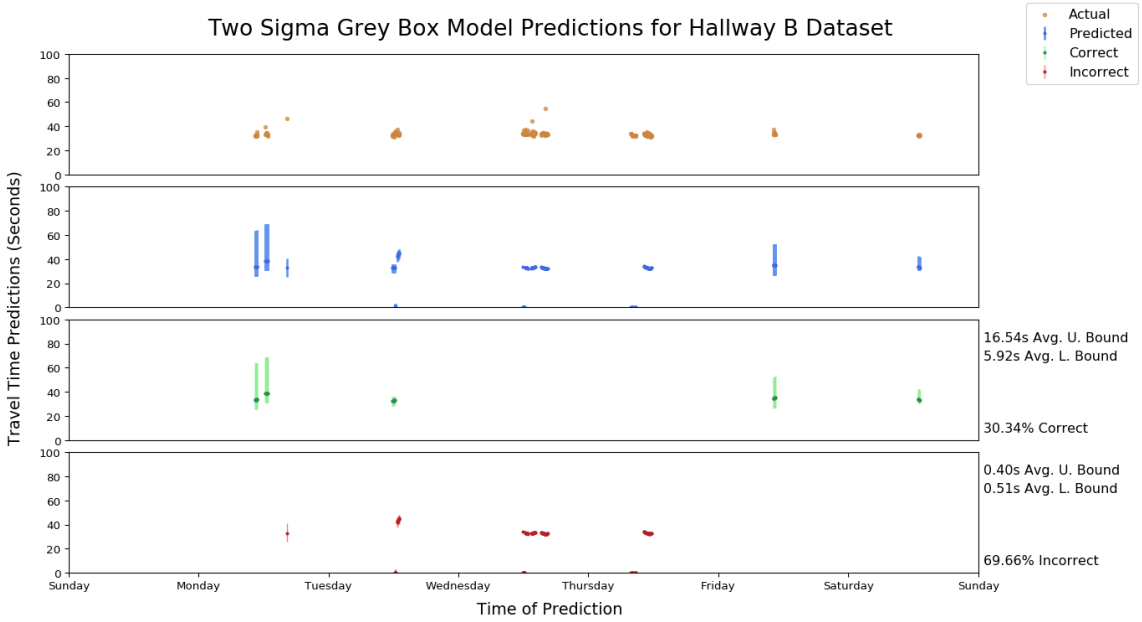

Figure B.13: Two Sigma Grey Box Model Predictions For Hallway Dataset B

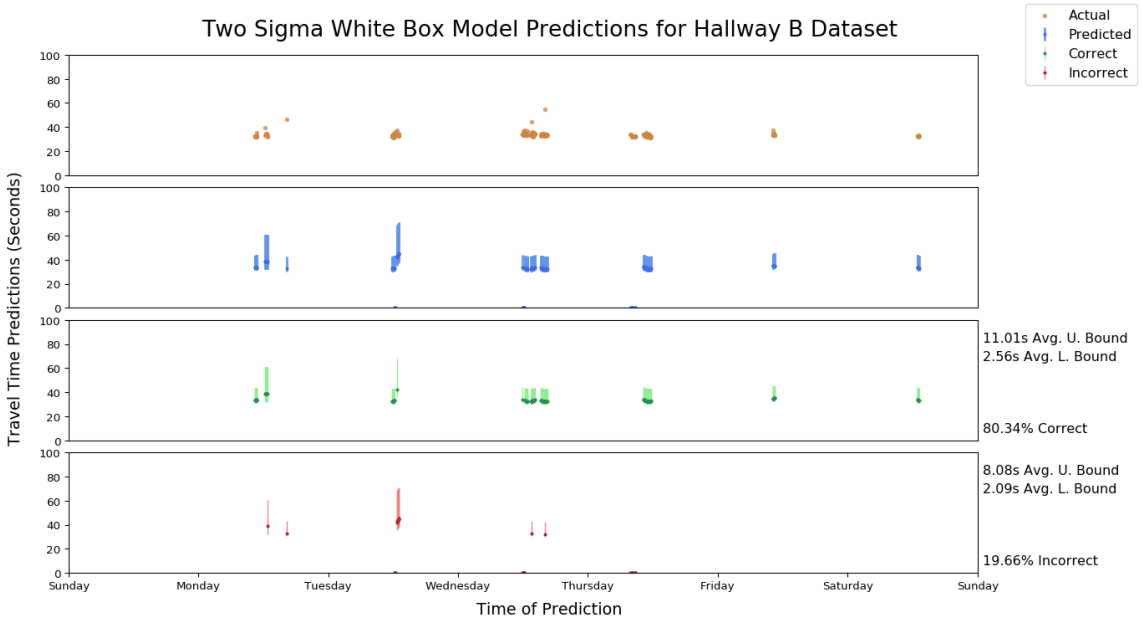

Figure B.14: Two Sigma White Box Model Predictions For Hallway Dataset B

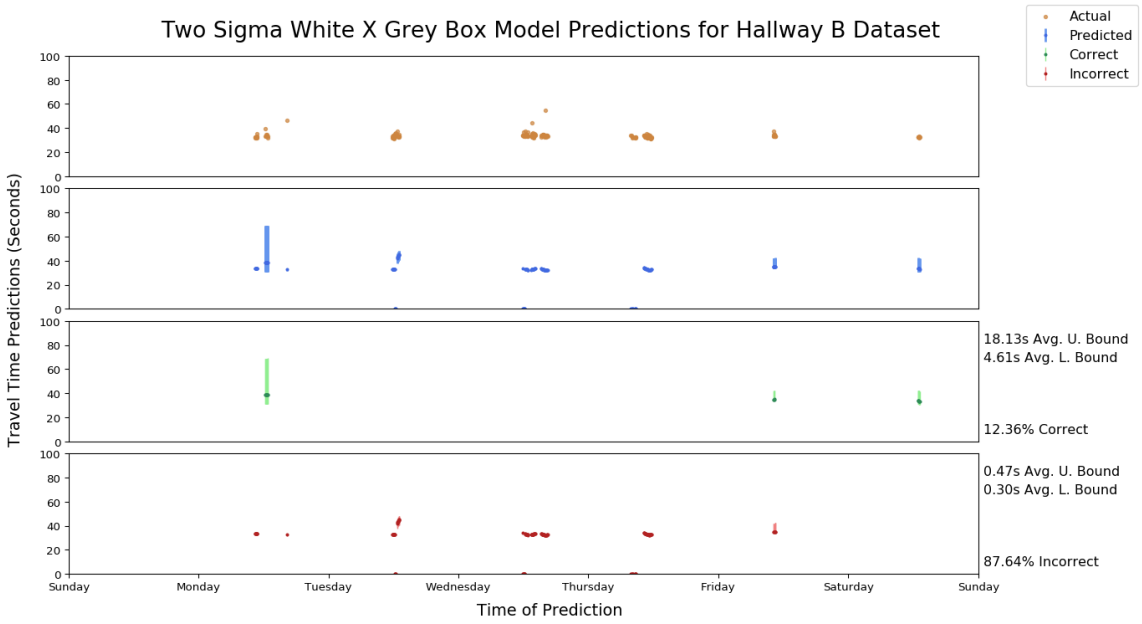

Figure B.15: Two Sigma White X Grey Box Model Predictions For Hallway Dataset B

B.3 Hallway C

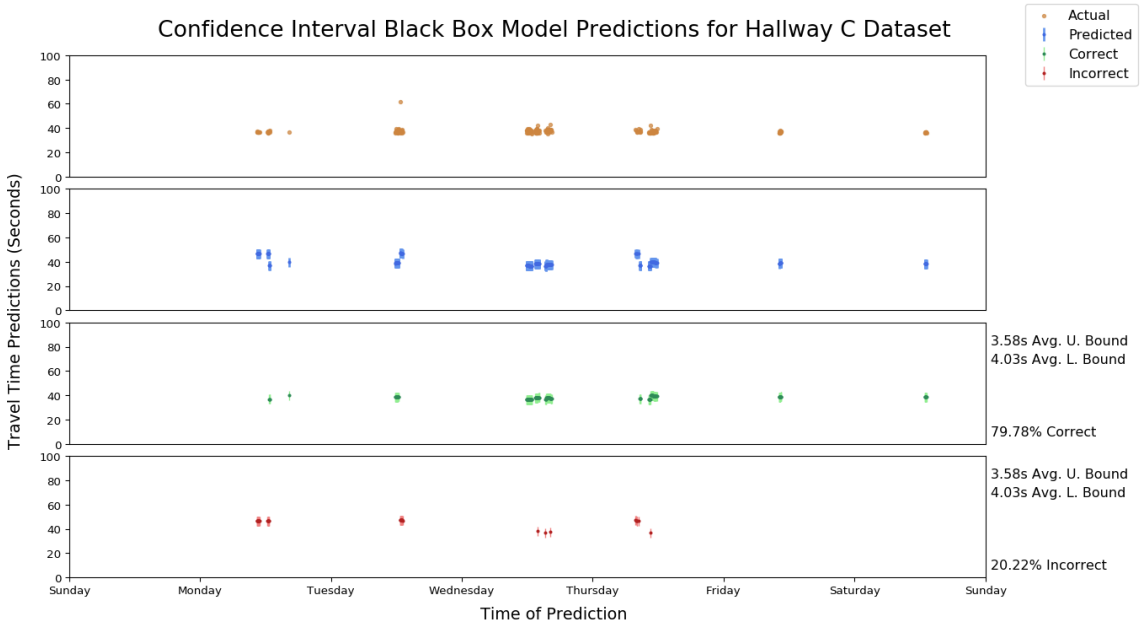

Figure B.16: Confidence Interval Black Box Model Predictions For Hallway Dataset C

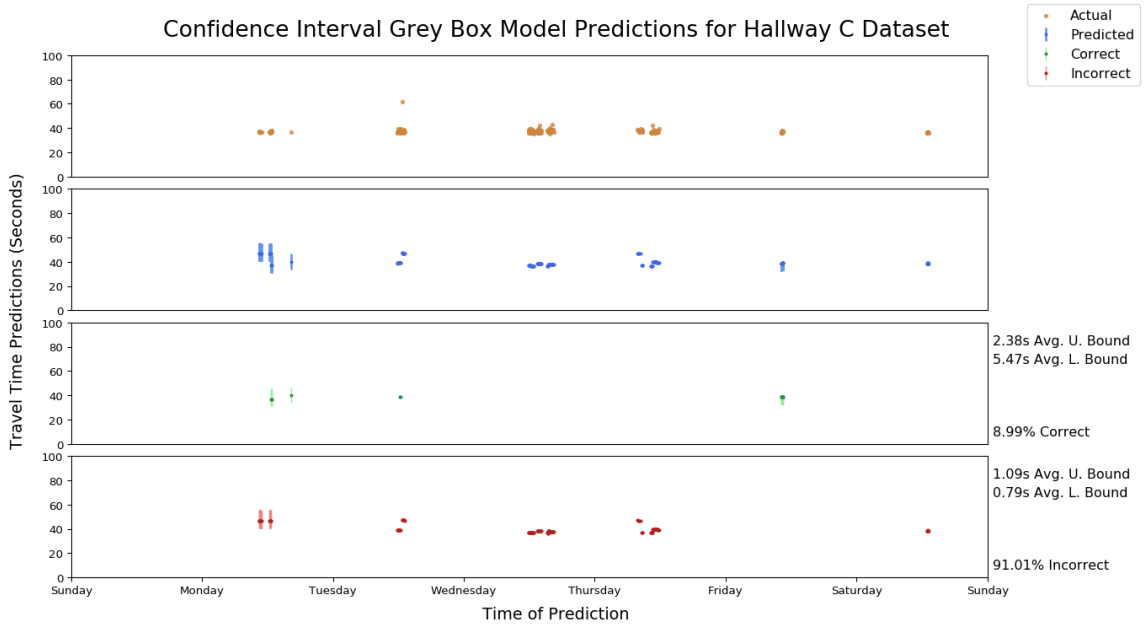

Figure B.17: Confidence Interval Grey Box Model Predictions For Hallway Dataset C

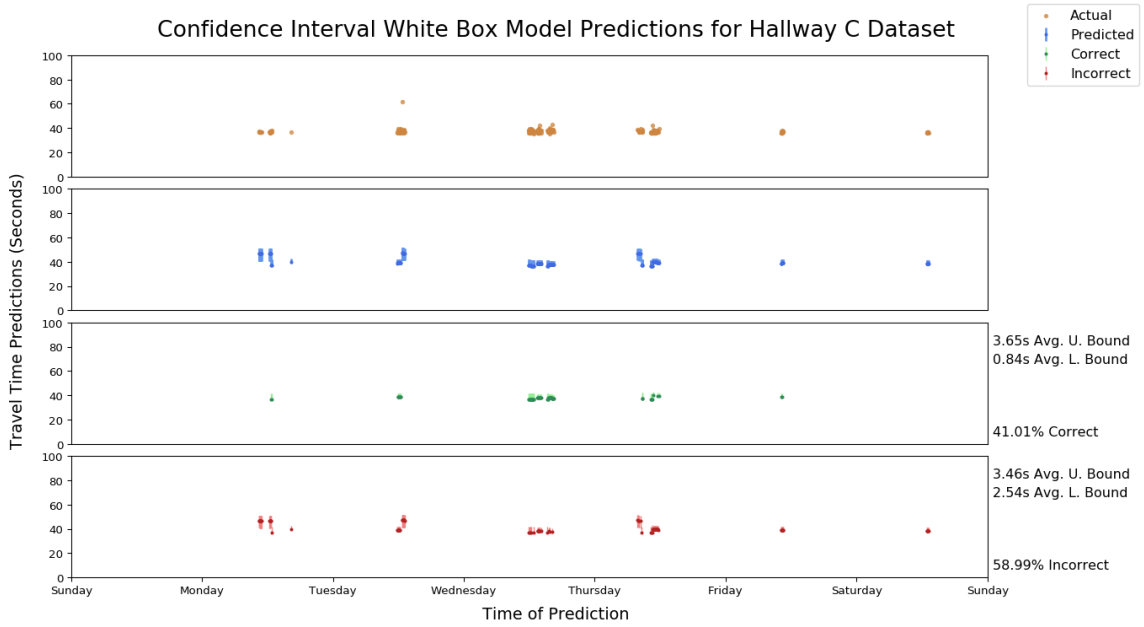

Figure B.18: Confidence Interval White Box Model Predictions For Hallway Dataset C

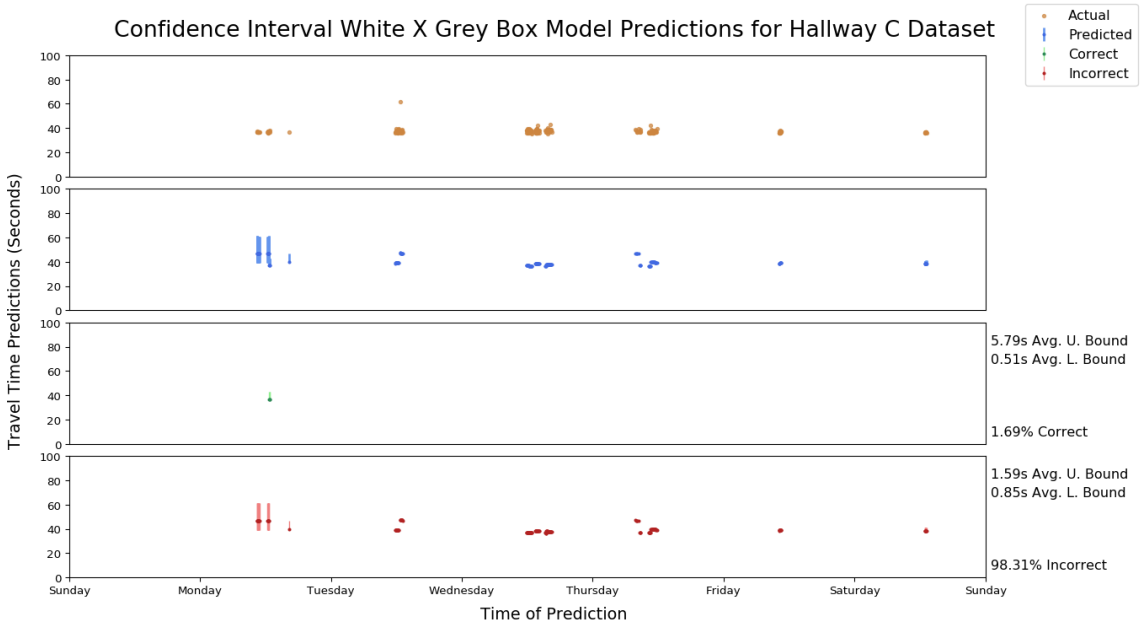

Figure B.19: Confidence Interval White X Grey Box Model Predictions For Hallway Dataset C

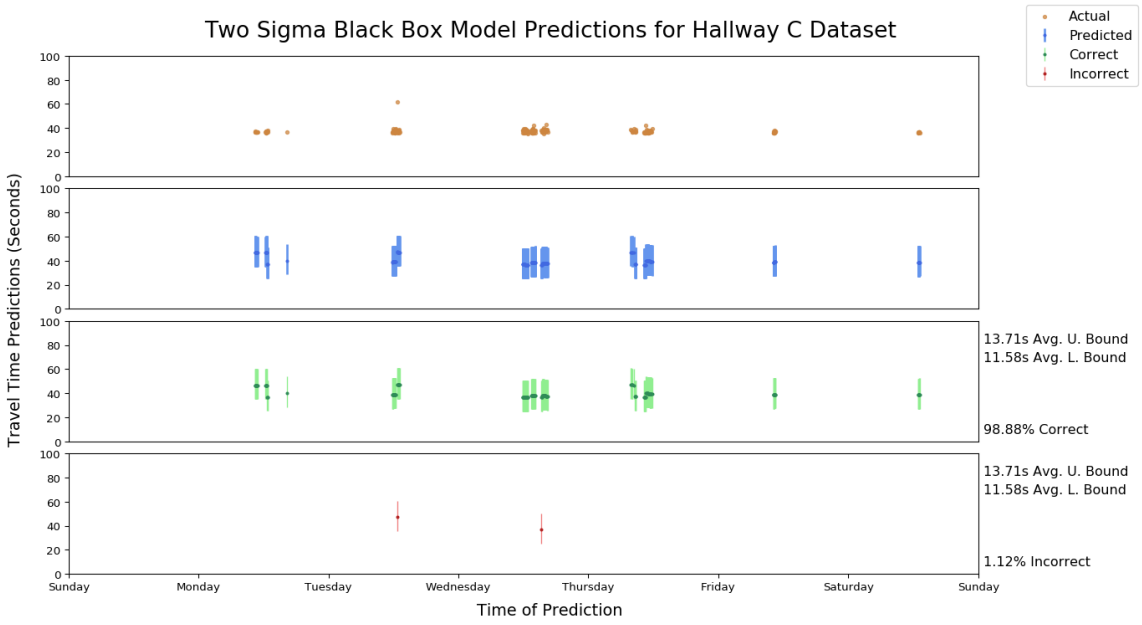

Figure B.20: Two Sigma Black Box Model Predictions For Hallway Dataset C

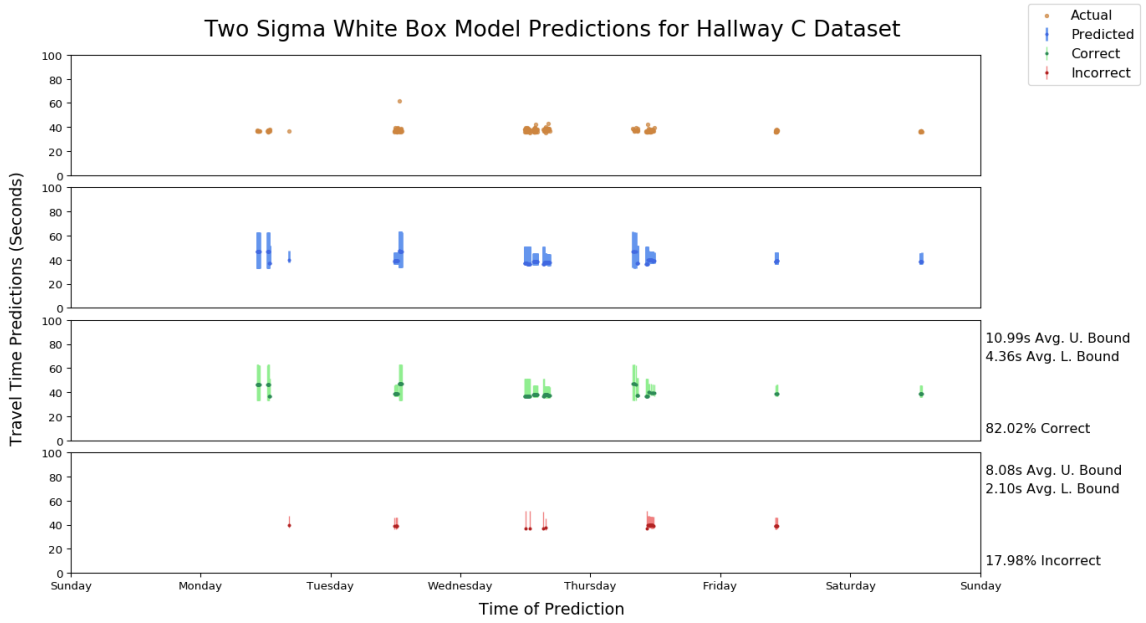

Figure B.21: Two Sigma White Box Model Predictions For Hallway Dataset C

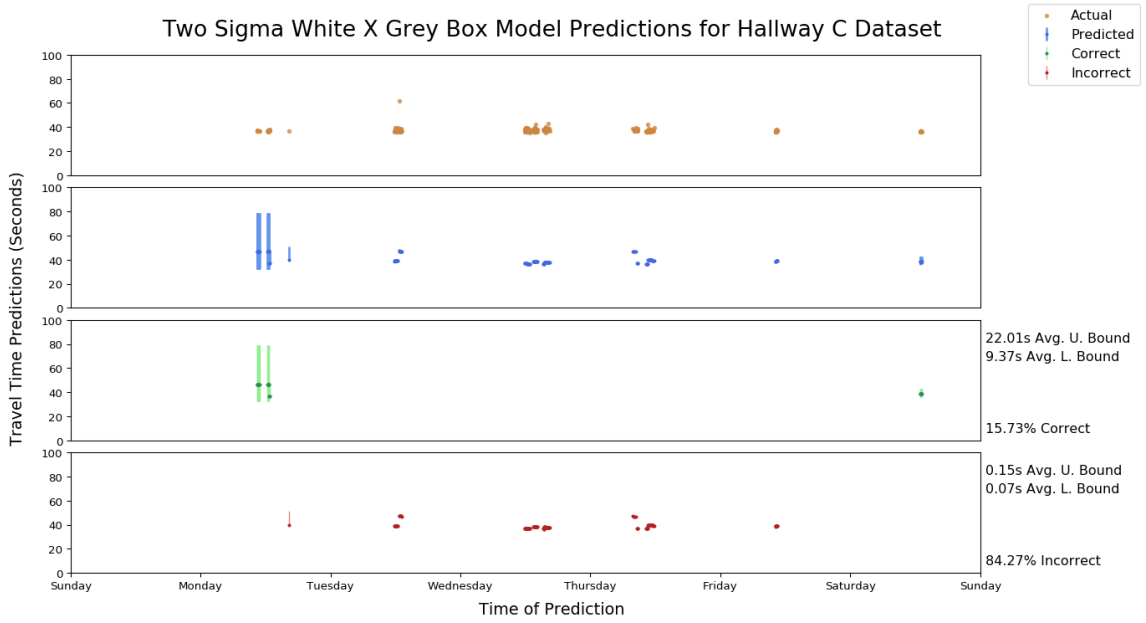

Figure B.22: Two Sigma White X Grey Box Model Predictions For Hallway Dataset C

B.4 Hallway D

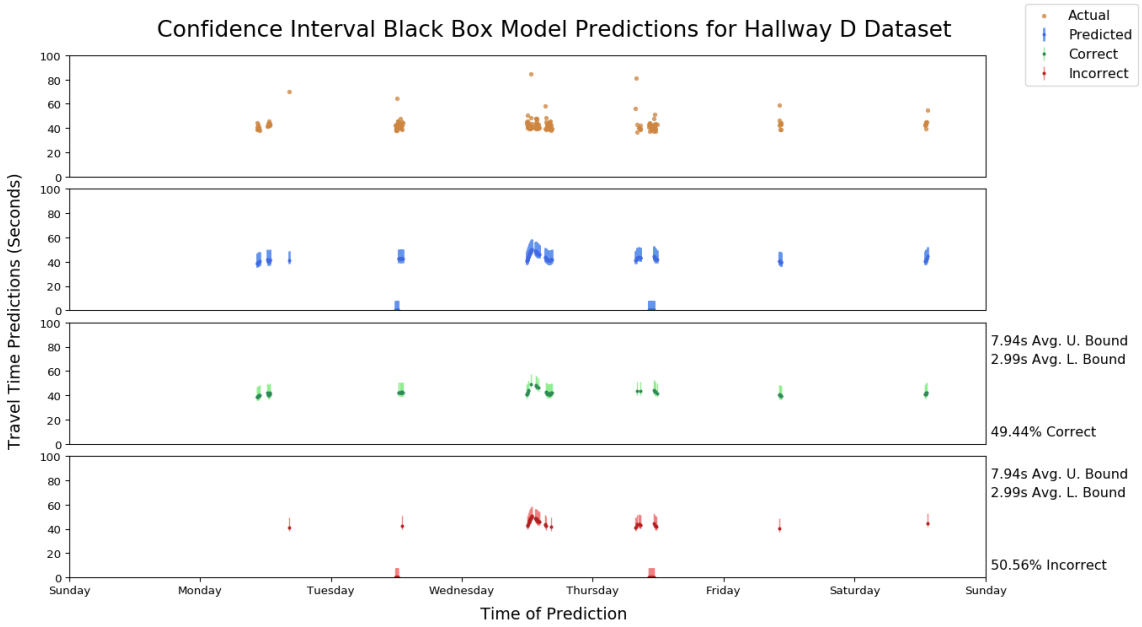

Figure B.23: Confidence Interval Black Box Model Predictions For Hallway Dataset D

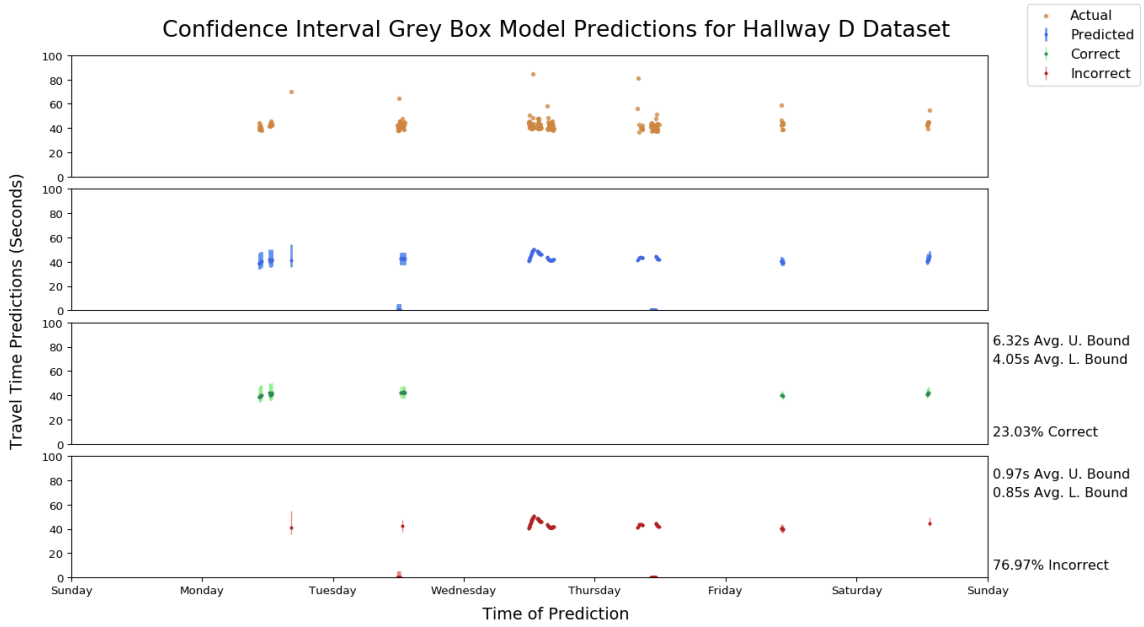

Figure B.24: Confidence Interval Grey Box Model Predictions For Hallway Dataset D

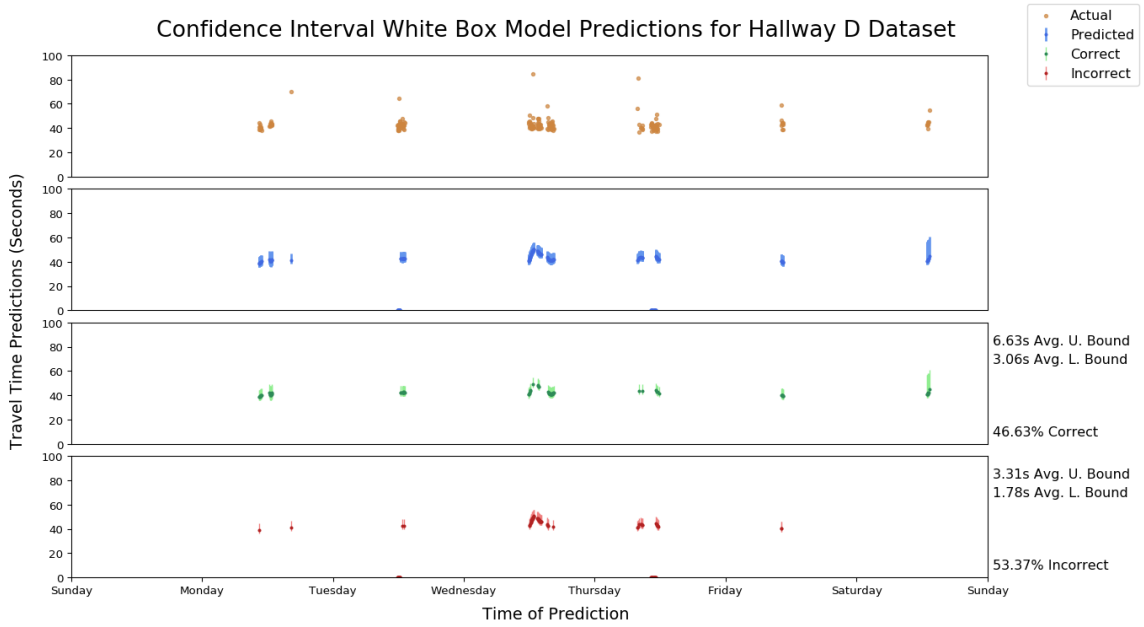

Figure B.25: Confidence Interval White Box Model Predictions For Hallway Dataset D

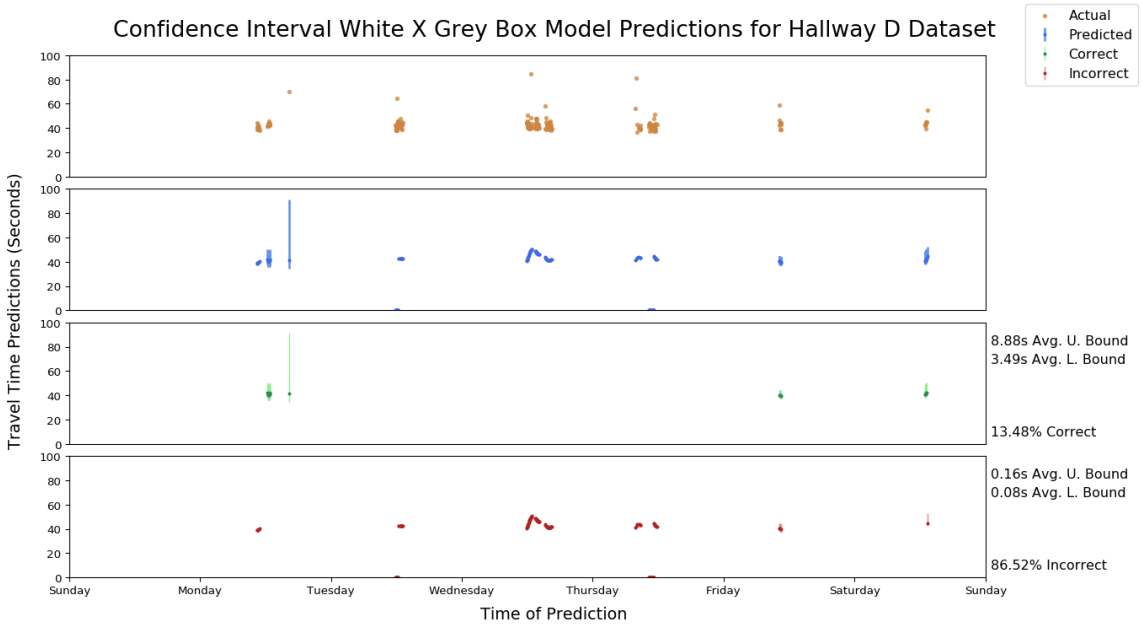

Figure B.26: Confidence Interval White X Grey Box Model Predictions For Hallway Dataset D

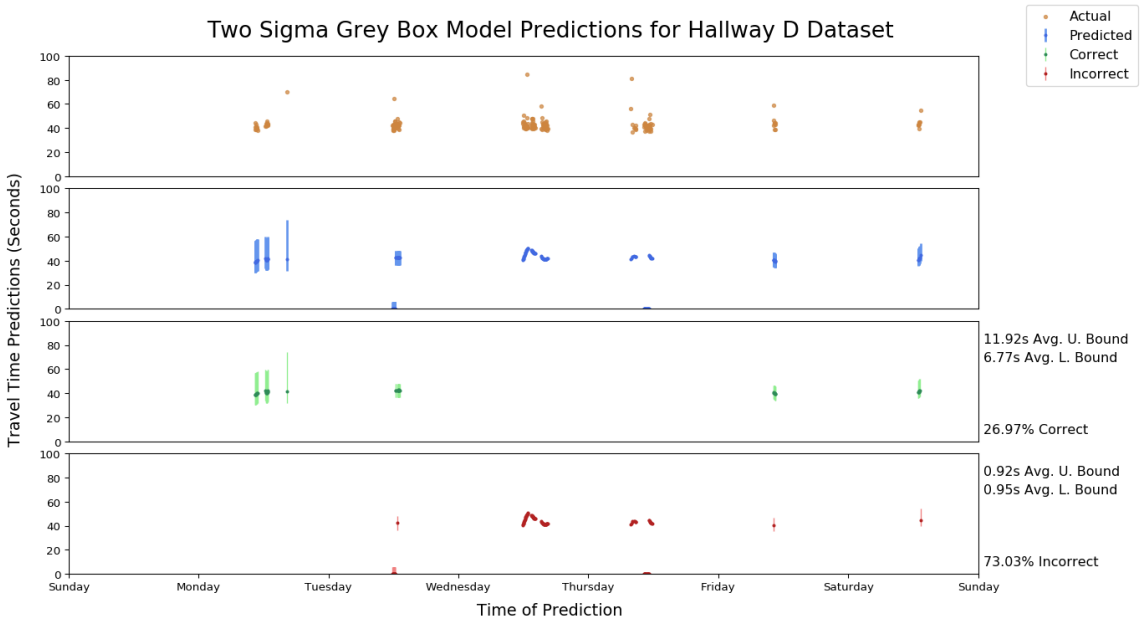

Figure B.27: Two Sigma Grey Box Model Predictions For Hallway Dataset D

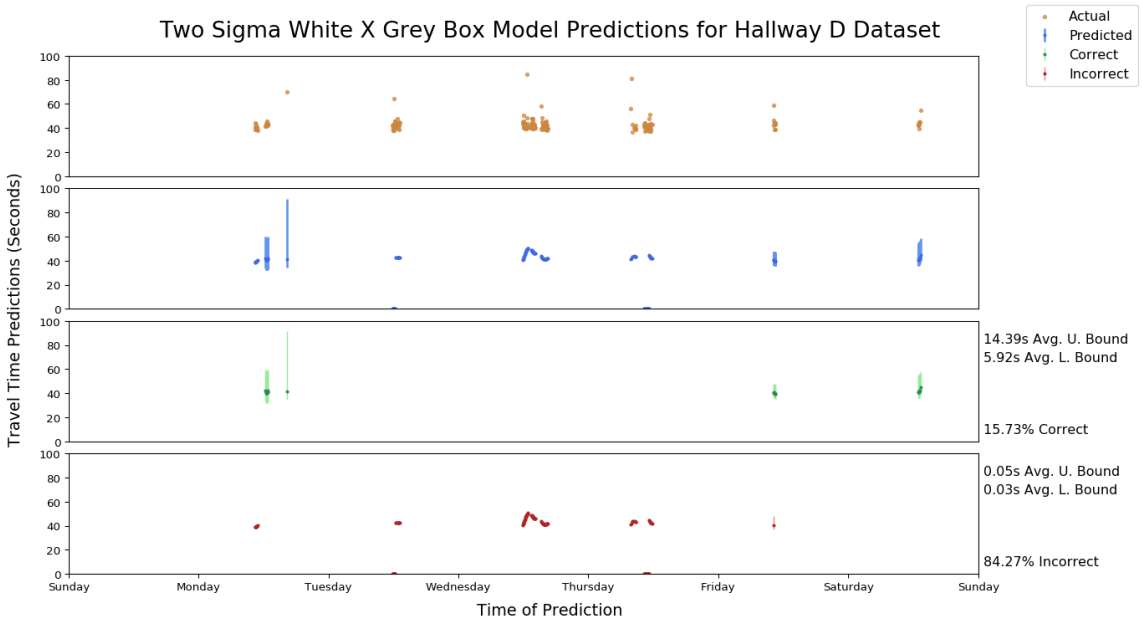

Figure B.28: Two Sigma White X Grey Box Model Predictions For Hallway Dataset D

C

Additional Results for Multi-Model Fusion Sparse Elevator Travel Time Dataset

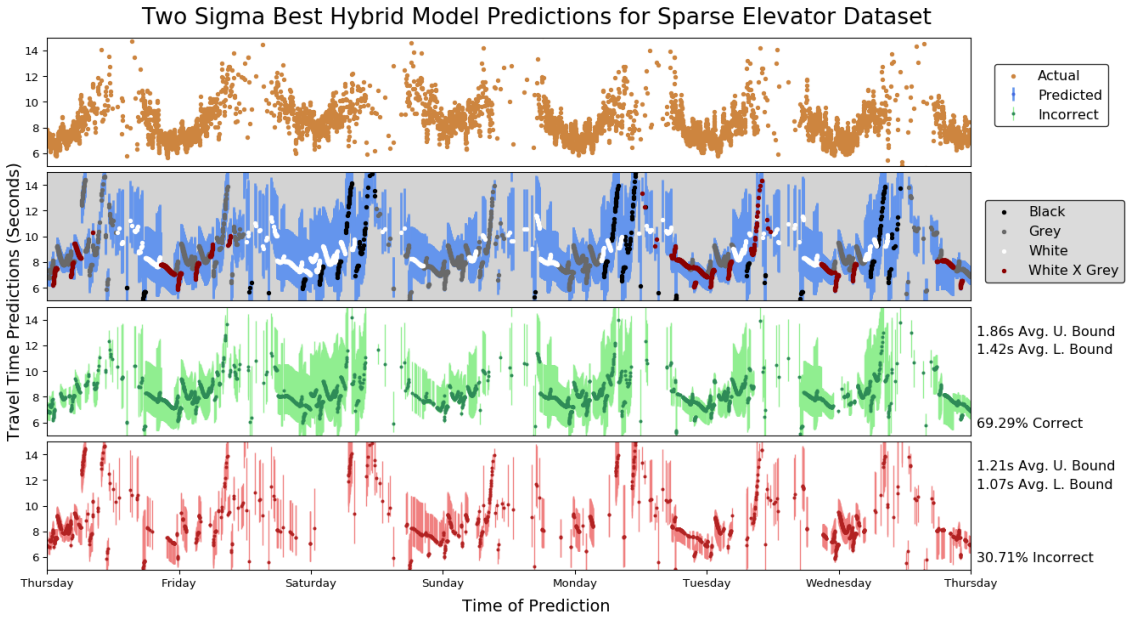

Figure C.1: Two Sigma Best Hybrid Model Predictions for Sparse Elevator Dataset

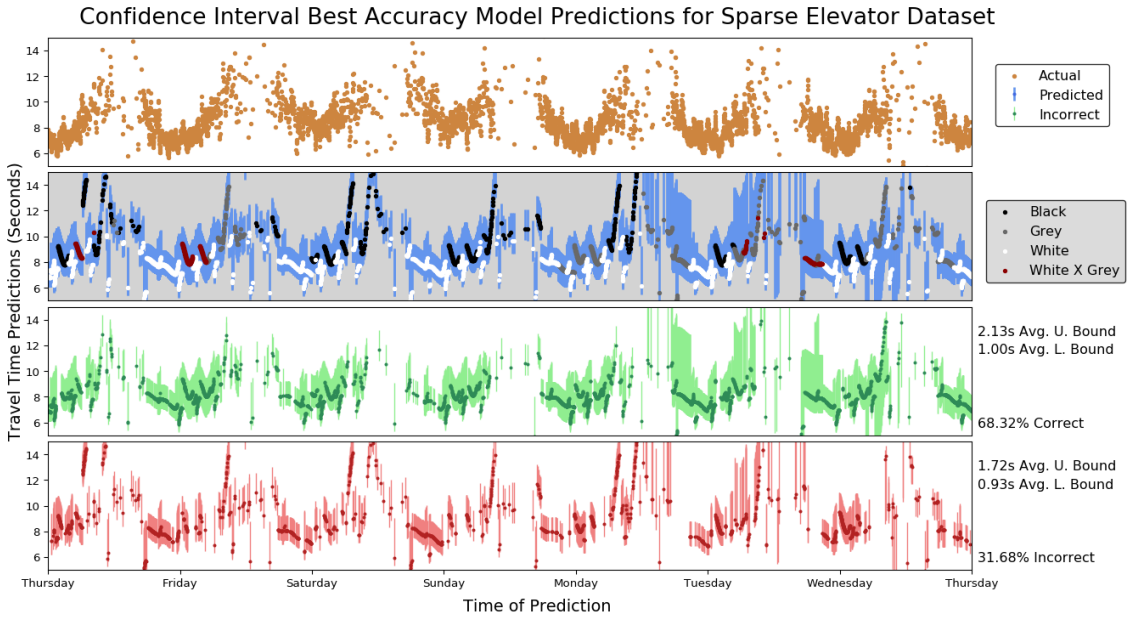

Figure C.2: Confidence Interval Best Accuracy Model Predictions for Sparse Elevator Dataset

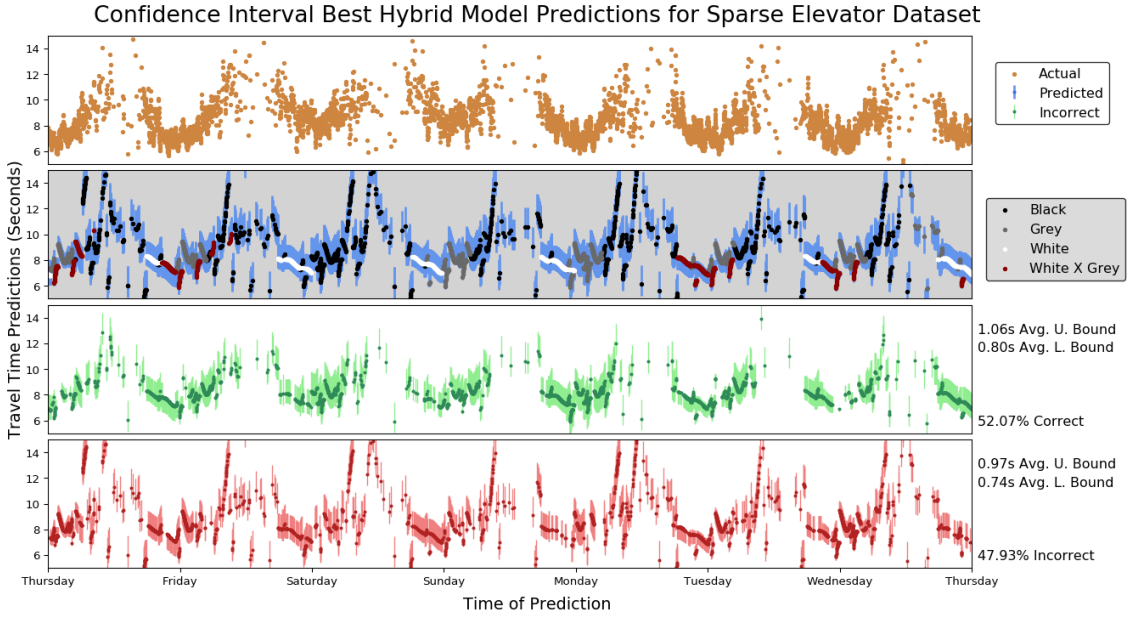

Figure C.3: Confidence Interval Best Hybrid Model Predictions for Sparse Elevator Dataset

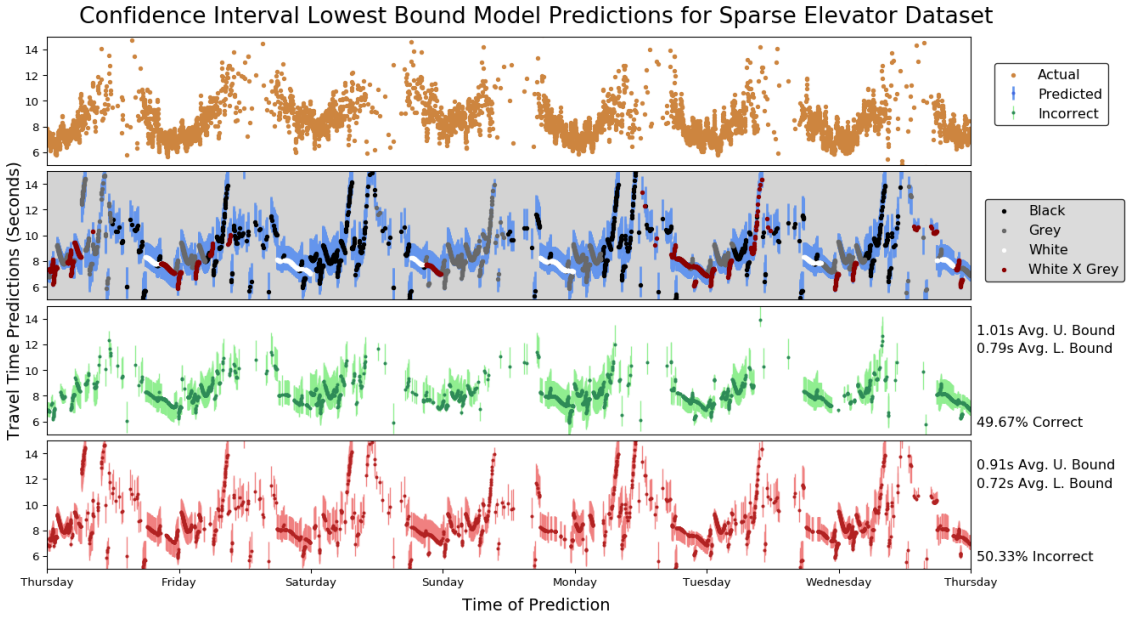

Figure C.4: Confidence Interval Lowest Bound Model Predictions for Sparse Elevator Dataset

References

[1] J.I. Allen, P.J. Somerfield, F.J. Gilbert. Quantifying Uncertainty in High-Resolution Coupled Hydrodynamic-Ecosystem Models. *Journal of Marine Systems*, 64:3–14, 2007.

[2] A. Kendall, Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017.

[3] A. Valentini, A. Micheli, A. Cimatti. Temporal planning with intermediate conditions and effects. *arXiv preprint arXiv:1909.11581*, 2019.

[4] B. Efron, R.J. Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1994.

[5] D.L. Shrestha, D.P. Solomatine. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19(2):225–235, 2006.

[6] E.O. Massey, E. Prassler, A.O. Sáinz, S. Blumenthal. Comparative Analysis of Techniques for Spatio-Temporal World Modeling. *Hochschule Bonn-Rhein-Sieg*, (January), 2019.

[7] H.G. Matthies. Quantifying uncertainty: Modern computational representation of probability and applications. In *Extreme man-made and natural hazards in dynamics of structures*, pages 105–135. Springer, 2007.

[8] J.M. Santos, T. Krajník, J.P. Fentanes, T. Duckett. Lifelong information-driven exploration to complete and refine 4-d spatio-temporal maps. *IEEE Robotics and Automation Letters (RA-L)*, 1(2):684–691, 2016.

[9] J.M. Santos, T. Krajník, T. Duckett. Spatio-temporal exploration strategies for long-term autonomy of mobile robots. *Robotics and Autonomous Systems*, 88: 116–126, 2017.

[10] J.P. Fentanes, B. Lacerda, T. Krajník, Tomas N. Hawes, M, Hanheide. Now or later? predicting and maximising success of navigation actions from long-term experience. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1112–1117, 2015.

[11] J.P. Fentanes, G. Cielniak, C. Dondrup, T. Duckett. Spectral Analysis for Long-Term Robotic Mapping. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3706–3711, 2014.

[12] T.J Page Jr. Multivariate statistics: A vector space approach. *JMR, Journal of Marketing Research (pre-1986)*, 21(000002):236, 1984.

[13] J.R. Abrahams, D.A. Chu, G. Diehl, M. Knittel, J. Lin, W. Lloyd, J.C Boerkoel, F. Jeremy. Dream: An algorithm for mitigating the overhead of robust rescheduling. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 3–12, 2019.

[14] K. Lund, S. Dietrich, S. Chow, J.C. Boerkoel. Robust execution of probabilistic temporal plans. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[15] N. Hawes, et. al. The STRANDS Project: Long-Term Autonomy in Everyday Environments. *IEEE Robotics and Automation Magazine (RAM)*, 24:146–156, 2017.

[16] N.K. Suryadevara, S.C. Mukhopadhyay, R. Wang, R.K. Rayudu. Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Engineering Applications of Artificial Intelligence*, 26(10):2641–2652, 2013.

[17] Oxford University Press (OUP). Lexico.com, 2019.

[18] T. Krajník, J.P. Fentanes, J.M. Santos, and T. Duckett. FreMEn: Frequency Map Enhancement for Long-Term Mobile Robot Autonomy in Changing Environments. *IEEE Trans. on Robotics and Automation*, 33:964–977, 2017.

[19] T. Krajník, T. Vintr S. Molina, J.P. G. Cielniak, T. Duckett. Warped Hypertime Representations for Long-term Autonomy of Mobile Robots. 2018. URL `http://arxiv.org/abs/1810.04285`.

[20] T. Vintr, Z. Yan, T. T. Krajnık. Spatio-temporal representation for long-term anticipation of human presence in service robotics. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[21] T.J. DiCiccio, B. Efron. Bootstrap confidence intervals. *Statistical science*, pages 189–212, 1996.

[22] Y. Geifman, G. Uziel, R. El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. *arXiv preprint arXiv:1805.08206*, 2018.