Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project

Comparative Analysis of Techniques for Spatio-Temporal World Modeling

Ethan 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

Erwin Prassler
Argentina Ortega
Sebastian Blumenthal

January 2018

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 Massey

Abstract

Your abstract

Acknowledgements

Thanks to ....

Contents

List of Figures

List of Tables

1

Introduction

A robots world model is its internal representation of its environment that allows it to reason and make decisions. This ability to make decisions, and thus how well a robot performs in an environment, can be directly correlated to the quality of the model that a robot contains. Historically, these models have been static two or three dimensional representations, but within the past decade or two multiple methods have been developed to introduce an additional dimension to these maps, the dimension of time. The inclusion of times allows for a robot to make decisions about when it may want to accomplish a task, or perhaps know to avoid a certain area at a given time. In the simple case of an office, a robot may learn to avoid areas of high traffic around the cafeteria during lunch time or learn that a shortcut between to builds is open, but only during work hours. These simple examples illustrate the type of knowledge and efficiency that can be gleaned by introducing a temporal component to a robots world model. This new type of world model has come to be known as a spatio-temporal world model as it is a model of the world that contains spatial information, that of the physical environment, as well as temporal information, how the environment changes through time.

Automatically guided vehicles (AVGs), often present in the field of logistics, stand to benefit a great deal from these improvements in world modeling. Moving logistics from point A to point B is an extremely common task in a wide variety of domains spanning industrial, commercial, and even residential applications. AVGs, have most prominently been used in industrial settings for a few decades already, but have been relegated to a discrete and limited set of predictable tasks. This is especially true when logistics must be transported through a particularly dynamic or human environment.

Recent work into introducing a temporal component to this world models has already begun, and is showing great promise. A variety of methods have been introduced to allow for an AVG to observe and make predictions about it's environment through time. However, since this field is relatively knew, and with new advancements and approaches being introduce even within the past view years, it is becoming increasingly harder to evaluate or choose between the different spatio-temporal world modeling options. It is for this reason that a method, or set of criteria, be devised for comparing and contrasting the variety of solutions. The analysis of these methods will not only allow for others to choose the most fitting approach for a given environment, but also expose deficiencies in the current approaches and guide future research efforts. Improvements in this field will ultimately result in more flexible AVGs that can operate in a wider variety of environments and for longer periods of autonomy.

1.1 Challenges and Difficulties

Historically, world modeling techniques could be thought of as simply a mapping and path planning problem in either two dimensional or three dimensional space. These problems have been studied for decades and thus there already exist a handful of well known solutions, each with there own advantages and disadvantages. However, with the fairly recent introduction of the fourth dimension, time, into the equation there has been the introduction of a number of different methods.

The early and simplistic approaches to introducing temporal components into world models started as early as 2002 [1] but within the past decade or so there has been an uptick both in the number of different approaches and the complexity of the methods. [**?**] [TODO ADD OTHER CITATIONS] With this increase in complexity and variety of approaches combined with a lack of historical perspective and analysis, it can be daunting task to select the 'correct' or even a well fitting spatio-temporal world model for a new project. A few papers have introduced some simple means for comparison, but these have been limited to a simple discussion about the space and time complexity of an approach, or an internal reflection on variations of a proposed method.

1.2 Motivation

With so many different methods and no historical knowledge or method of comparison this paper aims to provide a template for comparing existing models that should be extensible to account for the inevitable release of future methods. To that aim the following goals shall be met:

- Summary of the major existing spatio-temporal world modeling techniques.
- Collection of performance measurement or other comparison techniques as defined by the papers themselves.
- Introduction of meta-information in order to better compare the existing world modeling techniques
- A quick and easy to use table for high level overview and comparison of techniques
- Example application of the aforementioned information to select a fitting technique for a real-world application
- Subsequent evaluation and discussion on the appropriateness of technique selected, especially with respect to the introduced meta-information

It is with this collection of existing comparison techniques, new meta-information, and a tangible example that other projects may be able to more easily evaluate and select the best fitting spatio-temporal world modeling technique for the project. Additionally, when new spatio-temporal world model techniques are introduced, it should be with relative easy that their information be integrated into this method for comparative analysis for future use.

1.3 Problem Formulation

In order to best choose between preexisting solutions for spatio-temporal world modeling and guide future development it is vital comparative criteria be established. This comparative analysis will set out to clarify and quantify these approaches. Although the comparative analysis will be general enough to be applicable for any project wanting to incorporate spatio-temporal wold modeling, it will ultimately be viewed through the lens of a specific real-world application with a focus in long-term planning. More details about the specific application will be discussed later however, it is important to note that viewing the various modeling methods through the lens of real-world application, is quite powerful.

Improvements in world modeling, specifically within the domain of spatio-temporal world modeling, have already yielded significant effects on the performance of robotic logistic systems. These improvements directly translate into decreases in travel time as well as increases in reliability that hinges on knowing what areas to avoid a what times. These improvements in turn create a much more powerful and scaleable logistics network with less downtime. This ultimately leads to more goods being delivered which saves both time and money, and in the case of hospitals, possibly lives.

Despite all of these benefits, and the numerous number of different approaches for spatio-temporal world modeling, there currently lacks any method for accurately comparing and contrasting the different approaches. It is with this in mind that this thesis will collect, describe, compare, and contrast these approaches. It will use the preexisting criteria already available when possible as some, but not necessarily all, of the work includes basic performance statistics. Furthermore, in work where these criteria are not mentioned explicitly, or are not otherwise available, an attempt to derive the information either via calculation or collected via simulation. Lastly, new criteria will be devised or otherwise assigned to allow account for information desired and not provided or other meta-information that would aid in comparing these methods.

Finally, an example study will be included which will attempt to select the best-suited approach for a real-world scenario, known as ROPOD. The real-world scenario in question involves moving logistics internally within a hospital. It consists of a central server in charge of planning and routing multiple robots. Additionally, it will be assumed planning will be done with OpenStreetMap and thus will use a graph-based approach. More details and specifics about this project will be discussed in a later section.

TODO: perhaps an introduction or allusion to the experiments to come?

2

State of the Art

Given the range of the different methods for implementing a spatio-temporal world model, the methods have been divided into groups. Most spatio-temporal world models are implemented on top of preexisting world modeling techniques and thus the majority of implementations are tied to a specific spatial representation. There are, however, exceptions to this with some models being built from the ground up effectively intertwining the spatial and temporal components. On the opposite end of this spectrum, there exists currently at least one method that can be used in combination with a multitude of different world models.

2.1 Map Dependent Models

2.1.1 Occupancy Grids

Occupancy grids were introduced in 1985 by Moravec and Elfes. [3] In simple two dimensional terms, they can be thought of as a grid placed over an environment. Each cell then represents the probability or belief that that that cell is either occupied or free. Free in the simplest case meaning that a robot would be able to traverse through the cell. This concept can of course be extended into the third dimension for a more complex world model.

Temporal Occupancy Grids

One of the earliest and most straight forward attempts to introduce a temporal component to a world model were by extending existing world models, occupancy grids in particular. This can be seen in Temporal Occupancy Grids: a Method for Classifying the Spatio-Temporal Properties of the Environment. [1] In this paper Arbuckle et al introduce the concept of temporal occupancy grids (TOGs). The authors noted that the key to these TOGs were that they "can differentiate between different patterns of occupancy, even when the absolute probability of occupancy is the same." That is to say, one could imagine a parking lot where it would be possible with TOGs to distinguish between cells that are parking spaces, cells that are pathways, and cells that are not for driving at all, such as a median. These TOGs additionally made it possible to detect where a door or elevator may be.

Temporal Occupancy Grids were accomplished by generated multiple occupancy grids in the same fashion as was traditionally done but each occupancy grid would represent, and be generated using samples from, multiple different time scales. With multiple occupancy grids spanning multiple time scales, the probability of a cell being occupied could be computed by a simple summation.

Hidden Markov Models

Hidden Markov Models (HHMs), are a type of Markov Chain that can be considered "a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations." [12]. In more general terms, an HMM can be though of as having N number of states S, that are hidden, or otherwise not directly observable. Each state can have M observations made about properties of these states which may reflect indirectly, to varying degrees of certainty, the actual state. Furthermore, each one of these states has a given probability distribution of transitioning from one state to another. It is from this information that a Markov Model or Markov Chain can be constructed.

TODO: Add image?

In the specific case of occupancy grids, each cell can be thought of having two states, free, and occupied. It is not feasible to be able to directly observe every given cell at all times, and specifically at the time of path planning and thus there states can be thought of as hidden. However, through past observation and data collection, there is data know about a cell throughout time. Thus this temporal data can be thought of as the observational data and be used to make predictions about state transitions.

Early combinations of HMMs with occupancy grids differed from previous dynamic world modeling approaches as this approach "does not depend on dynamic object detection and high-level object models; it considers only the occupancy of the space at a lower level of abstraction"[11]. By relying on and collecting lower, more easily observable data, larger amounts of data could be collected and processed over greater periods of time. Since each cell was dependent only on previous observations of that cell throughout time, the increase in data quantity and the discrete nature of the predictions lent themselves would improve state predictions.

Meyer-Delius [11] also introduced the concept of online learning to this approach. Traditionally, offline learning had been used where a robots navigational system would hold copy of a world model produced a some time before operation. It has possible that from the time the map was generated to the time at which the robot was operating that objects in the robots environment may have changed. With the introduction of online learning, the robot would be able to observe these changes and factor them in to its navigational system. This was the first addition to attempt to avoid the static nature of the transition states of the HMM.

Further improvement to occupancy grids with HMMs came with the concept of modeling trajectories of objects in the environment[17]. This is an important improvement because the dynamic motion of objects in an environment, such as humans walking a hallway, could now be better modeled. This process was dubbed Input-Output HMM (IOHMM) due nature of how cells of the grid would communicate with one another. Each cell would not only look through it's own historical data but also be able to communicate with its neighbors. In effect, this could allow a cell in hallway to be able to predict occupancy based off of a nearby cell that is currently occupied.

2.1.2 Spatio-Temporal Hilbert Maps

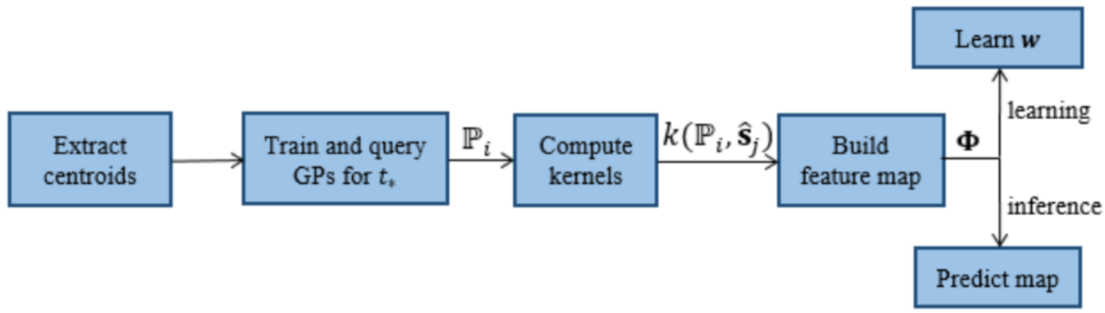

Figure 2.1: Spatio-temporal Hilbert map training process (GP - Gaussian Process) [15]

In contrast to the discrete nature of occupancy grids, Hilbert maps provide a continuous representation of an environment which allows for arbitrary world model resolution. They rely on "fast kernel approximations that project the data in a Hilbert space where a logistic regression classifier is learnt" A stochastic gradient optimization can then applied. This approach is similar to that of a Gaussian processes occupancy map but with a much lower computational cost computational cost computational cost computational cost. Having been introduced as recently as 2016, Hilbert maps, and the addition of a temporal component, are still a fairly new field of research but already some of the authors from the original paper have already begun to introduce a temporal component to this new form of world model. [13, 15]

In static Hilbert maps, the kernel can be thought of as the location of an obstacle or object. When introducing the temporal dimension, the centroid of a moving object is extracted from raw data over time. This data can the be used and trained on to create a model that can predict the direction and speed of an object at a given location at a given time. It is particularly well suited to short-term predictions such as car traffic on a road or at an intersection. [15] [16]

2.2 Map Independent Models

2.2.1 FreMen

Frequency Map Enhancement, or FreMEn as it became known, is a technique for spatio-temporal world modeling that can be used independent of mapping or world modeling technique. It was introduced by Tomáš Krajník, Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, an Tom Duckett in 2014. It's original goals focused on improving mapping for long-term scenarios. Additionally, it was notice that a large number of previous approaches had focused on mapping multiple static environments over time which worked well for environments that changed slowly, but were not necessarily well fit for highly dynamic environments. FreMEn was designed to counter these issues. [4] Although initially used with octomaps, three dimensional occupancy grids, it was later decoupled from this mapping technique allowing it to be an extremely diverse and flexible technique.

In it's original and most basic form, FreMEn assumes that an environment can be broken down into multiple independent components. It is then further assumed that these independent components will take one of two binary states. Examples of this include a door being open or shut, a room being occupied, or a cell in an occupancy grid being free or occupied. Each one of these states can not always be directly observed, and the tools e.g. sensors available to the robot my have noise and cannot be taken as one hundred percent ground truth. Thus, each one of these components has a certain probability assigned to it. This probability defines the likelihood of it being in a given state, e.g. a door open or closed. Finally, since these states can be observed multiple times over a given period of time, their probabilities can then be defined as functions dependent on time.

At the heart of FreMEn lies a well known mathematical tool commonly used for signal processing, the Fourier Transform. Since FreMEn focuses on long term observations of dynamic, often human, environments, it is assumed that harmonic patterns will develop over time. The Fourier Transform can then be applied to these long term observations to convert them into the spectral domain for storage. Furthermore, because the Fourier Transform is easily reversible, with the inverse Fourier Transform, one can easily convert between the stored observations in the spectral domain back to the time domain. This allows for predictions at any given time t. Not only is this extremely useful for future predictions, but this method can also be used to analyze the accuracy of the model by comparing previously observed data from the past to the models predictions. Using this historical accuracy one can then tune the order of the spectral model to obtain more accurate historical predictions with hopes of also having more accurate future predictions. More information on this process can be found in the original paper[4] and the follow up FreMEn paper [8].

Improvements and Additions

As ground breaking and as flexible as FreMEn is, due to the many assumptions made, it is not without its flaws. One major assumption made is that areas of observation can be observed not only frequently but periodically in the most strict sense. That is to say, it is not only important that a location be visited and observed, but that the observations follow a pattern of equally spaced and timed observations. This is due to the Fast Fourier Transform (FFT) technique that is used in the original papers. Thus, latter authors devised and implemented other methods of storing data and making predictions. This often involved phase shifting or modifying the amplitude of the observation as well as using a modified equation derived from the Fourier Transform instead of the standard FFT. More information can be found in the paper [14].

Another major limitation of FreMEn is its assumption that all observable behavior can be modeled with binary states. An attempt to solve this issue was to replace the Bernoulli distribution of FreMen with the specific hope of being able to better represent human patterns present in environment such as an office building or hospital. Specifically, they "extend the technique (FreMEn) by employing both

Poisson processes as the counting model to replace the binary states of FreMEn and a new way of selecting the most prominent frequency components of the Fourier spectrum."[7] This approach, however, does not come without its own set of assumptions. Since human activity is assumed, the authors also assume behavior would be best grouped by the work week. For example, data sampled for a two months would be broken into 8 week sections. This works extremely well for patterns that are persistent on a weekly basis but perhaps not as well for seasonal changes for example. At the very least, one must apply some critical thinking to how data should be group depending on the desired application.

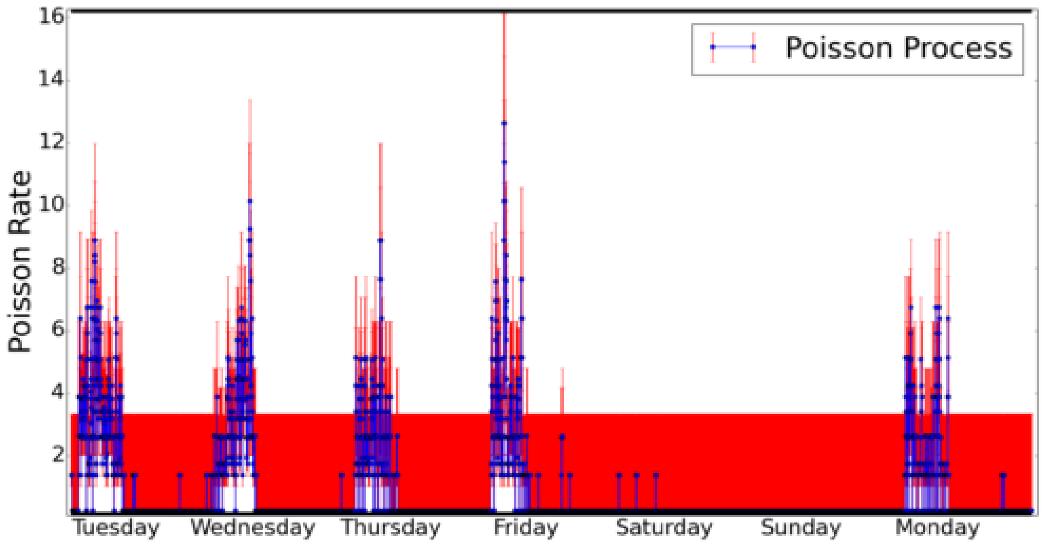

Figure 2.2: Lambda time series of a corridor using Poisson Process [7]

TODO: perhaps a section about the different FreMEn applications available e.g. FreMEn grids, FroctoMaps etc

2.3 Existing Methods for Evaluation or Comparison

2.3.1 Current Methods

2.3.2 Example 1

2.3.3 Example 2

2.3.4 Limitations and Areas of Oversight

3

Criteria for Comparison

How I am planning to compare/evaluate the various methods. TODO this is just some stubbed out notes about what I will cover here
item mostly inward looking. Very little outward looking analysis.

3.1 Criteria

- Space and time complexity (e.g. big O)
- Future Prediction Accuracy
- Historical Recreation Accuracy
- Efficiency of Stroage
- offline vs online learning
- feasibility of use with a multi-robot system
- meta-information
  - availability of work (e.g. libraries or source code)
  - implementation compelxity
  - suitable fields of application

3.2 How to use this criteria

TODO discussion and table goes here

4

ROPOD: A Case Study

4.1 What is ROPOD?

4.2 Evaluation Using Established Criteria

4.2.1 Proposed/Selected Method

4.2.2 Areas of Strength

4.2.3 Areas of Weakness

5

Experimental Setup

Having applied the theoretical knowledge derived in Section 3 to the ROPOD case study in Section 4 we have begun to narrow down the options for spatio-temporal world modeling in this particular case. However, a theoretical comparison will only suffice for so long. Given the focus ROPOD places on real world environments it is critical that some operational tests be preformed before method selection. Not only will these experiments serve as a guide for ROPOD, but they will also act as a template for the comparison of future spatio-temporal world modeling techniques.

5.1 Environmental Representation

Given the complexity and size of the target environment for ROPOD, a large hospital, it is necessary to pair down features of the building until only the core components remain. The three dynamic environmental components being target are doors, path planning with carts that are often strewn about the hallways and surrounding rooms, and elevators. Therefore, a model environment has been designed for the simulations to be run on that contains these three key components. The model environment that has been designed takes heavy influence from the actual environment but some notable have been made. The model has a decreased area to allow for faster model training and path planning. Additionally, extraneous rooms and hallways have been removed. A comparison between the actual hospital and the designed model can be seen below.

TODO add picture

5.2 Common Assumptions

In order to insure only the desired component is being tested at any given time a set of assumptions are made.

- All robot components are working correctly (no internal faults)
- Other than the object under test (e.g. doors), all other objects in the environment are static
- All information other than the objects under test are perfectly known
- Observations made/provided by the training data are assumed to be ground-truth

5.3 Commonalities in Approach

Although different components will be under test, each experiment will be run in a similar manor.

The experimental setup is as follows:

- Training data consisting of observations made every 15 minutes over a simulated month will be provided to the models
- All data generation will be done using the same program and the specifics of how the data is generated will be discussed further below in the relevant experiment section
- Using the same generation procedure, a new month will be generated
- The models will be trained using the original month and then tested against both the original month and the new test month
- In the case of the doors and elevator, only the objects themselves will be modeled and the comparison will only be how well the generated models can predict the ground thruth

- In the case of the more advanced hallway scenario, predictions about objects in the environment will be used to plan a path and this path will be compared with paths planned using the ground truth
- All of these paths will then be compared using the criteria described below.
- All experiments will be done on the same hardware
  - ASUS UX330UAK Laptop
  - i5-7200k 2.5GHz
  - 8GB DDR3
  - Arch Linux 4.19.4

5.4 Data Generation

Data generation is done using a combination of built in Python libraries and the NumPy library. Each object is broken down into a series of days which is then further broken down into a series of behaviors. Each behavior represents the likelihood of an object being in a given state between two times. Furthermore, each behavior has a starting state and an ending state, which if different, are swept through linearly. E.g. if a behavior is modeling a binary state of a door that starts at 100% open and then becomes 100% closed an hour later at the half hour mark the door would have the door being at a 50% likelihood of being open. Additionally, it is possible to specify the amount of Gaussian noise that is added on top of the behavior where the nominal state is used as mu and the sigma is used to specify the magnitude of the noise. Months of data are thus generated by walking through these behaviors in 15 minute increments for each day in the month where each month is assumed to have exactly 31 days.

For example, if a door was open from noon until midnight every day, and then closed from midnight until noon there would be two behaviors one from midnight until noon and one from noon until right before midnight. In this case, the starting and ending values of each behavior would be the same and no noise would be introduced. This would create a sharp, well defined change at exactly noon and midnight every day. Further details and specifics about data generation will be discussed in their respective sections.

5.5 Model Parameters

TODO expand?

Four models will be tested: Duckett's TODO, FreMEn, HyperTime, and Gaussian Mixture Models. HyperTime and FreMEn are of particular interest as they will be able to represent non-binary states. HyperTime will be using the expectation mean variant TODO why? The Gaussian Mixture Model, despite not being expected to preform as well as the others, has been included as a comparison with previous experiments like that done in TODO SITE. FreMEn and HyperTime will both use an order of three as that has, on average produced, the best results as seen in TODO SITE and improve Duckett info. The Duckett approach will contain three sub-models. The models will have a TODO wording .75, .25, and .05 TODO replacement rate? all with a size of 20. This is to closely match the approach in TODO PAPER. The three internal models will be averaged together to produce a prediction.

It is important to note that unlike the other three approaches, Duckett does not inherently support the ability to predict events arbitrarily in the future. It is limited to a single immediate prediction. Therefore, future predictions will done by simply projecting historical predictions forward on a monthly basis. That is to say, whatever Duckett's prediction was on the third of the first month it will be the same on the third of the next month.

The Duckett implementation was done using Python 3 while Gaussian Mixture Models, FreMEn, and HyperTime were all written in C++. The Duckett implementation was written by the author of this paper TODO phrasing? while the other three are available on Tom Krajnik's Github page. TODO link? Finally, the non-binary version of HyperTime, although not currently freely available on Github, can be obtained upon request as mentioned in the binary only repository.

5.6 Comparison Criteria

In order to evaluate the intricacies of the selected models a wide variety of data points have been selected for comparison. A focus has been placed on collecting data relevant to accuracy and scale-ability of the modeling technique given the eventual scope of the ROPOD project.

- Accuracy to Ground Truth
- Accuracy to Historical Recreations
- Planning Run-time
- Planning Memory Consumption

5.7 Doors as Dynamic Objects

5.7.1 Experimental Motivation

As is the case in many places of employment many areas of a building may not be accessible to the public, and by extension the robots, outside of work hours. This could come in the form of a given hallway between two areas being locked after 17:00 as the day works go home. In another case, it could be as simple as someone preferring to having a door shut to a hallway during a loud or chaotic time of the day.

Regardless of the reason, it is certain that the states of doors are often both dynamic exhibiting both periodic and long term changes. In the ideal case, a robot, much like humans, would learn when certain doors are closed and be able to plan accordingly. Making an accurate prediction can save time, but making an inaccurate predication can also be costly. An in accurate prediction would force a robot to not only backtrack, but also recalculate that path required to get to a target. Additionally, it may not be possible to make deliveries at all times. In the worse case, a robot may even manage to get itself locked in an environment unable to return back to it's base and eventually run out of power requiring human intervention. For these reasons and many more, the door experiment is an excellent example of the benefits of spatio-temporal world modeling.

5.7.2 Experimental Details

In order to keep this test as straight forward as possible, only the door object itself has been modeled. The doors could belong to a room where a package must delivered or perhaps a hallway that could be used as a shortcut. Each model will be directly, or indirectly be tasked with predicting the state of this door. Figure 5.1 displays an example of three doors in ward 24 of the hospital that may display the dynamic behavior previously mentioned.

When evaluating the predictions made by the models a simple 50% confidence value will be used. That is to say, if a model predicts that the door will be 50% or more likely to be open will result in the robot attempting to deliver the package.

It is clear that this 50% cutoff does not take into account the penalties of making a wrong prediction, but this particular experiment was designed to investigate the accuracy of the prediction. The hallway experiment discussed below will delve a little deeper into the ramifications of inaccurate predictions. That being said, how the information of the prediction is handled afterwards is undoubtedly valuable, but is outside the scope of this current research.

5.7.3 Data Generation

As mentioned above, both the training and the test data for all models was generated using the same program.

The specifics for each door are as follows.

Door A

Door A is models a door that is highly influenced by the daily schedule of a 9:00 to 17:00 job and has a high amount of periodic behavior with a small amount of noise. There are 5 behaviors which model a standard work day. The time from midnight

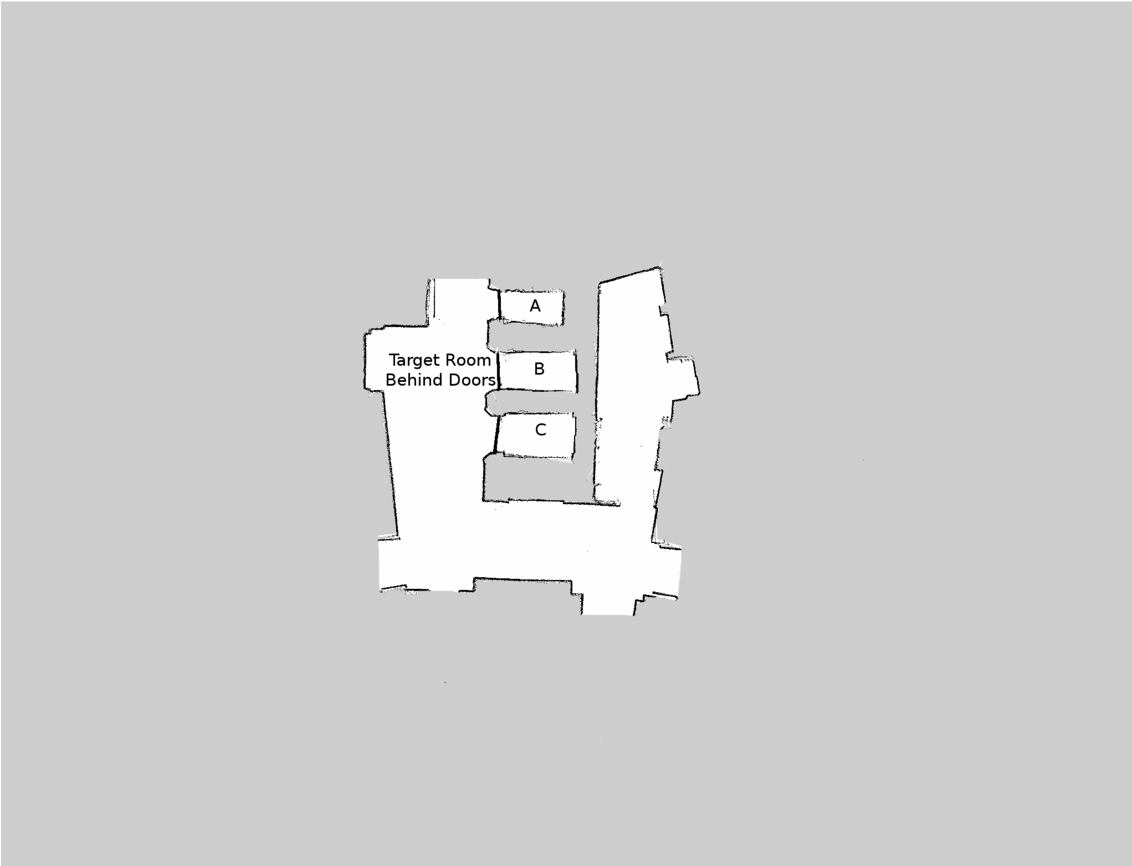

Figure 5.1: Multiple rooms behind doors in ward 24.

until work starts at 9 has the door in a normally closed state. From 9:00 until 12:00 the door is likely open. 12:00 until 13:00 is considered lunch time and thus the door is likely closed. Work resumes at 13:00 until 17:00 and thus likely open. From 17:00 until midnight the door remains in a likely closed state. Finally, the door is always closed on weekends with a 100% probability and no noise.

When the door is in a likely open state it has 70% chance of being open. Likewise, when in a likely closed state it has a 70% chance of being closed. Finally, when noise is introduced during the weekday it uses a standard deviation of 0.1 where mu is either 0.3 or 0.7 and 0.5 is the cutoff for being open or closed.

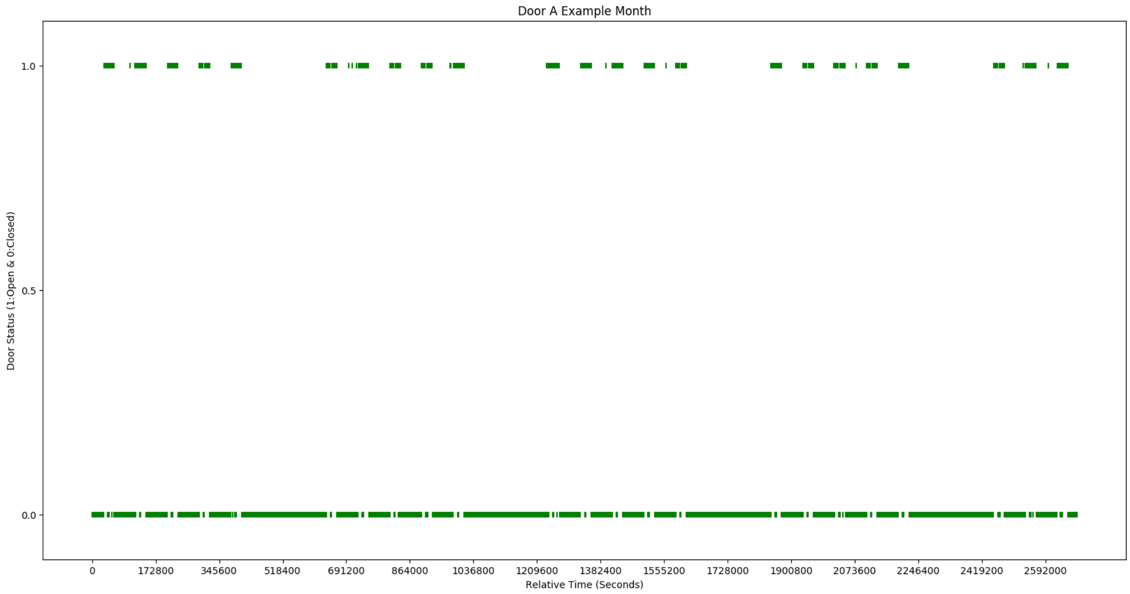

Figure 5.2: The training data for Month A

Door B

Door B, in a similar manor to door A, also tries to encapsulate the periodic behaviors of a work day but with a little more noise. Door B is always open from midnight until the start of the work day at 9:00. During the work day the door is a constant state of flux, being open and shut at random. This is modeled using a mu of 0.5 and a standard deviation of 0.1. This ends when the work day is over at 17:00 and the door goes back to remaining open until the next work day starts. To add additional periodic complexity, such as a weekly meeting or delivery, every third weekday the door will be closed the entire day. This trend is continued across month boundaries. This means if the doors was closed on the day before the last day of the first month it will again be closed on the second day of the next month. Finally, on weekends the door will always be open with 100% certainty and no noise.

Door C

Door C does not attempt to model any periodic behavior but instead test long-term change in an environment. A simple illustration of the long-term non-periodic change

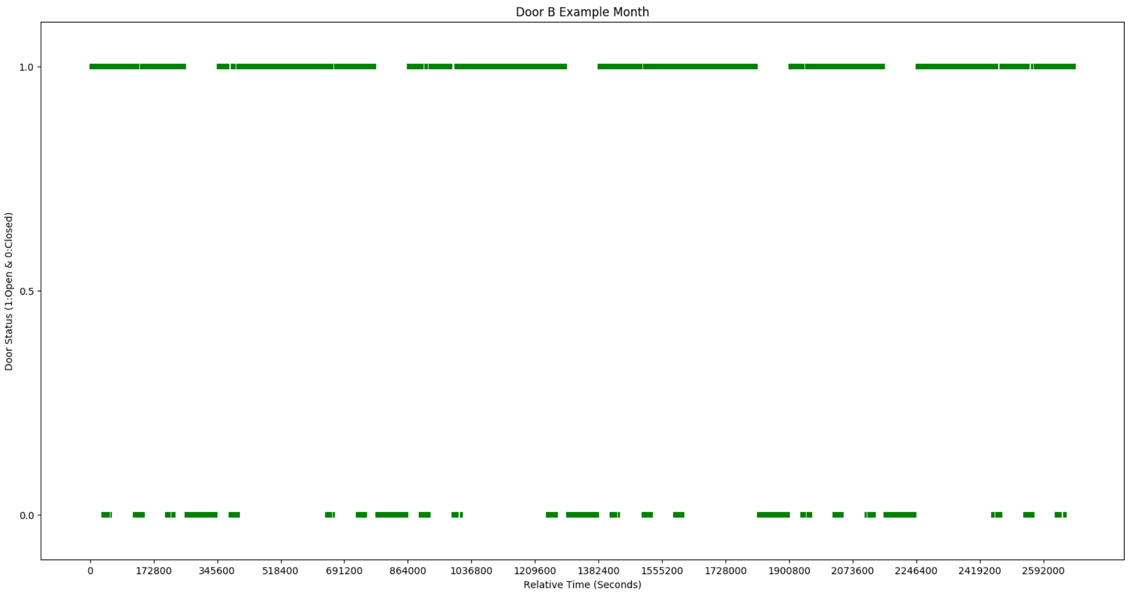

Figure 5.3: The training data for Month B

could be that of construction. The door has always been open in the past but it leads to a wing of the building that is now either being remodeled or has been removed. In the test data, this behavior is achieved by having the door be open for the first three weeks and then being closed for the rest of the training month and through the test month. In this model no noise was introduced.

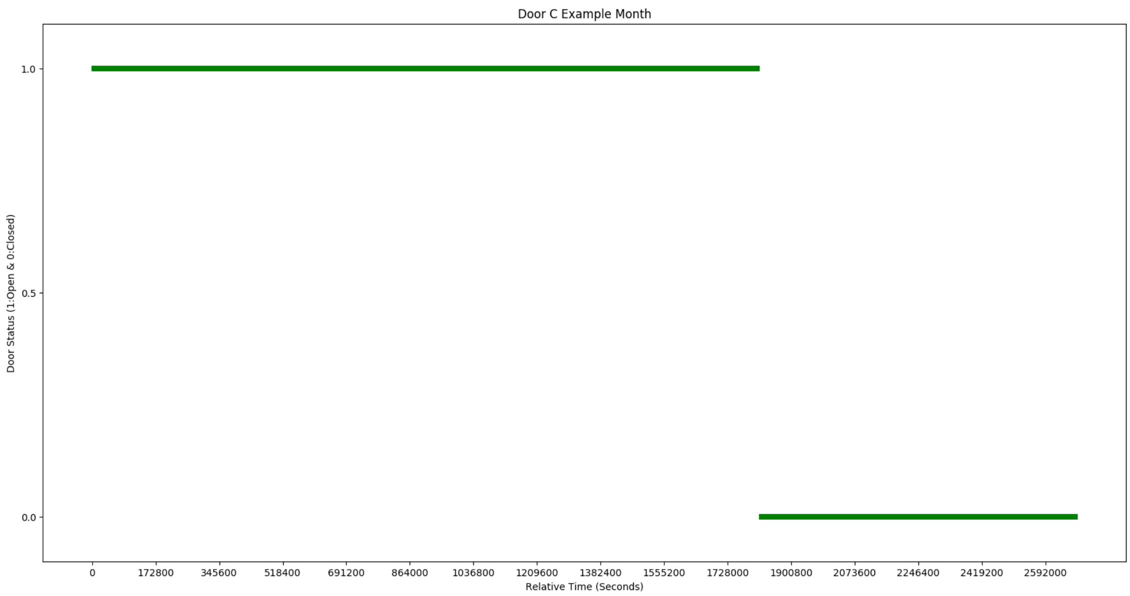

Figure 5.4: The training data for Month C

5.8 Path Planning in Congested Hallways

5.8.1 Experimental Motivation

Individual object representation and performance is vital for initial evaluation of an algorithm, but more expansive test must be completed to build a complete picture of an algorithms real-world performance. It is for this reason multiple dynamic objects will be used to model an environment that closely replicates that of ROPODs target environment. Not only does this serve as an excellent test case for ROPOD but it also serves as an initial evaluation into the scalability of a method.

When planning paths with multiple dynamic objects it is critical that not just few, but all of the objects are correctly predicted. If any of the objects state are predicted incorrectly a robot may need to reroute dynamically or in the worst case it may not be able to arrive at its destination at all wasting time and resources.

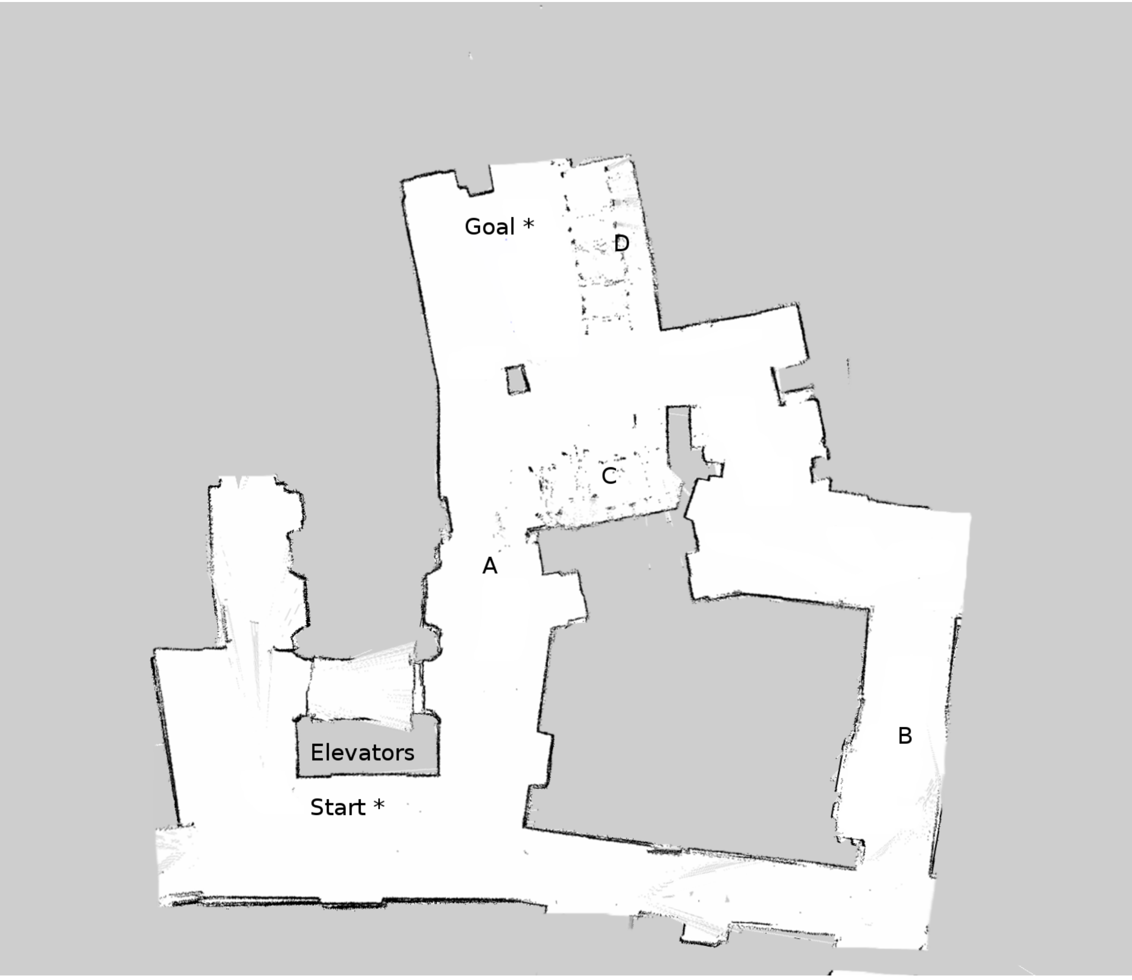

Figure 5.5: The path from the elevator to the storage area is often congested with multiple dynamic obstacles.

5.8.2 Experimental Details

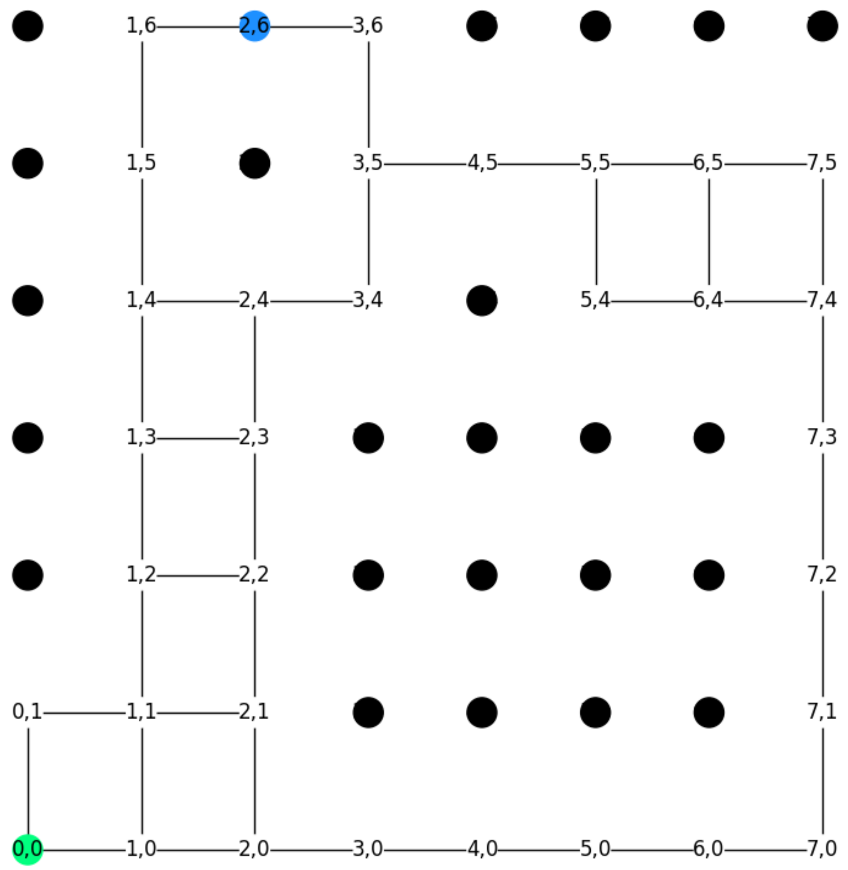

Figure 5.6: Graph representation of the hospital basement with no obstacles

The specific area under test in this experiment is a representation of the basement area of the hospital ROPOD will be deployed in. Figure 5.5 is a slightly cleaned up version of ROPOD generated map. Areas have been marked to highlight where common dynamic obstacles occur. Figure 5.6 shows a graph representation of this same area, simplified for path planning. Figure 5.7 shows the same area but with all dynamic obstacles present. The obstacles and there frequencies are as follows:

- Meals - Three times a day
- Laundry - Once per day
- Delivery - Twice per week
- Trash - Once every three days

The edges in the graphs are the possible routes the robot is able to take. Node 0,0 represents the start state which is roughly where the robot would be after taking the elevator down. Node 2,6 is the goal state and represents an entrance into either a storage area or another hallway.

5.8.3 Data Generation

The data for these test was generated using the same program as the doors and in the same manor. The graph generation and path planning was done using NetworkX a free and open source tool that allows for the creation and subsequent path planning on graphs amongst other functionality. More information can be found on there website at https://networkx.github.io/

The specifics for each door are as follows. TODO perhaps a graph that shows all at once for a week? TODO where do I clarify that this is only for example and not real data again? TODO standardize order

Meals - Nodes 1,2 & 1,2

Meals are a periodic and consistent behavior in the hospital. They happen three times a day regardless of which day of the week it is. Meals and their respective dishes will be staged in the hallway before being moved to another area or delivered. For ease of simulation, carts begin to pile up at node 1,2 at 4:00 and continue to build up until overflowing into node 1,3 at 6:00. These two nodes continue to be

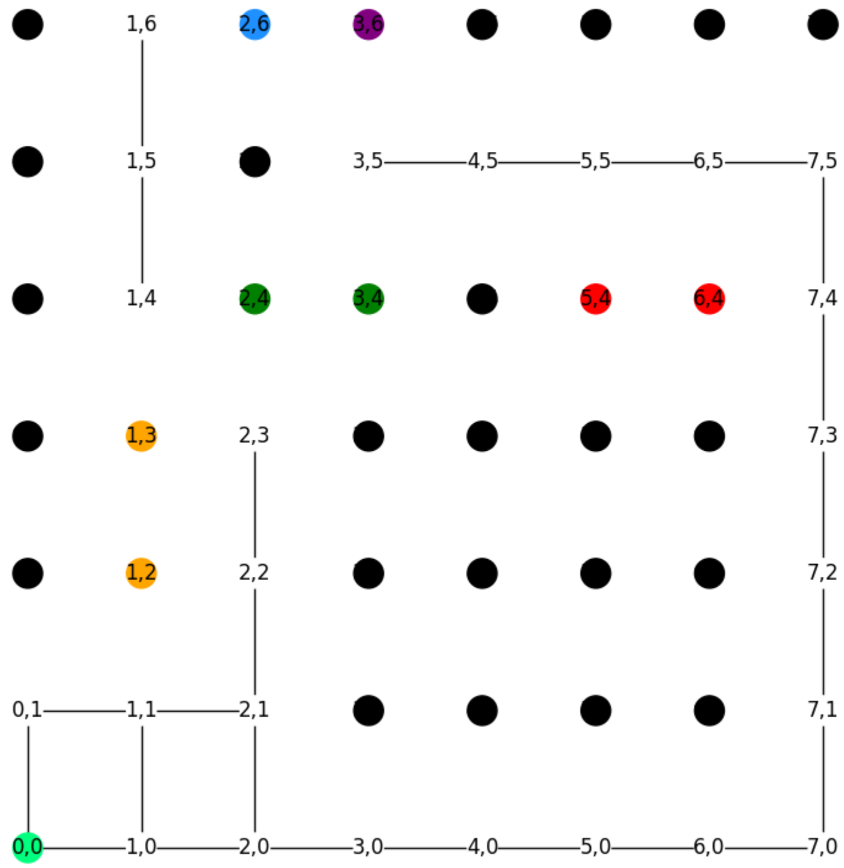

Figure 5.7: Graph representation of the hospital basement with all obstacles modeled

occupied until 8:00 when all carts are sent up for delivery once again making them free for traversal. The process repeats again starting at 12:00 and 16:00 using the same time steps.

Laundry - Node 3,6

Laundry, similar to meals, is assumed to build up periodically and daily causing node 3,6 to be blocked. Laundry is said to begin to build up slightly before midnight and begin causing a blockage at 00:00 until 12:00 at which point enough laundry has been processed that the node is again traversable.

Deliveries - Node 5,4 & 6,4

Deliveries are assumed to happen twice a week, once on Monday and then again on Friday. For the sake of simulation, deliveries block nodes 5,4 and 6,4 the entire day of the delivery, i.e. from 00:00 until 23:59.

Trash - Node 2,4 & 3,4

Trash or other objects will periodically be temporally stored in the hallway. For the sake of simulation, trash is said to begin to build up at node 3,4 and then expand into node 2,4. The first day of the first month, both nodes will be free for traversal. After the first day, at precisely 00:01 the first node is said to be blocked. After another day, the trash has grown large enough to block the second node. This cyclic behavior continues to happen when the second test month is generated. In summary, the first day both nodes are free, the second day the first node is occupied, the third both are occupied, and then on the forth day they are back to both being free for traversal.

Areas of Particular Note

As visible in Figure 5.7 it is not always possible to reach the goal from the starting node. Additionally, certain obstacles also block paths that might normally be considered first. An example of this is the obstacles caused by the meal nodes which prevent to most direct path to the goal upon leaving the elevator. Choke points are also a concern. These happen when an obstacle or obstacles completely cut of a given path. Examples of this are when both laundry and the first trash node are present or when both trash nodes and both meal nodes are present.

6

Experimental Results

6.1 Notes on Analysis

TODO: add this section note about common things like how Duckett Historical is actually live and future is using old data mapped to new notes about cyclic and decay artifacts in accuracy over time

6.2 Doored Areas

6.2.1 Door A

Given the periodic, human, and slightly random nature of Door A, it serves as an excellent starting point and litmus test of for various spatio-temporal world modeling techniques. Door A's results establish a trend that can be seen throughout the rest of the results. This is particularly visible in figure 6.3 where the models accuracy are evaluated over time.

Do to the periodic nature of the data, the techniques based off of extracting periodicities (i.e. FreMEn and HyperTime) do an excellent job of recreating passed events as well as predicting the future. HyperTime's slightly better prediction is most likely due to the extra steps taken to wrap the data back in on itself temporally. This additional training and accuracy comes at a great cost however. HyperTime manages only a small performance increase over around one percent over FreMEn with a total

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	84.11%	77.80%	92.51%	97.66%
Prediction Accuracy	77.15%	79.43%	87.08%	88.87%
Computation Time (Milliseconds)	610	50	70	5630
Memory Usage (KB)	31036	34968	34656	37192

Table 6.1: Door A Data Overview

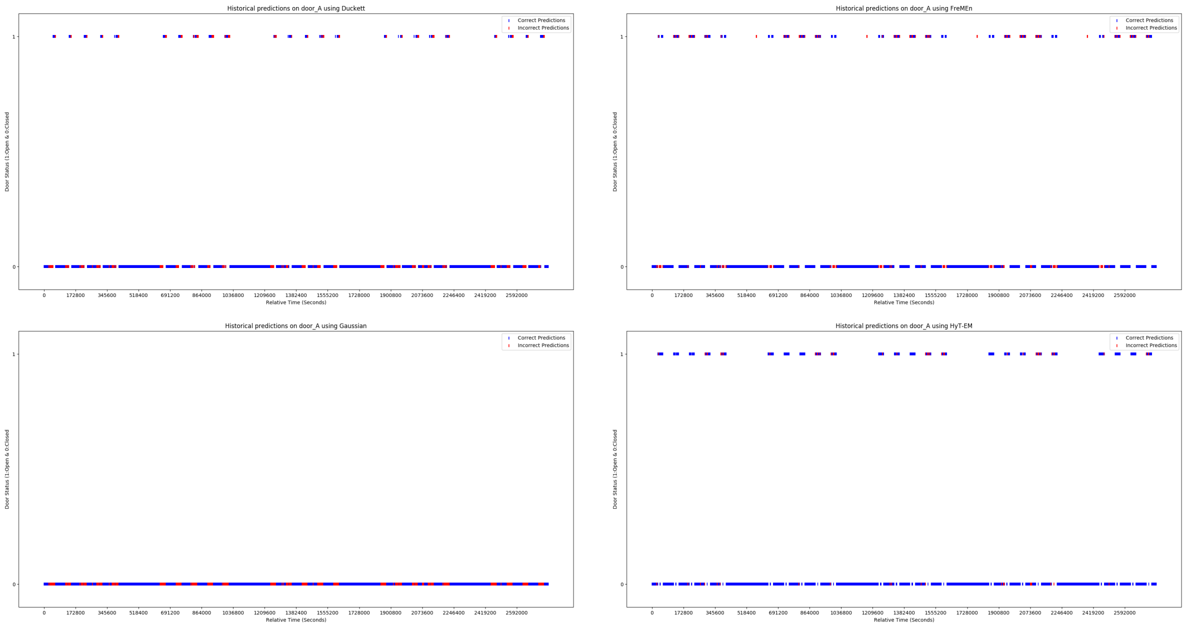

Figure 6.1: Historical Recreations - Door A

training and prediction time taking on the order of two magnitudes longer of both Gaussian and FreMEn. Finally, it appears that the Gaussian model completely failed to predict any open state. This is thought to be due to the sparseness of positive, or open states resulting in the model oscillating between 0 and just under 0.5 thus never reaching the open threshold.

6.2.2 Door B

Door B demonstrates a fact the will be come increasingly clear with future experiments. Methods that use a modified version of the Fourier transform as described in TODO: CITATION require a certain threshold of frequency to be met

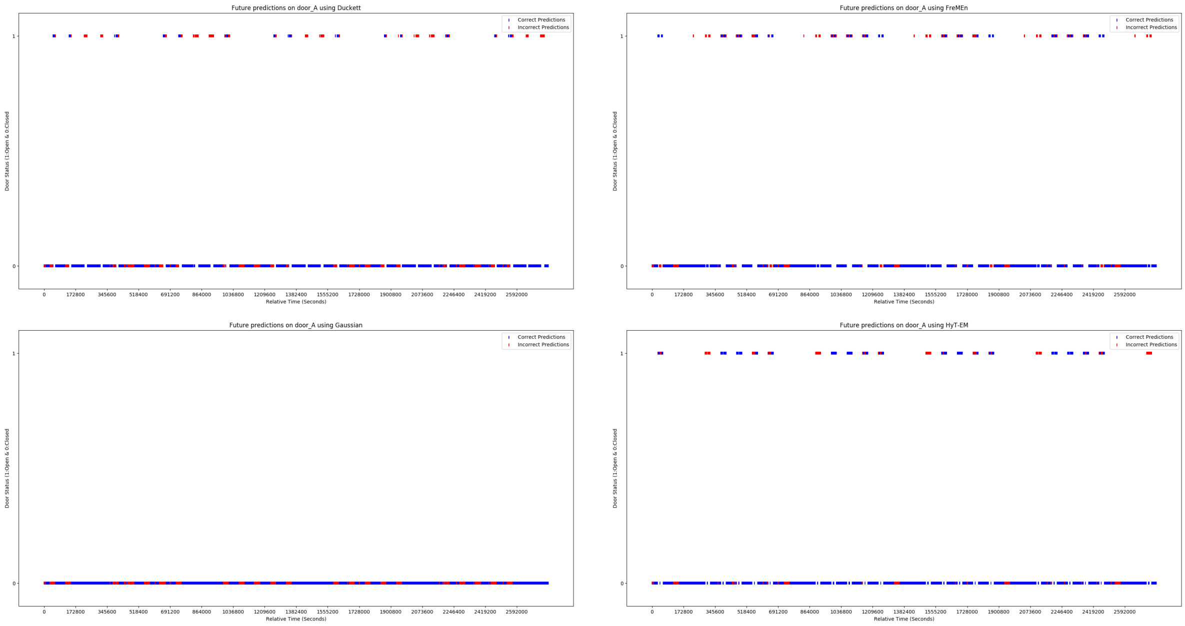

Figure 6.2: Future Predictions - Door A

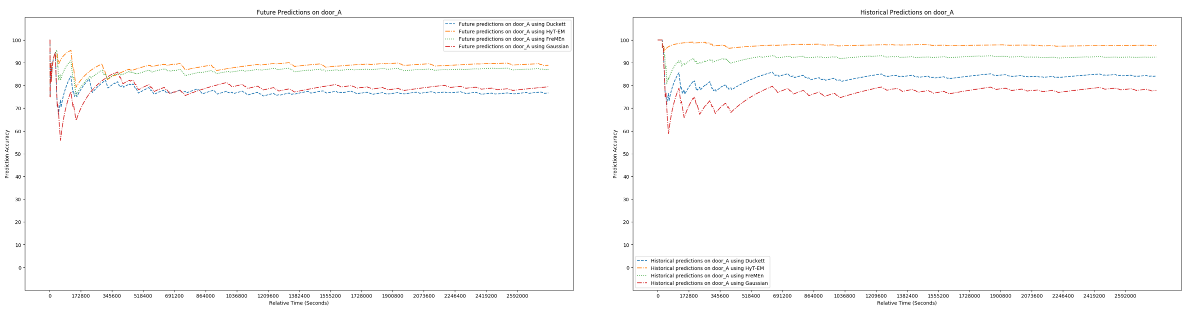

Figure 6.3: Model Accuracy Over Time - Door A

in order to accurately predict. In fact, it's interesting to note that the very thing that allows these methods perform so well with periodic behavior causes issues with datasets with non periodic behavior or datasets with minimal periods of periodic data.

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	85.71%	59.81%	75.20%	71.55%
Prediction Accuracy	69.24%	62.17%	76.95%	75.78%
Computation Time (Milliseconds)	600	60	80	1440
Memory Usage (KB)	31036	34644	34892	37692

Table 6.2: Door B Data Overview

Duckett, relying almost solely on averages, does surprisingly well on this experiment beating out all other models in both historical and prediction accuracy. It is, however, important to note the flaws in Duckett's long-term prediction. Due to the fact that future predictions are not directly possible using Duckett and thus previous month data is used, it is clear to see that while Duckett performed overall well historically, it is not without problems in future prediction. Of particular interest is its failure to continue to accurately predict a periodic behavior that does not occur on month boundaries. Since the behavior that happens every three days does not happen on the same day between the two months Duckett incorrectly predicts its occurrence as happening on the same days as last month.

In terms of resource usage, a similar trend to door A is observed. Gaussian and FreMEn predictions take under 100 milliseconds while Duckett takes around an order of magnitude longer, and HyperTime yet another order of magnitude longer. Finally, similar to door A, all approaches use about the same amount of memory for training and predicting within about 10 megabytes.

6.2.3 Door C

With door A exemplifying the occasionally periodic and somewhat noisy behaviour in the real world, door C serves as almost the exact opposite, modeling a one time, long-term change. Perhaps some what expectedly, the results in terms of prediction accuracy are almost completely opposite that of door A with Duckett

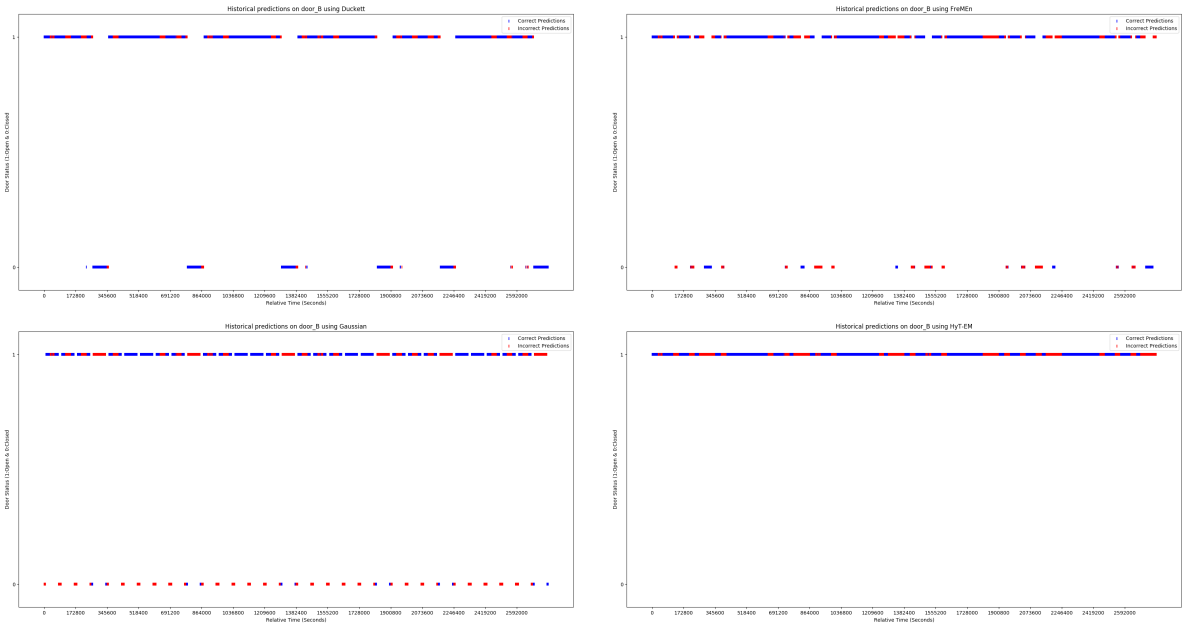

Figure 6.4: Historical Recreations - Door B

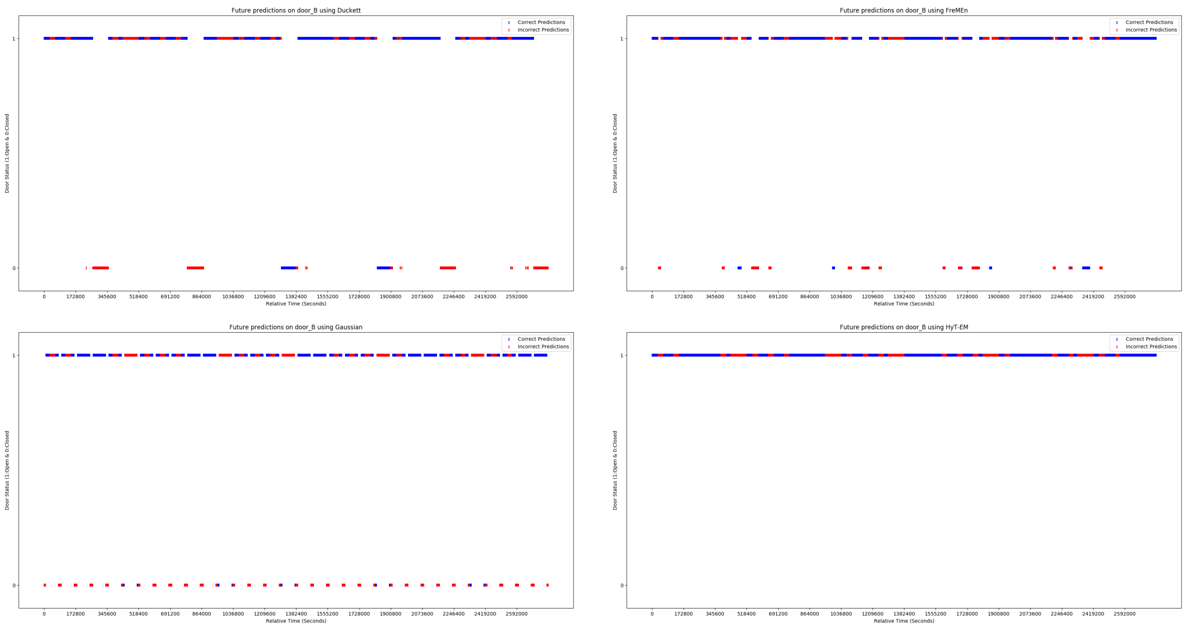

Figure 6.5: Future Predictions - Door B

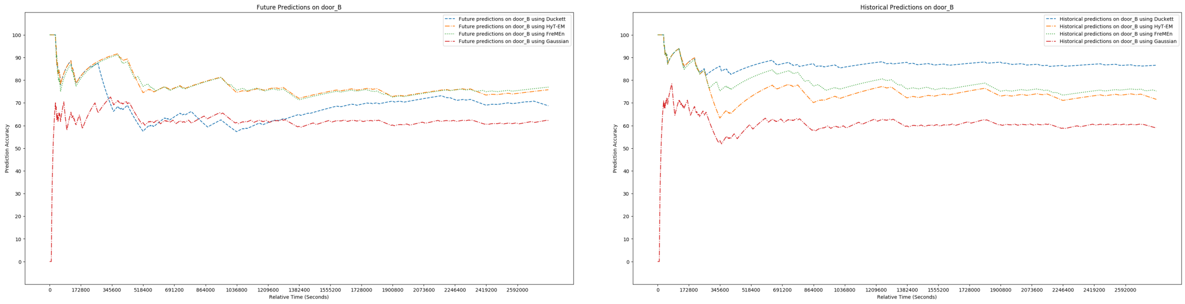

Figure 6.6: Model Accuracy Over Time - Door B

having the best performance both historically and with future predictions.

Unfortunately, despite its better performance by the numbers in table 6.3, looking at the resulting graphs in table 6.7 and 6.8 shows a somewhat initially disappointing result. As discussed in the door B experiment, Duckett, merely uses it's historically prediction again for future predictions. This is clearly visible by the prediction of the door being open for the first three weeks in the future predictions. However, Duckett was not designed for long-term future predictions and instead meant to be used "live". When this is accounted for, Duckett's performance is once again impressive. In fact, as discussed TODO: where will this be discussed? the historical predictions made by Duckett can be view as it's "live" predictions and thus it would be expected that in the real world it would continue to predict the door as being closed as long as long-term future predictions are not requested. This means that it is likely Duckett's performance would be closer to 100% for future actives for as long as the door remained shut.

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	99.56%	62.75%	67.74%	67.74%
Prediction Accuracy	31.82%	14.58%	00.00%	00.00%
Computation Time (Milliseconds)	570	60	70	530
Memory Usage (KB)	31224	35004	34976	37208

Table 6.3: Door C Data Overview

As alluded to above in door B's experiment, the lack of periodic data has caused

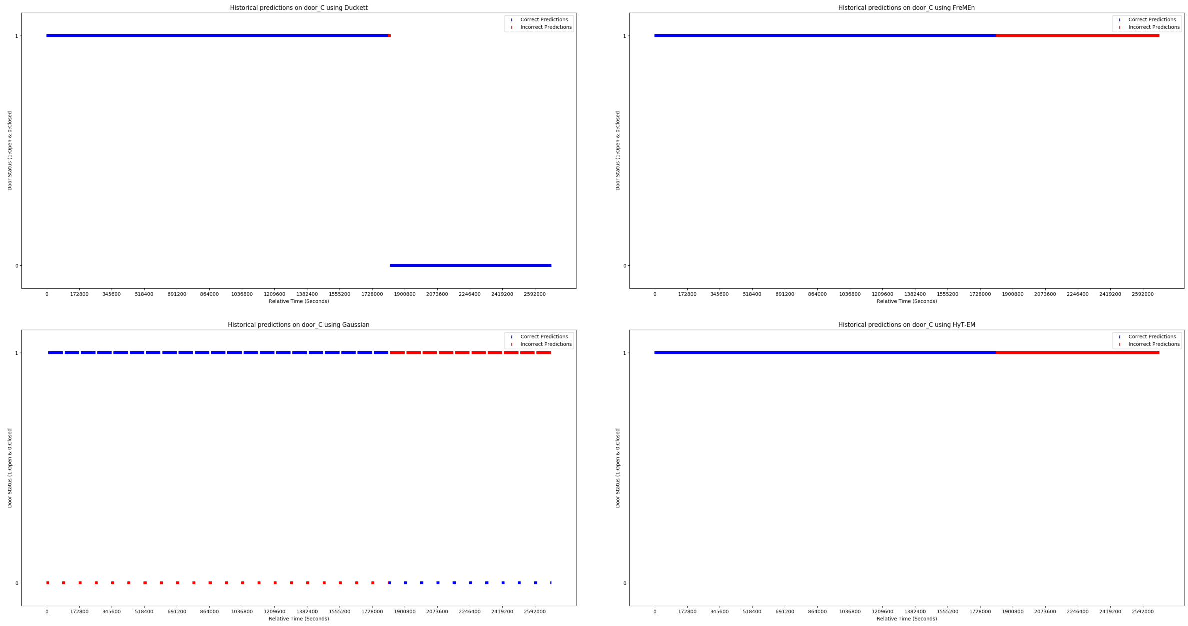

Figure 6.7: Historical Recreations - Door C

both FreMEn and HyperTime to take the easiest prediction for the training data and predict that the door will always be open. Unfortunately, this causes the future predictions to be entirely inaccurate. The only possible redemption for the Fourier based approaches on this long term changes is that the prediction would eventually flip to always being closed, but only after the total number of observations of the door being closed surpassed that of the door being open. In this case, that would take a total of just over 6 weeks if training was being done every night.

The resource usage statistics in figure 6.3 do interestingly break slightly from the norm. It appears that due to the simplistic predictions of the Fourier approaches, HyperTime has shaved off an order of magnitude in computation time. This is believed to be because the prediction model created by HyperTime, after the initial naive approach, has a larger error than the first and thus it immediately quits and stops attempting to improve the model. This is irrelevant however, seeing as the predictions are completely inaccurate so any gain in computational time is meaningless.

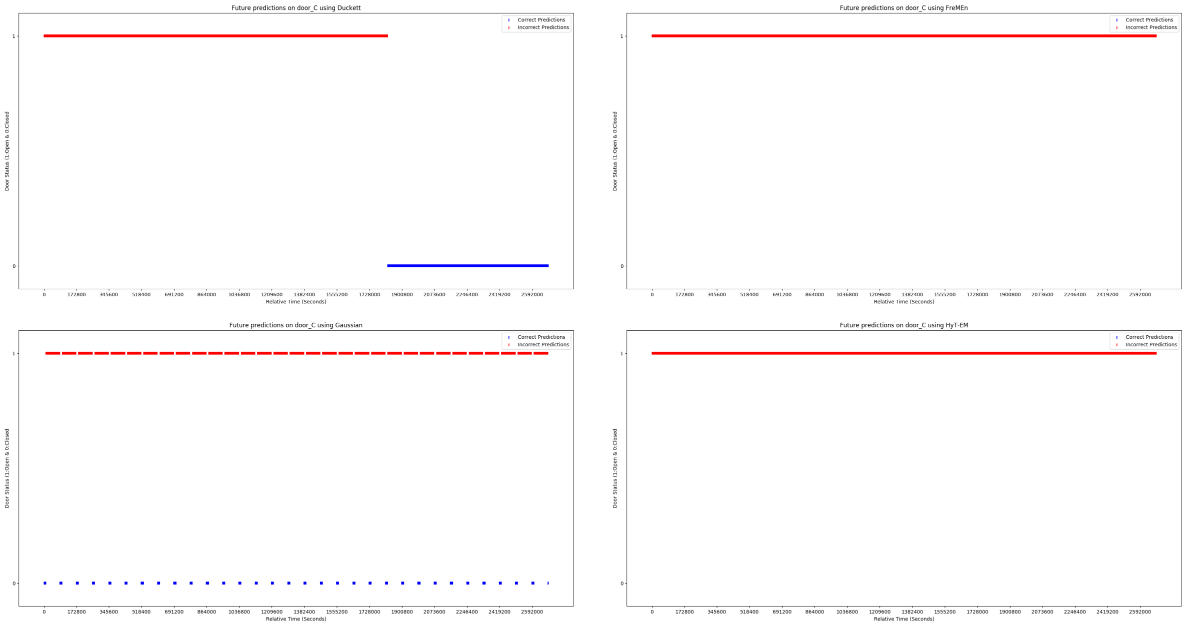

Figure 6.8: Future Predictions - Door C

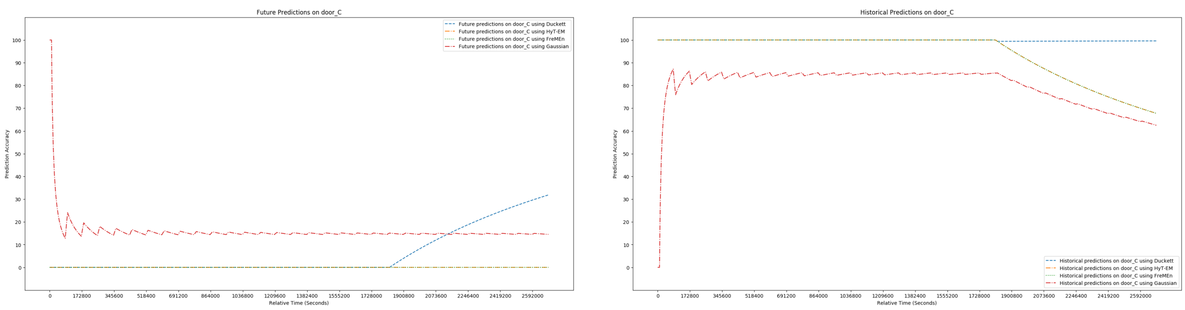

Figure 6.9: Model Accuracy Over Time - Door C

6.2.4 Final Thoughts

In terms of memory usage, all four methods appear to be relatively similar, which is to be expected.

6.3 Congested Hallways

TODO pair down images and move them to the end of the paper

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	91.13%	61.49%	64.52%	64.52%
Prediction Accuracy	32.26%	64.05%	63.21%	67.74%
Computation Time (Milliseconds)	600	60	70	1140
Memory Usage (KB)	31120	35364	35552	37192

Table 6.4: Hallway Trash Section 0

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	90.96%	61.09%	67.74%	67.74%
Prediction Accuracy	35.79%	61.09%	67.74%	67.74%
Computation Time (Milliseconds)	610	60	70	2120
Memory Usage (KB)	31116	35400	34928	37520

Table 6.5: Hallway Trash Section 1

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	92.31%	55.24%	96.17%	99.83%
Prediction Accuracy	68.92%	55.24%	95.97%	99.87%
Computation Time (Milliseconds)	610	60	90	6790
Memory Usage (KB)	31032	35660	35520	38372

Table 6.6: Hallway Delivery Section

6.4 Busy Elevators

Duckett good at

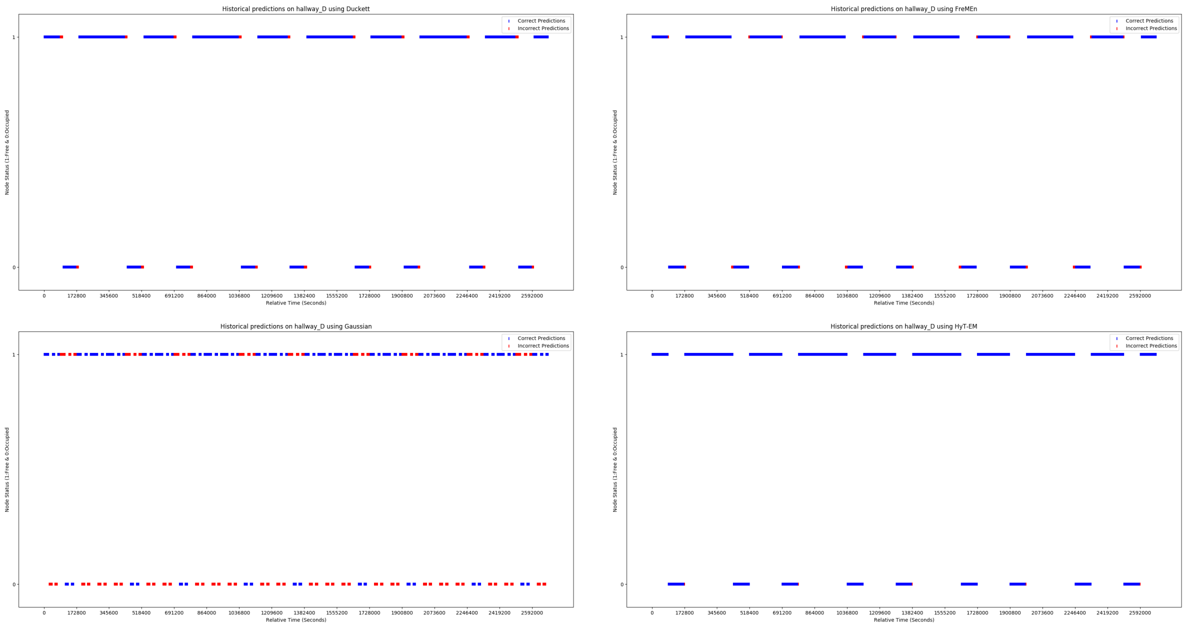

Figure 6.10: Historical Recreations - Hallway Delivery

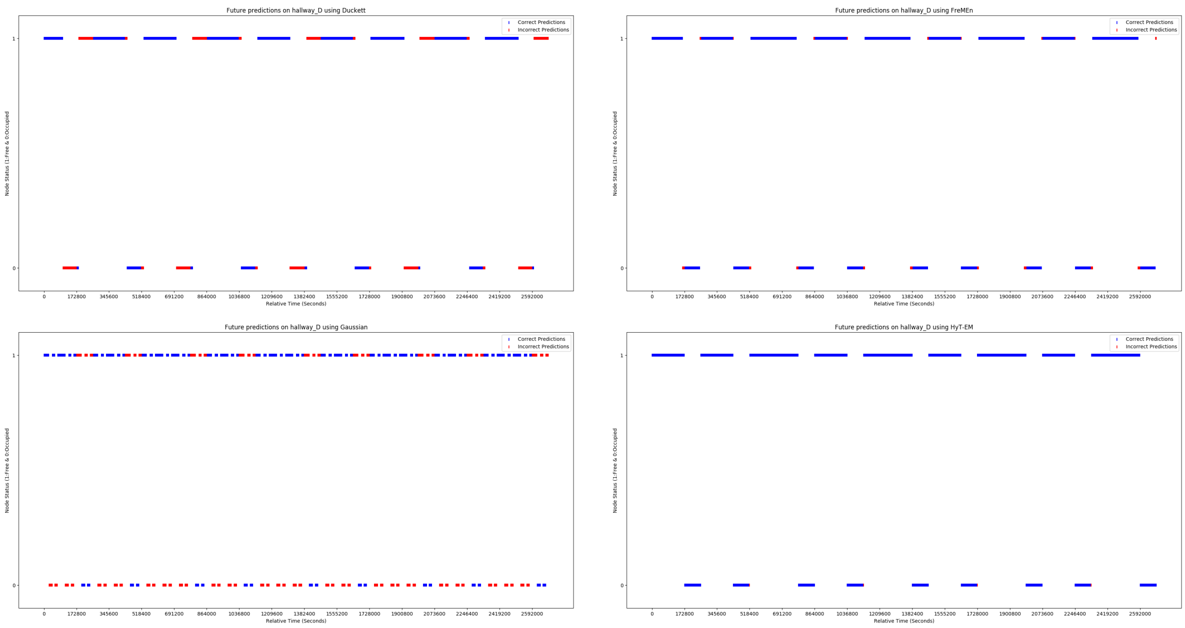

Figure 6.11: Future Predictions - Hallway Delivery

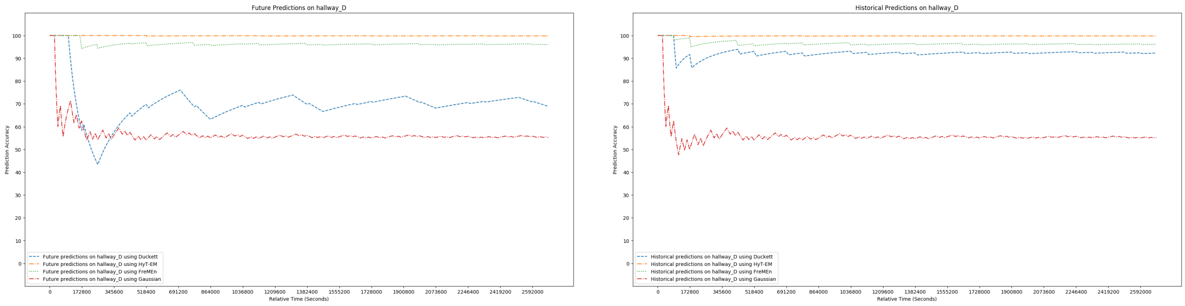

Figure 6.12: Model Accuracy Over Time - Hallway Delivery

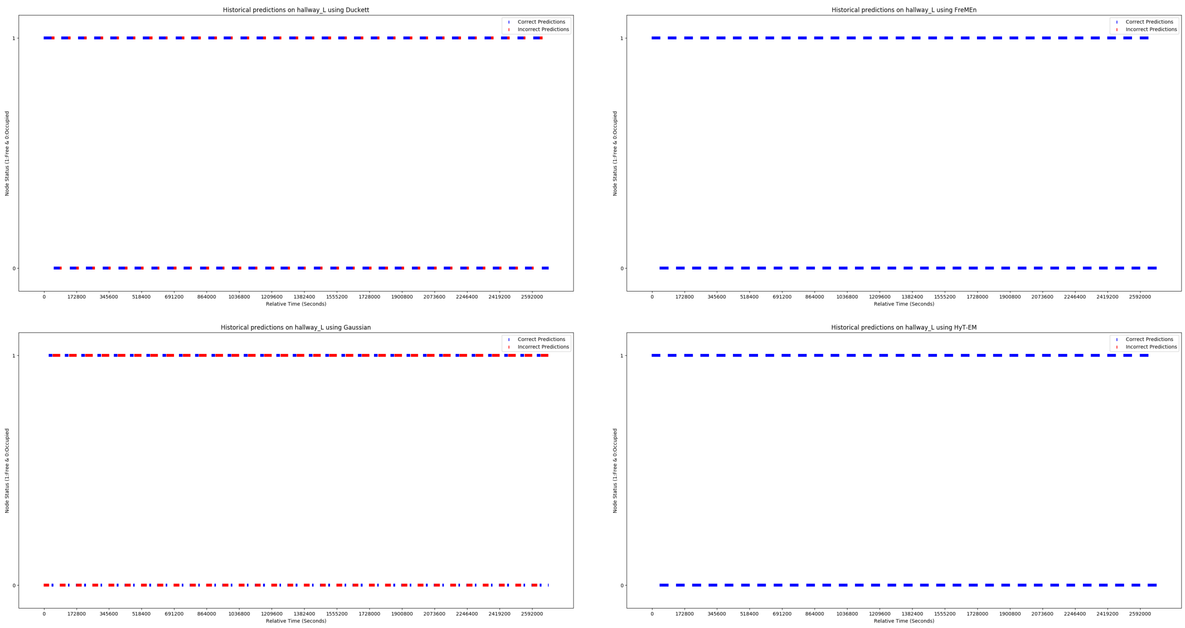

Figure 6.13: Historical Recreations - Hallway Delivery

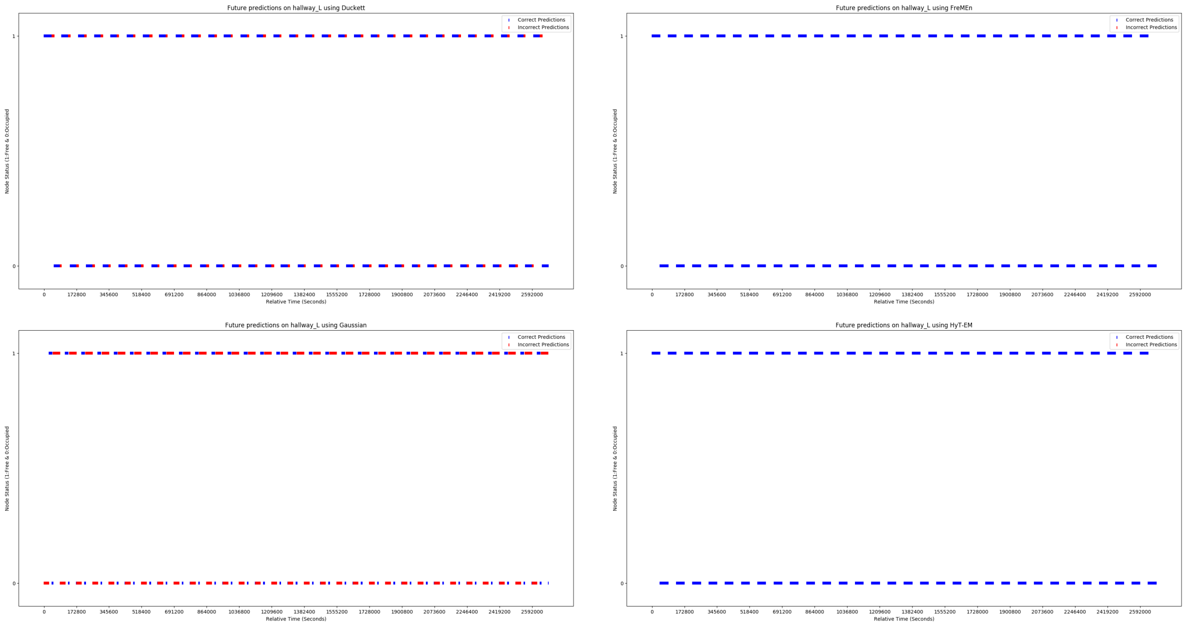

Figure 6.14: Future Predictions - Hallway Laundry

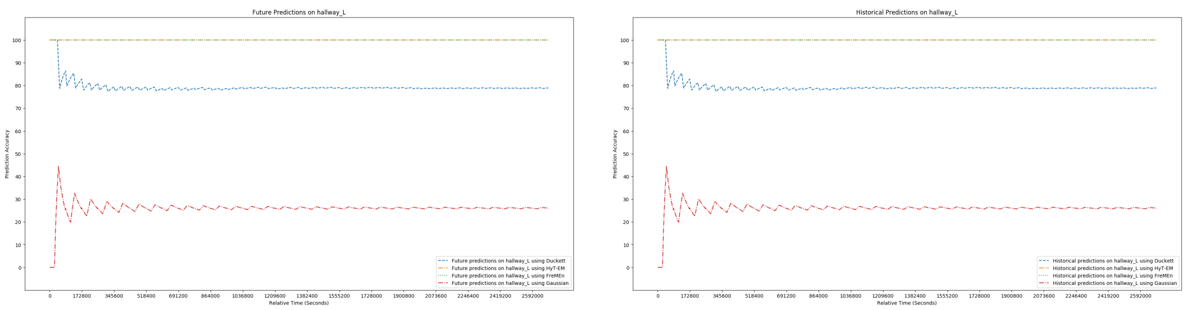

Figure 6.15: Model Accuracy Over Time - Hallway Laundry

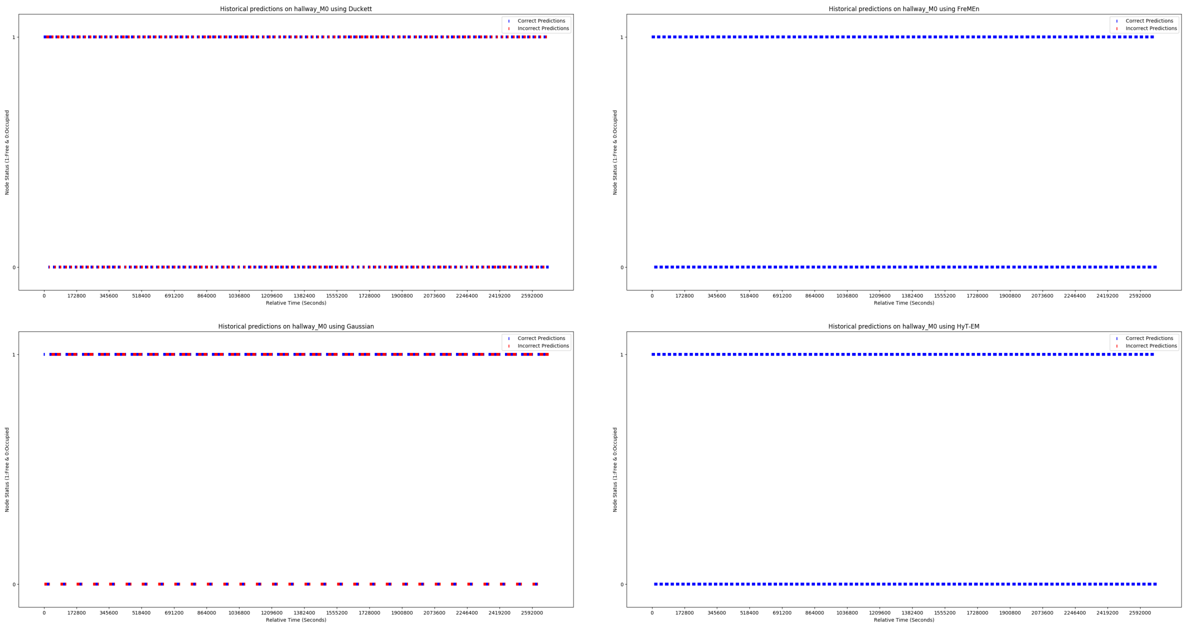

Figure 6.16: Historical Recreations - Hallway Meal Section 0

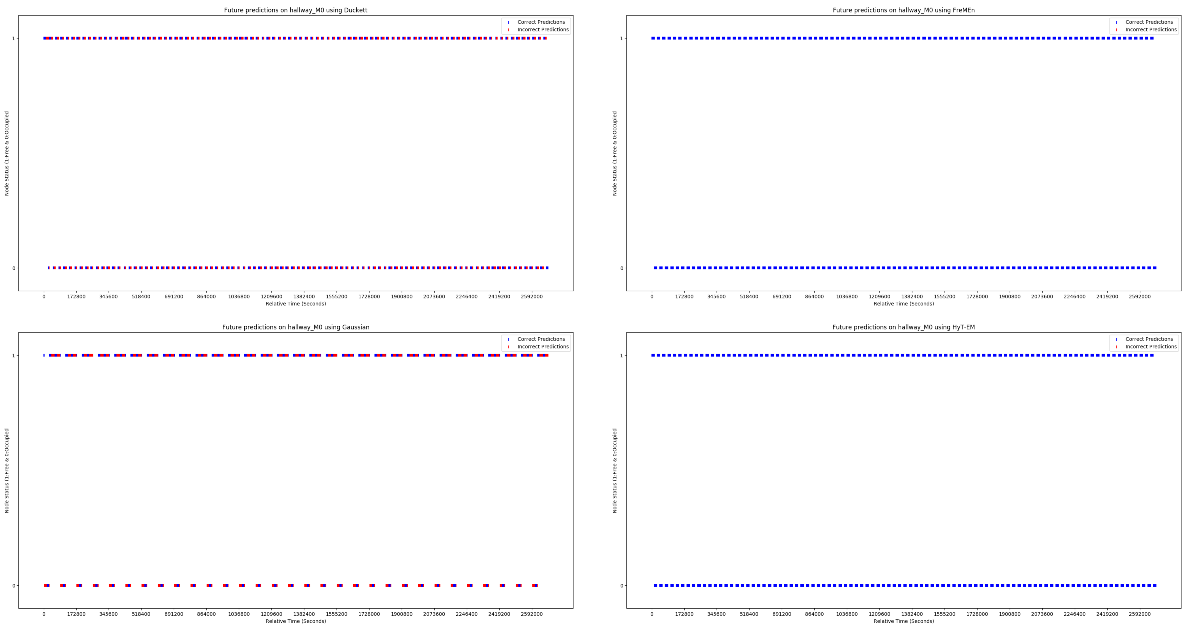

Figure 6.17: Future Predictions - Hallway Meal Section 0

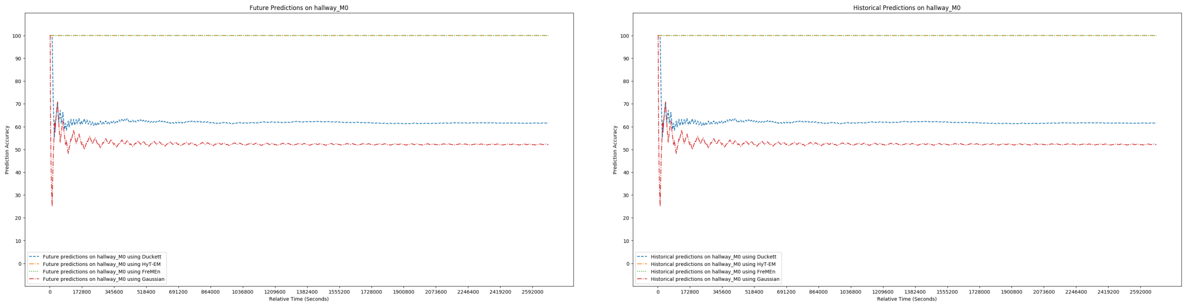

Figure 6.18: Model Accuracy Over Time - Hallway Meal Section 0

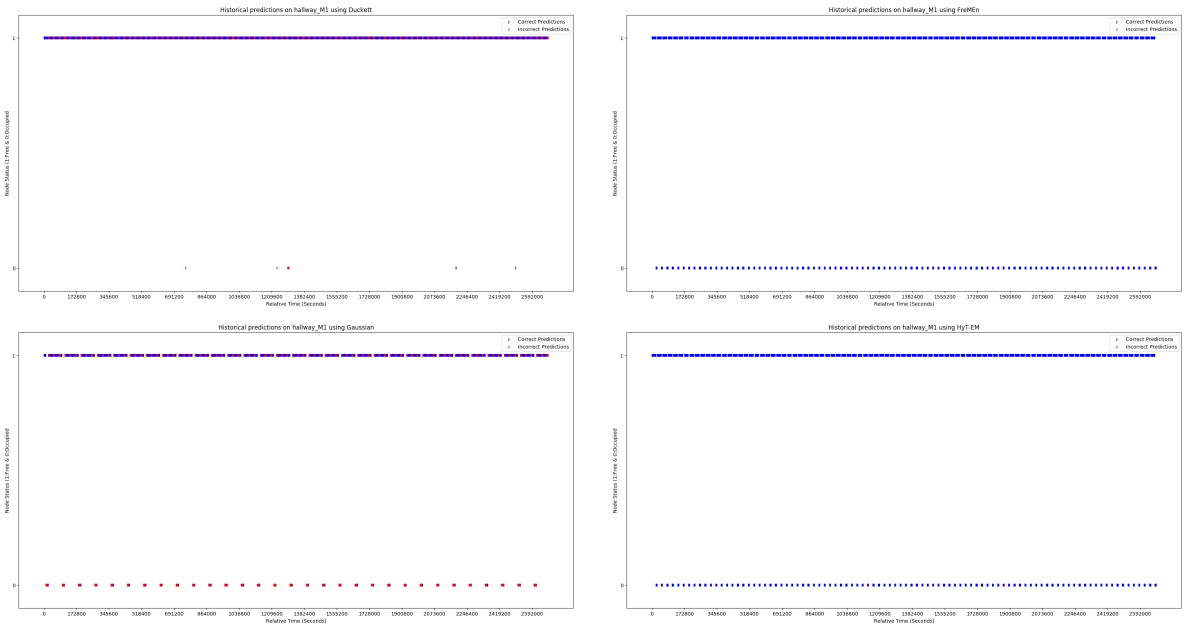

Figure 6.19: Historical Recreations - Hallway Meal Section 1

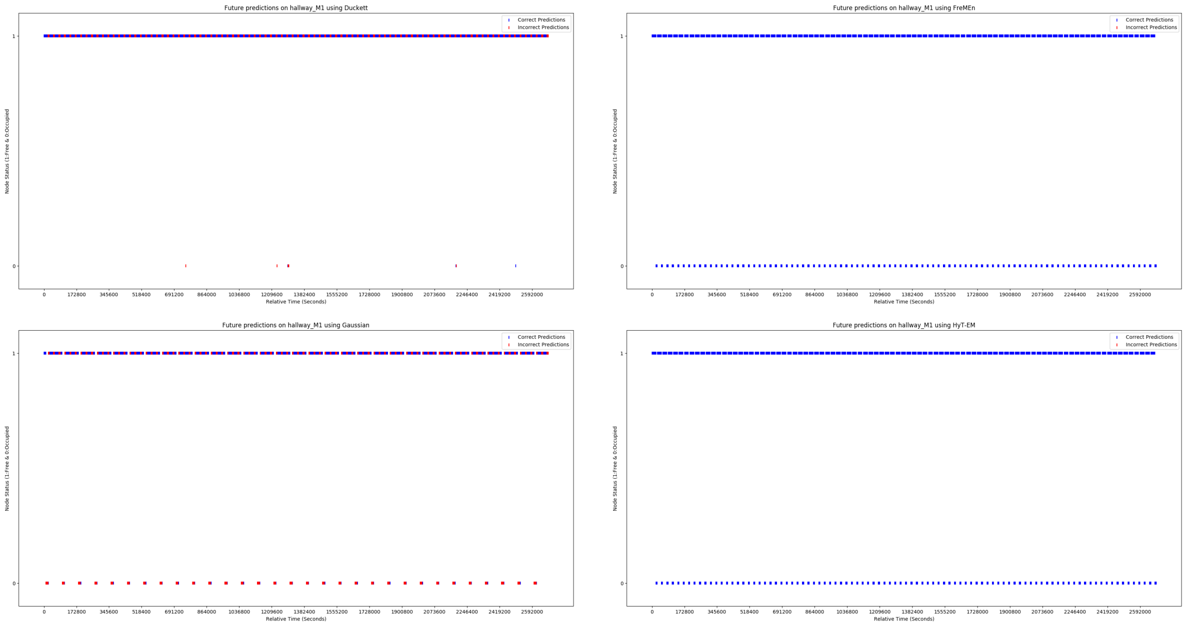

Figure 6.20: Future Predictions - Hallway Meal Section 1

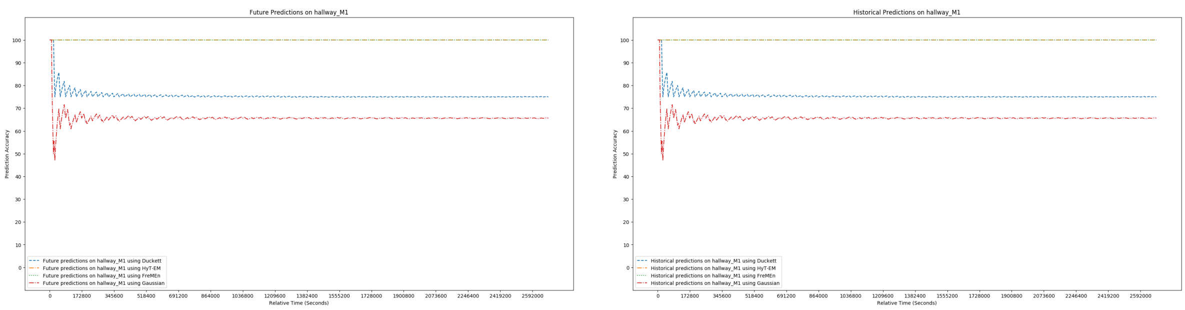

Figure 6.21: Model Accuracy Over Time - Hallway Meal Section 1

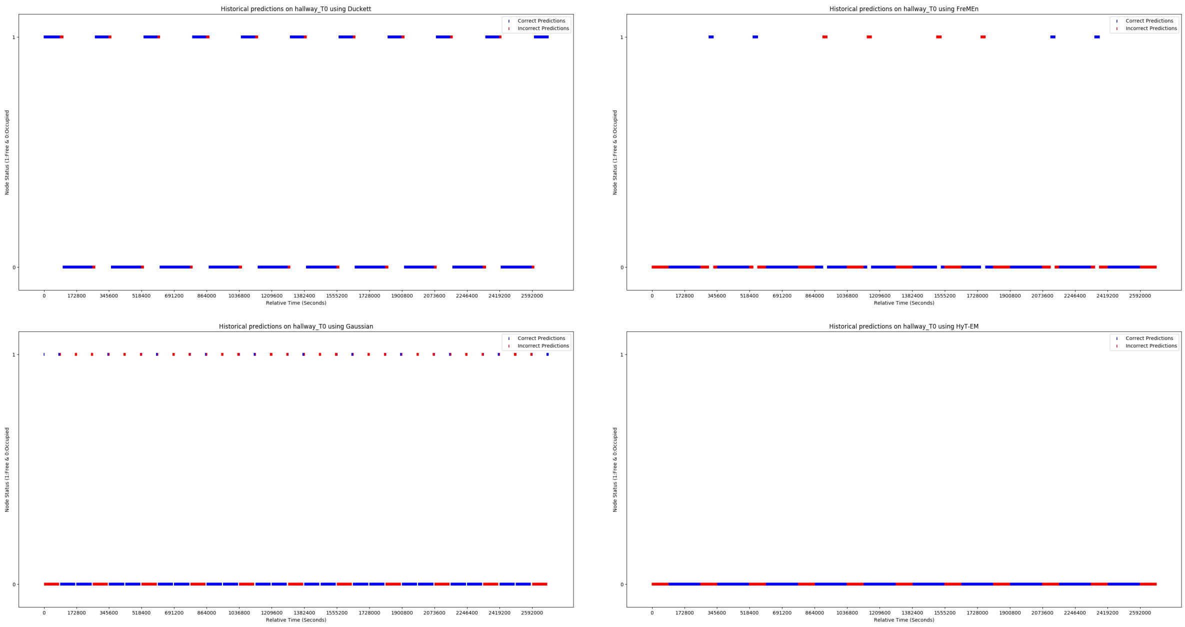

Figure 6.22: Historical Recreations - Hallway Trash Section 0

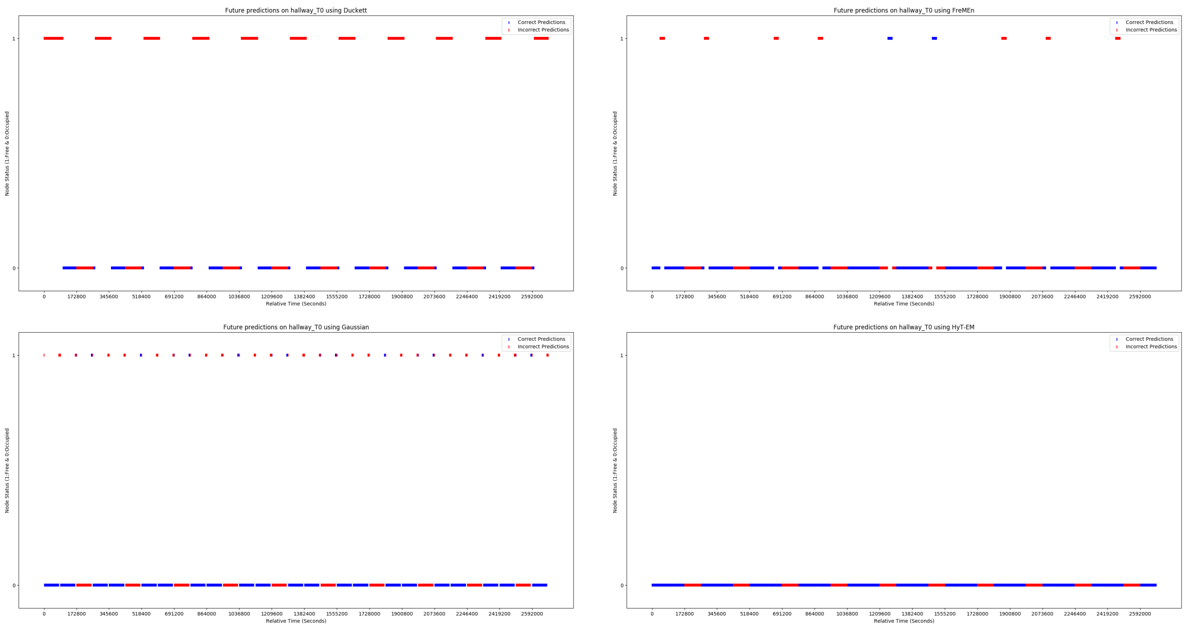

Figure 6.23: Future Predictions - Hallway Trash Section 0

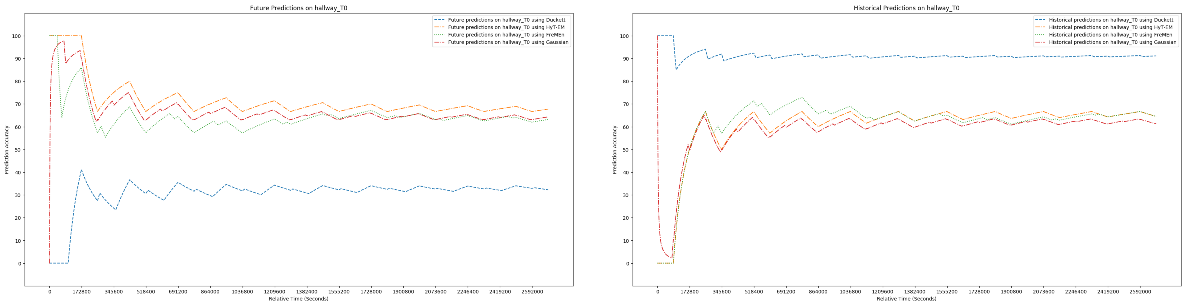

Figure 6.24: Model Accuracy Over Time - Hallway Trash Section 0

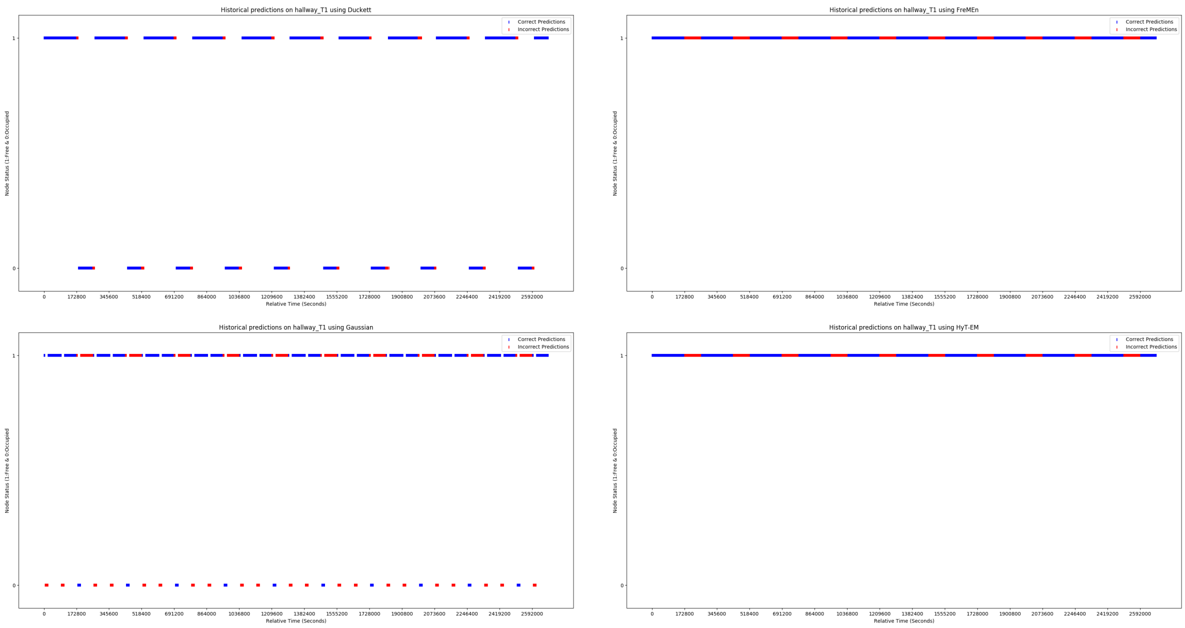

Figure 6.25: Historical Recreations - Hallway Trash Section 1

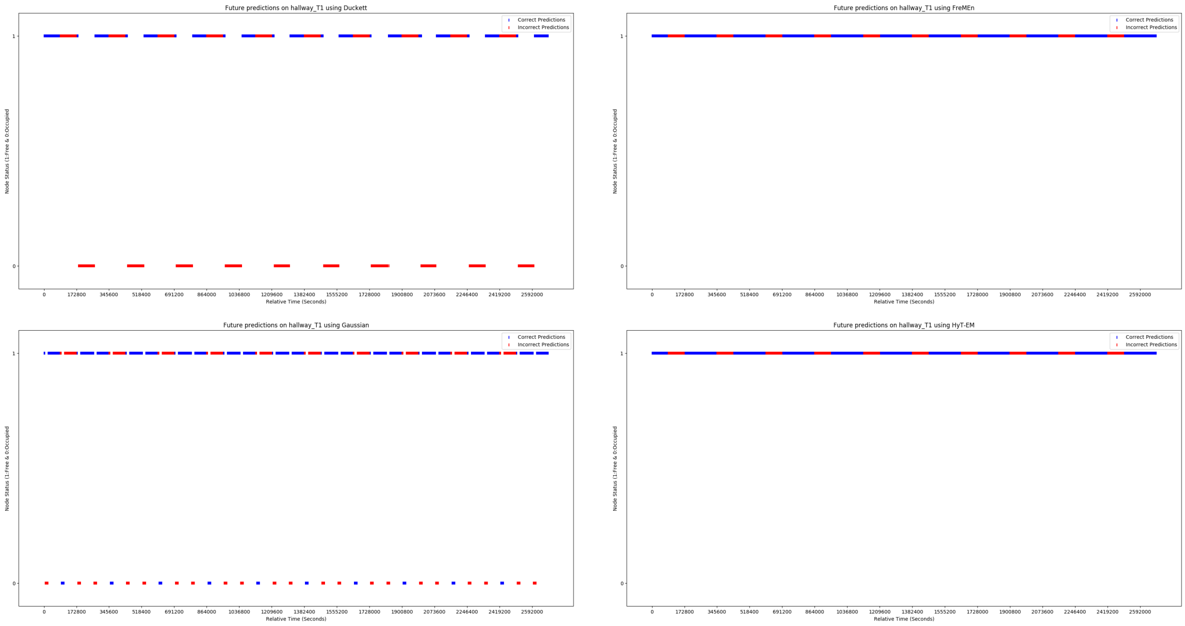

Figure 6.26: Future Predictions - Hallway Trash Section 1

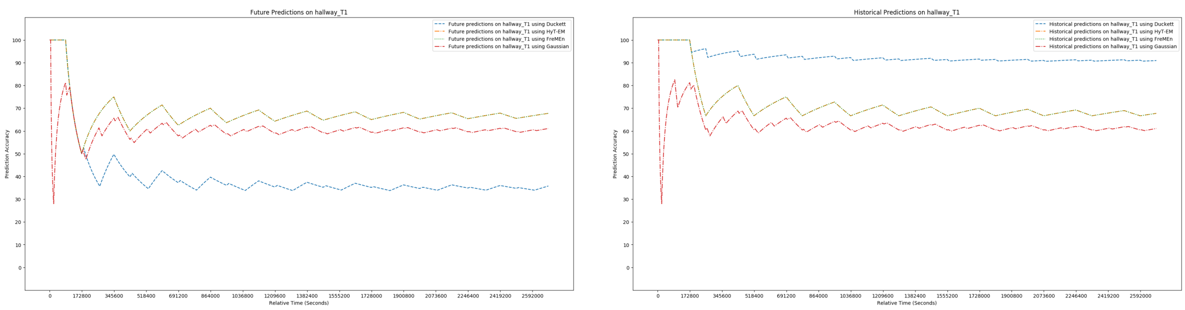

Figure 6.27: Model Accuracy Over Time - Hallway Trash Section 1

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	61.63%	52.08%	100.00%	100.00%
Prediction Accuracy	61.63%	52.08%	100.00%	100.00%
Computation Time (Milliseconds)	620	60	70	880
Memory Usage (KB)	30896	35524	35600	38288

Table 6.7: Hallway Meal Section 0

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	74.93%	65.62%	100.00%	100.00%
Prediction Accuracy	74.93%	65.62%	100.00%	100.00%
Computation Time (Milliseconds)	600	60	70	990
Memory Usage (KB)	31384	35388	35588	37608

Table 6.8: Hallway Meal Section 1

	Duckett	Gaussian	FreMEn	HyperTime
Historical Accuracy	78.93%	26.04%	100.00%	100.00%
Prediction Accuracy	78.93%	26.04%	100.00%	100.00%
Computation Time (Milliseconds)	610	60	70	1860
Memory Usage (KB)	30984	35160	35172	37632

Table 6.9: Hallway Laundry Section

	Duckett	Gaussian	FreMEn	HyperTime
Number of Hard Errors	1790	2418	1488	1488
Number of Soft Errors	442	62	80	80
Average Additional Cells Traversed	5.74	6.34	3.12	3.12

Table 6.10: Historical Path Planning Results

	Duckett	Gaussian	FreMEn	HyperTime
Number of Hard Errors	1790	2418	1488	1488
Number of Soft Errors	444	62	80	80
Average Additional Cells Traversed	5.74	6.34	3.12	3.12

Table 6.11: Future Path Planning Results

7 Conclusions

7.1 Contributions

7.2 Lessons learned

7.3 Future work

A

Design Details

Your first appendix

B

Parameters

Your second chapter appendix

References

[1] D. Arbuckle, A. Howard, and M. Mataric. Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. *IEEE/RSJ International Conference on Intelligent Robots and System*, 1(789349985):409–414, 2002. doi: 10.1109/IRDS.2002.1041424.

[2] Claudio Coppola, Tomas Krajnik, Tom Duckett, and Nicola Bellotto. Learning Temporal Context for Activity Recognition. *Ecai 2016: 22Nd European Conference on Artificial Intelligence*, 285:107–115, 2016. ISSN 0922-6389. doi: 10.3233/978-1-61499-672-9-107.

[3] Alberto Elfes. A Sonar-Based Mapping and Navigation System. *Computer*, pages 19–24, 1985.

[4] Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett. Spectral Analysis for Long-Term Robotic Mapping. *IEEE International Conference on Robotics and Automation*, pages 3706–3711, 2014.

[5] Jaime Pulido Fentanes, Bruno Lacerda, Tomas Krajnik, Nick Hawes, and Marc Hanheide. Now or later? Predicting and maximising success of navigation actions from long-term experience. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):1112–1117, 2015. ISSN 10504729. doi: 10.1109/ICRA.2015.7139315.

[6] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrová, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Körtner, Rares Ambrus, Nils Bore, John Folkesson, Patric Jensfelt, Lucas Beyer, Alexander Hermans, Bastian Leibe, Aitor Aldoma, Thomas Faülhammer, Michael Zillich, Markus Vincze, Eris Chinellato, Muhannad Al-Omari, Paul Duckworth, Yiannis Gatsoulis, David C. Hogg, Anthony G. Cohn, Christian Dondrup, Jaime

Pulido Fentanes, Tomáš Krajník, João M. Santos, Tom Duckett, and Marc Hanheide. The STRANDS Project: Long-Term Autonomy in Everyday Environments. *IEEE Robotics and Automation Magazine*, 24(3):146–156, 2017. ISSN 10709932. doi: 10.1109/MRA.2016.2636359.

[7] Ferdian Jovan, Jeremy Wyatt, Nick Hawes, and Tomáš Krajník. A poisson-spectral model for modelling temporal patterns in human data observed by a robot. *IEEE International Conference on Intelligent Robots and Systems*, 2016-November:4013–4018, 2016. ISSN 21530866. doi: 10.1109/IROS.2016.7759591.

[8] Tomáš Krajník, Jaime P Fentanes, João M Santos, and Tom Duckett. FreMEn: Frequency Map Enhancement for Long-Term Mobile Robot Autonomy in Changing Environments. *EEE Transactions on Robotics*, 33(4):964–977, 2015.

[9] Tomáš Krajník, Jaime Pulido Fentanes, Marc Hanheide, and Tom Duckett. Persistent localization and life-long mapping in changing environments using the frequency map enhancement. *IEEE International Conference on Intelligent Robots and Systems*, 2016-November:4558–4563, 2016. ISSN 21530866. doi: 10.1109/IROS.2016.7759671.

[10] Joao Pedro Machado dos Santos. Lifelong Information-driven Exploration for Mobile Robots to Complete and Refine Spatio-Temporal Maps in Changing Environments. *University of Lincoln*, page 130, 2017.

[11] Daniel Meyer-Delius, Maximilian Beinhofer, and Wolfram Burgard. Occupancy Grid Models for Robot Mapping in Changing Environments. *Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 2024–2030, 2012.

[12] L.R. R Rabiner. Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Ieee*, 77(2):p257–286, 1989. ISSN 00189219.

[13] Fabio Ramos and Lionel Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *International Journal of Robotics Research*, 35:1717–1730, 2016. ISSN 17413176.

[14] Joao Machado Santos, Tomas Krajnik, Jaime Pulido Fentanes, and Tom Duckett. Lifelong Information-Driven Exploration to Complete and Refine 4-D Spatio-Temporal Maps. *IEEE Robotics and Automation Letters*, 1(2):684–691, 2016. ISSN 2377-3766. doi: 10.1109/LRA.2016.2516594.

[15] Ransalu Senanayake, Lionel Ott, Simon O'Callaghan, and Fabio Ramos. Spatio-Temporal Hilbert Maps for Continuous Occupancy Representation in Dynamic Environments. *Advances in Neural Information Processing Systems 29*, pages 3918–3926, 2016. ISSN 10495258.

[16] Ransalu Senanayake, Simon O'Callaghan, and Fabio Ramos. Learning highly dynamic environments with stochastic variational inference. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2532–2539, 2017. ISSN 10504729. doi: 10.1109/ICRA.2017.7989294.

[17] Zhan Wang, Patric Jensfelt, and John Folkesson. Modeling Spatial-Temporal Dynamics of Human Movements for Predicting Future Trajectories. 2015.