Data-Driven Methods for Time Series Forecasting, Classification, and Uncertainty Quantification

Diya Sashidhar

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

J. Nathan Kutz, Chair

Thomas D. Rea

Ivana Bozic

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Data-Driven Methods for Time Series Forecasting, Classification, and Uncertainty Quantification

Diya Sashidhar

Chair of the Supervisory Committee:
Professor J. Nathan Kutz
Department of Applied Mathematics

The increased availability of time series data has led to a burgeoning interest in data-driven modeling and time series analysis. The ability to model temporal data can not only enable us to discover patterns inherent in a set of collected measurements, but also predict future trends. However, collected temporal measurements are oftentimes artifacted with noise, making it difficult to discern the actual signal. This presence of noise can greatly bias models, resulting in inaccurate forecasts with high uncertainty. In this thesis, I demonstrate how data-driven methods can be applied to a wide array of artifacted data while circumventing noise-induced bias.

I first show the application of various data-driven methods and signal processing techniques on labeled time series data. Specifically, I apply supervised machine learning and signal processing techniques on corrupted electrocardiograms (ECGs) in order to classify pulse status in patients undergoing cardiac arrest.

I then introduce a data-driven method that leverages statistical bagging [18] and optimized Dynamic Mode Decomposition (optDMD) [8] in order to produce accurate long-term forecasting and spatial and temporal uncertainty quantification for unlabeled, non-stationary time series. I then highlight the robustness of this method by applying it to corrupted flu transmission data in order to predict future flu trends as well as gain insight into temporal cycles of modes.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I'm incredibly thankful to have such a supportive and wonderful advisor, Nathan Kutz. Nathan not only helped me become an independent scientist, but also supported whatever steps I needed to take in order to further my career. Nathan set me up with collaborations that helped me gain perspective on the biological aspects of research and encouraged me to take summer internships that were starkly different from my research. His undying optimism and enthusiasm made me continue to enjoy math even when projects had not gone as planned. Apart from advising me academically, Nathan taught me how to be a better professional. Nathan taught me how to manage my time, work efficiently, and advocate for myself.

Nathan not only supported my growth as a researcher, but also my general well-being. There were times during the course of my graduate school career where I felt lost and dejected and it was truly a blessing to have an understanding and supportive advisor. When I came to this department, I never imagined having so many mutual interests as my advisor. I will always cherish the beginning of our meetings where we'd drink coffee and discuss the latest season of football. I'll definitely miss arguing about our views on Cam Newton as a quarterback and whether the Seahawks will make it to the Super Bowl. Having Nathan as an advisor is truly a dream come true.

I'd also like to thank Drs. Tom Rea, Peter Kudenchuk, Heemun Kwok, Jason Coult, and Jennifer Blackwood for taking me under their wing and introducing me to resuscitation research. It has truly been a delight getting to know each of them and I couldn't have asked for better collaborators. Their passion and enthusiasm for the field is infectious and always motivated me to ask more questions and seek solutions. Their provided insights and wisdom helped me grow as a researcher and think critically about scientific problems. I'm especially thankful for the help provided by

Heemun and Jason. Their vast knowledge of the field, work ethic, and attention to detail was integral to the success of my work.

I'd like to thank Mauricio Santillana. His insights and epidemiological contributions were incredibly valuable and deepened my understanding of infectious disease modeling. I'd also like to thank my committee members for taking the time read through my thesis and provide suggestions. I'd also like to thank my cohort. My cohort was instrumental in my preparation for qualifying exams and coursework and were great companions throughout my PhD. I'm thankful for having a cohort comprised of friends. Our weekly happy hours, Parnassus runs, and food runs on the Ave will always be dear to me.

I'm also grateful for my North Carolina friends: Jamie Plummer, Ashlyn Johnson, Brooke Whitfield, Madison Bridges, Adrienne Maung, Abhishek Komandur, and Cody and Ashley Long. During my time in graduate school, I often felt deep pangs of homesickness. My weekly calls with Jamie had the remarkable ability of making thousands of physical distance feel like a few feet. Knowing that Jamie is just a phone call away and has provided me much solace throughout my PhD career. Our shared interests of food, TV, and and great deals spurred many of our conversations, which immediately effaced any feelings of loneliness. Ashlyn provided endless sources of wisdom and support throughout my college and PhD career. Our late night trips to D.H. Hill are times I'll always cherish. Having started the graduate school process together, I found it incredibly comforting to have someone like Ashlyn to share various aspects of graduate school. My calls with Cody and Ashley were incredibly comforting as well. Our love for gossip sparked many of our conversations and never ceased to invigorate me. My friendship with Brooke started in high school over our hatred for chemistry and has continued to grow ever since. Despite having lived on opposite sides of the country for the past few years, Brooke has been a prevailing presence in my life. Throughout my darkest times of graduate school, I knew I could count on Brooke to cheer me up with her endless humor and snarky remarks.

I'm grateful for my friendship with Madison. Madison introduced me to the mandolin, which greatly improved mental health during my fourth year. She also introduced me to the whole genre of bluegrass music. Her carefully curated playlists helped me get through the toughest times of my PhD. Her passion for music and optimistic attitude are truly contagious, and have positively impacted my life.

My friendship with Adrienne and Abhishek started in middle school. They've supported me throughout most of my life and I'm very much thankful for that. Adrienne and Abhishek's quick wit and humor make me immediately forget about any worries on my mind and launch into laughter. Adrienne and I also share a love for trashy reality TV. Our Love is Blind zoom sessions helped me get through much of the pandemic.

I'm grateful for my friendship with Bobby Baraldi. Having gone to the same high school, college, and graduate school as me, Bobby could empathize with much of my PhD experience. It was always comforting knowing that a fellow North Carolinian was just a few offices away. Bobby's presence helped fight much of the homesickness I experienced, especially during my first year. Bobby's determination, ambition, and work ethic never ceases to amaze me. His ability to be incredibly productive motivated me to strive harder throughout my graduate school career. I'm thankful for my friendship with Weston Barger. Our love for fried food, coffee, and college football made us instantly friends. Weston provided academic and emotional support throughout most of my PhD. Weston helped me improve my coding skills by teaching me about Python pipelines and was always down for Orange King when I was dejected by my research. My friendship with Weston kept me afloat for much of my PhD, and I'm so grateful to have him in my life. From providing great advice to sending me into bouts of belly laughs, Bobby and Weston have profoundly impacted my life more than words can express.

I'd also like to thank Sheridan Grant. There were many ups and downs during my graduate career, and Sheridan was there every step of the way. Sheridan believed

in me during times where I myself did not. His constant support and encouragement helped me achieve my best performance and even instilled confidence in myself. Sheridan also introduced me to hiking and backpacking, a hobby I very much relied on in graduate school. Together, Sheridan's fun-loving spirit and witty humor has the ability to turn my worst days into my best ones.

Finally, I'd like to thank my family. My dad is an integral part of my love for math. He was dedicated to providing me a great math education, starting at a young age. Throughout my childhood, my dad would test me on math concepts and challenge me to think critically. His constant support and mentorship throughout these years has directly shaped me into the mathematician I am today and I am forever thankful. I'd also like to thank my mom for providing endless emotional support and wisdom. The number of hours I've talked to my mom during the course of my PhD is truly astounding, and I am thankful for every minute of it. She taught me skills such as perseverance and good work ethic, which were essential for the completion of my PhD. She and my dad served as constant role models throughout my life, motivating me to always strive harder. It is truly a blessing to have such inspiring figures as parents. I'd also like to thank my brother, Dev. Dev has the remarkable ability of brightening any room he sets foot in. His witty humor and ability of quote literally anything (from Shrek 2 to a year-old conversation) has uplifted me even during my darkest times. My brother inspires me every day with his wisdom, maturity, patience, and compassion. I'm lucky to have my best friend as my brother. I'd also like to thank my aunty, Pushpa Vadlamudi. Having received a physics PhD herself, Aunty provided empathy and wisdom throughout my graduate school career. Her vast knowledge of plant care and endless support helped me tremendously throughout my time as a graduate student, and I am grateful for that.

DEDICATION

To my family: Chuck, Sree, and Dev

Chapter 1

INTRODUCTION

With advances in technology, we see a rise in the availability of data within a myriad of fields such epidemiology, medicine, and fluid dynamics. This abundance of available data makes processing and interpreting data dynamics quite hard without the use of signal processing and advanced data science techniques. Data-driven modeling and machine learning has profoundly impacted our ability to understand underlying phenomena as well as predict future states. From predicting weather patterns to self-driving cars, data-driven modeling and machine learning have greatly shaped not only our day-to-day lives but also transformed scientific and engineering fields [38, 99].

However, understanding and predicting future data trends can be quite difficult, as much of this data originates from complex, yet dynamic systems. These dynamical systems have many different moving parts and are often sensitive to many external factors. For example, in the case of epidemiological data, much of this data can be confounded by environmental, geographical, and even political factors. Such factors can even affect data collection, resulting in missing data for long periods of time or even sharp changes in behavior.

Albeit quite challenging, forecasting data trends is a valuable tool, as it enables one to predict future behaviors of complex systems. Furthermore, knowledge of these behaviors can promote different courses of action in response. We rely on these forecasts to inform the decisions we make in our daily lives. For example, on an individual scale, knowledge of the future weather patterns affects what we pack for a trip. On a larger scale, knowledge of predicted adverse weather patterns can result in issued warnings for evacuation. These decisions based on time series forecasts extend to even modern medicine. In the case of resuscitation data, knowledge of a patient's underlying physiological status can prompt rescuers to make informed decisions and hence take different medical actions, potentially altering a patient's survival outcome [27].

While data comes in many forms, we focus on the subset of time series data. Time series data is especially difficult to model as it is sequential and has the distinct characteristic of temporal ordering. More importantly, we'd like to extract key behaviors and features of this data over an interval of time and predict future values and states. However, unlike with other types of data, forecasting with time series data can be quite hard to validate, as we are predicting trends of time points that have not occurred yet and have the potential to be affected by external factors that aren't present in the training data.

With the help of time series analysis and machine learning, we can learn a great deal from this data. The nature of this data influences the scientific questions posed and hence, the modeling approach. For instance, time series with have labeled outcomes (labels) can enable classification of time series. With labeled datasets, a multitude of powerful supervised learning techniques such as Neural Networks and Support Vector Machines can be applied, allowing us to predict various outcomes associated with a given time series.

Most time series, however, do not have labeled outcomes. In the case of medical data, for instance, times series and images are labeled by professionals in the field. Thus, much labor is required in order to prepare a labeled dataset. As a result, these labeled sets can be hard to find. Without these labels, supervised learning is no longer be applicable. However, although the possibilities within supervised learning are limited, there is still much that can be done with unlabeled datasets. In the case of time series data specifically, we can apply data-driven modeling to create diagnostics of a data set. In the case of seasonal or periodic data, such diagnostics can be used to capture independent modes of the data and their respective periodicities [22]. Specifically, we can capture areas that have spatial variance and oscillating dynamics inherent in the data itself. When combined, these modes (eigenvectors), periodicities (eigenvalues), and amplitude, can not only reconstruct the time series, but also forecast future data points.

Motivation

In the realm of mathematical modeling, we assume that outputs of a dynamical system are associated with its inputs. Broadly speaking, we aim to find a parameterized function, $f_\theta : \mathbb{R}^{m \times n} \to \mathbb{R}^m$ to enables us to predict m outputs, $\mathbf{Y}$, based on m inputs, $\mathbf{X}$:

$$\mathbf{Y} = f_\theta(\mathbf{X}) \tag{1.1}$$

This value of $y \in \mathbf{Y}$ varies greatly: it can be a classification of an input, or a *label*, or even the next value in a time series. While both cases can lead to important discoveries of the data and future states, the value of the independent variable, y, greatly affects which questions are asked. The case of y being a label prompts questions relating to classification while the latter case prompts questions related to future *values* of the time series. Together, the value of y and the resulting scientific question influences the choice in model.

Supervised machine learning can be used to answer questions that arise from datasets with labeled outcomes, or when y_n is a single value associated with a given input x_n for $n = 1...m$. This family of models can also be applied in order to answer questions regarding forecasting, such as predicting the value of the next time point in a series. This family of algorithms has be used for a myriad of tasks ranging from image classification to pattern recognition [38, 99]. Models such as Neural Networks and Support Vector Machines learn inherent patterns in the data and associate them with the provided outcomes. A classic example is the application of Neural Networks to labeled dog ($y = 1$) and cat ($y = -1$) image libraries in order to classify these images. When trained on large amounts of data, these models can classify a wide range of cat and dog images with remarkable accuracy. Neural Networks can also be used to learn trajectories of a set of ordinary differential equations (ODE). These neural networks are trained by using the next step of a ODE system, $y = x_{n+1}$ as the outcome a given input, x_n for $n = 1...m$ time points. With this framework, these networks can map trajectories for differential equations as complex as the Lorenz equations, sometimes surpassing the performance of Runge-Kutta-based ODE solvers [22]. While this broad category of models can be very powerful, these methods often require large amounts of data and computational time. Thus, in cases where the data is scarce, these models are no longer viable options.

Unsupervised learning is a family of methods that are applicable when the value of y is unknown. These methods apply transformations, dimensionality reduction, and clustering techniques in order to find these values of y. As opposed to supervised learning, an appealing

feature of this family of the methods is the ability to visually inspect distinctive grouping of various of time series. K-means clustering, for instance, groups points in clusters by classifying points based on their distance from the pre-specified initial conditions. These pre-specified initial conditions are replaced by means of the new clustered points, causing the re-classification of points in an iterative manner until the convergence criteria is reached (points are no longer re-classified). While these methods have many advantages such as computationally cheapness and data visualization, there are some drawbacks to using this family of methods. Finding this distinctive grouping of data points is difficult in practice. This distinctive grouping depends greatly on not just the right transformation, but the correct *order* of data transformations. In addition, unsupervised learning can be a risky approach, as it is not guaranteed that the assigned grouping of points make sense.

In the case of temporally ordered data, we wish to predict the next p values a time series based on past values n values. Specifically, we adapt Equation 1.1 such that

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_m) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k \times m} \tag{1.2}$$

and

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_{m+1}) & \mathbf{x}(t_{m+2}) & \cdots & \mathbf{x}(t_{m+p}) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k \times p}, \tag{1.3}$$

where $m, p, k \geq 1$. The dimensionality of the variables, $x(t_i) \in \mathbb{R}^k$ for $i = 1...m$ is representative of the type of time series data. In the case where k = 1, $x(t_i)$ can be temporal ordered data. For $k > 2$, $x(t_i)$ can be vectorized snapshots of the data, or more broadly speaking, spatiotemporal data. Similar to the value of the outcome variable, y, the dimensionality of the dependent variable, $x(t_i)$, greatly affects the resulting analysis.

As with many predictions, the ability to quantify certainty of model outputs and estimates is an invaluable tool. In other words, for every outputted $y \in \mathbf{Y}$, we'd like to generate a range of possible values for this y. These ranges, or confidence intervals, can provide insight regarding how far we can predict a given time series with reasonable certainty. In the

case of spatiotemporal data ($k > 2$), this uncertainty quantification (UQ) can be applied in a spatial setting, enabling the user to see which spatial regions in the model have high uncertainty.

We now introduce methods in the statistical and machine learning realm used for predicting future data points in a time series. These methods are often used temporal data, but in some cases can be applied to spatiotemporal data. We also provide a brief background on UQ, a tool that when integrated into models can not only increase model robustness, but also provide valuable insight regarding model uncertainty.

1.1 Statistical Models for Time Series Forecasting

The most common statistical methods for time series forecasting reside in the family of Box-Jenkins models [91]. This family includes models such as autoregressive moving average models (ARMA), autoregressive integrated moving average models (ARIMA), and seasonal autoregressive moving average models (SARIMA), which are based on the assumption that the current (and future) value of a stationary time series relies on past values and errors of the times series [54].

One of the main components of these models is the Autoregressive (AR) model. This model defines the current value, x_n, as a linear combination of the past p values of the time series, x_{n-k} for $k = 1...p$. An autoregressive model of order of p, AR(p), is defined below [91]:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ...\phi_p x_{t-p} + w_t, \tag{1.4}$$

where w_t is the noise at time t and $\phi_i \in \mathbb{R}$ for $i = 1...p$. The order of p in this case is pre-specified and is often attained via grid-search.

The other component of these models is the Moving Average (MA) model. This model defines the current value, x_n, as a linear combination of the past q errors of the time series. A moving average model of order of q is defined below [91]:

$$x_t = \theta_1 w_{t-1} + \theta_2 w_{t-2} + ...\theta_q w_{t-q} + w_t, \tag{1.5}$$

where w_i for i = 1...q is the noise for the past q time points and $\theta_i \in \mathbb{R}$ for $i = 1...q$. Similarly, the order of q in this case is pre-specified and is often attained via grid-search.

In the case of ARMA, these two models can be combined to present a linear combinations of the past p values and q errors as shown below [54]:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ...\phi_p x_{t-p} + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ...\theta_q w_{t-q} + w_t, \tag{1.6}$$

where $\phi_i \in \mathbb{R}$ for $i = 1...p$ are coefficients for the past p values, w_i for i = 1...q is the noise for the past q time points, and $\theta_i \in \mathbb{R}$ for $i = 1...q$ are their corresponding coefficients [91].

However, this type of modeling is only applicable to series that are agnostic of time, or stationary series. Specifically, these time series do not have trends or seasonality associated with them. In these cases, ARIMA integrates differences (via its differencing parameter, d) to circumvent issues associated with trends and SARIMA incorporates seasonal autoregressive, seasonal moving average, and seasonal differencing orders, P,Q, and D respectively [54].

In the case of forecasting, we predict the next value based on the past n values. In other words, assuming order n, $x_{n+1} \in \mathbb{R}^k$, is forecasted a linear combination of past values of the time series. For simplicity, we write for AR(n):

$$\mathbf{x}_{n+1} = f(\mathbf{x}_1, ...\mathbf{x}_n) \tag{1.7}$$

The list of dependencies grows when we forecast with ARMA(n,n):

$$\mathbf{x}_{n+1} = g(\mathbf{x}_1, ...\mathbf{x}_n, \mathbf{w}_1, ...\mathbf{w}_n) \tag{1.8}$$

As model complexity increases, more parameters are incorporated in the model, making hyperparameter grid searches computationally expensive. While the models such as SARIMA can accommodate for non-stationary time series, the computational cost can be a major drawback. In addition, these recursive, one-step-ahead methods can compound errors over time, resulting in poor long-term forecasts [68].

1.2 Machine Learning Models for Time Series Forecasting

In the realm of machine learning, Neural Networks are the prevailing models in time series forecasting. These models are comprised of many pre-specified composite functions, or lay-

ers, that map given inputs to the corresponding output. Similar to Equation 1.1, a Neural Network is trained with inputs, X, and outputs, y. During this training process, a subset of the predictor and response variables are inputted into the model. Using optimization and backpropagation, the model then learns the associated weights of these layers, or connectivity matrices [22], in order to best fit the inputted and outputted data. In other words, the neural network can be thought as the following:

$$\mathbf{y} = f_M(\mathbf{A}_M, ...f_2(\mathbf{A}_2, f_1(\mathbf{A}_1, \mathbf{x})), \tag{1.9}$$

where $f_1, ...f_M$ are activation functions that map one layer's outputs to the subsequent layer's inputs and $A_1, ...A_M \in \mathbb{R}^{k \times k}$ are the corresponding connectivity, or weight, matrices in a m-layer Neural Network [22]. In the case of time series forecasting, the x in this case can be the current time point, x_n, while y can be the following time point, x_{n+1}. This procedure is applied in a systematic fashion in order to get the optimized weights for this dynamical system.

Recurrent Neural Networks, on the other hand, recursively change its weights and hidden states as it feeds time points into the network one by one [22]. Unlike a basic Feed Forward Neural Network (Equation 1.9), this neural network is not stateless and relies greatly on temporal ordering of the input series. A given state is dependent on the previously calculated hidden state, h_{t-1}, and the current time point.

$$h_t = f(h_{t-1}, \mathbf{x}_t) \tag{1.10}$$

The trained weights are then applied to the hidden state to calculate the outcome, y_t. These weights can be applied to a time series in order to forecast future time points.

However, one drawback of the RNN framework is the lack of long-term memory [75]. The RNN has trouble remembering hidden states that are several time steps prior. As a result, important information and dynamics can get lost over time. Models such as Long Short Term Memory (LSTM) and Gated Recurrent Units (GRUs) prove to be leading models in time series forecasting purely because of their gated memory [90]. Specifically, each node possesses the ability to remember and forget long-term information. Known for

its success in speech recognition, the LSTM architecture contains forget gates that enable it to remember important information and carry this information throughout its training process [75]. Similarly, a GRU possesses an update gate, which essentially filters important data and re-evaluates which information to keep at each step.

As opposed to statistical methods like ARIMA, these models work by recognizing inherent patterns in the data. Moreover, instead of taking the one-time-step ahead approach based on the preceding time points, these models work by learning the weights associated with the pre-specified layers and applying these weights to a completely different test set. In the case of a feedforward net (Equation 1.9), we simply input a test point, x_m and predict $y = x_{m+1}$ using our newly trained connectivity matrices, $A_1, ..., A_M$. Similarly, for the RNN frameworks, we can choose to predict the next time point or even a single-shot prediction in which the trained model predicts the next m time points.

While these models can be quite powerful, there are a few drawbacks regarding their implementation. They require large amounts of data for their training sets and test sets (as opposed to ARIMA). In addition, these models are black box approaches - much of the inner mechanisms are unbeknownst to the the user. Thus, while these tools can forecast future behavior, it does not provide insight to the underlying dynamics of the system.

1.3 Koopman Theory and Forecasting via Dynamic Mode Decomposition models

In addition to this multitude of statistical and machine learning forecasting methods, there is a burgeoning field of research devoted to Koopman theory-based regression frameworks geared towards time series analysis. Koopman theory is based on the assumption that we can represent a non-linear dynamical system in terms of an infinite dimensional linear operator, K, acting on a Hilbert space of measurement functions, g [22]. This can be represented in the following way:

$$K_t g = g \circ F_t, \tag{1.11}$$

where F is a discrete-time dynamical system of the form:

$$x_{k+1} = F_t(x_k), \tag{1.12}$$

where x_k represents the state of the system at time k for $k = 1...n$ measurements. In the case of a discrete-time dynamic system, Equations 1.11 and 1.12 can be combined to form Equation 1.13.

$$Kg(x_k) = g(x_{k+1}) \tag{1.13}$$

Thus, for all g and x_k, Koopman operator is able map the observable state at any given time step to the subsequent time step [63].

There are many benefits to representing these dynamics in a linear framework. These frameworks are not only easy to work with but are heavily researched and well understood. However, obtaining a infinitely-dimensional linear operator is still quite difficult. In lieu of finding this linear operator, we aim to obtain a finite-dimensional matrix approximation. One key aspect of this matrix approximation is that we can generate its spectral decomposition and hence model inherent behaviors of this nonlinear system. This approximation can be achieved via Dynamic Mode Decomposition [88,89], a well-known data-driven regression framework that is strongly connected to Koopman spectral theory.

Similar to Equation 1.13, Dynamic Mode Decomposition (DMD) uses a direct set of measurements to create a linear approximation of the Koopman operator that advances spatiotemporal data forward in time. More precisely, the goal of DMD is to find linear mappings from one point to the next in a given spatiotemporal series [63]. DMD is an elegant, yet appealing modeling tool as it merges the dimensionality reduction capabilities of Singular Value Decomposition with the temporal frequency identification capabilities of Fast Fourier Transforms. This algorithm aims to obtain the best-fit linear approximation $\mathbf{A} \in \mathbb{R}^{k \times k}$ such that:

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k, \tag{1.14}$$

for j = 1...m where each $x_j = x(t_j) \in \mathbb{R}^{k \times m}$ are vectorized snapshots of the state of a system for m time points. However, while acquirement of this linear operator A can be very useful, calculating this linear operator can be computationally expensive, especially when spatiotemporal measurements are high dimensional. Moreover, calculating the corresponding spectral decomposition of A requires more energy, making this problem computation-

ally demanding. Rather than calculating the spectral decomposition of A explicitly, DMD circumvents this issue by applying dimensionality reduction techniques to generate eigenvectors, Φ, and eigenvalues, λ. Based on the provided spatiotemporal measurements, DMD creates eigenvectors, or modes, that correspond complex eigenvalues of the form $\lambda = a + bi$, where a is the growth or decay rate and b corresponds to the frequency of oscillation [22].

There are three outputs to a given DMD model: eigenvectors (Φ), eigenvalues (λ), and amplitude (b). Together, these three outputs can be used to not only reconstruct a given times series, but forecast future time points. The most basic DMD model, exact DMD, has the low rank approximation in continuous time :

$$x(t) = \sum_{j=1}^{r} \Phi_j e^{\omega_j t} b_j = \mathbf{\Phi} \exp(\mathbf{\Omega} t)\mathbf{b}, \tag{1.15}$$

where r is the rank of the approximation, Φ_j is mode j for $j = 1...r$, and with corresponding continuous eigenvalue, $\omega_j = log(\lambda_j)/\delta t$, and mode amplitude b_j. As shown in Equation 1.15, the following summation can be written in matrix form, where the jth columns of $\mathbf{\Phi} \in \mathbb{C}^{m \times r}$ correspond to Φ_j for $j = 1...r$, $\mathbf{\Omega} \in \mathbb{C}^{r \times r}$ is a diagonal matrix with ω_j along the diagonal, and $\mathbf{b} \in \mathbb{C}^r$ is a column vector comprised of the values b_j. Reconstruction of the time series can be generated by simply inputting $t = 1...n$. Likewise, a forecasted time series p time points ahead can be generated by inputting $t = n + 1...n + p$.

There are many advantages to using these data-driven regression-based frameworks. For one, these models are numerically simple to compute and do not require large amounts of data. While more data is always beneficial, DMD can run successfully on orders of magnitude less data than those required for neural networks. In addition, DMD does not require a significant train-test split of the data. Thus, one can simply devote most, or even all of the data towards the training process. As opposed to the statistical methods prevalent in this field, DMD can predict future forecasts in a single-shot and doesn't not take the one-time-step ahead approach as found in ARIMA. Finally, DMD is interpretable. Each of the eigenvalues and eigenvectors in this system contain a wealth of information such as periodicity, variance, as well as the dominance (amplitude) of each mode.

Over time and increasing interest in the DMD models, there have been adaptations of

the general DMD framework such as the total least-squares DMD [48], optDMD [9], and Forward-Backward DMD models [29], not only enabling non-uniform time intervals, but also accounting for inherent bias in the model. With each of these variants, the DMD abilities continue to grow, rendering more accurate models.

1.4 Uncertainty Quantification

In many of these models, we are forecasting values of time points that may not have even occurred yet, making validation impossible. During these cases, we'd like to know how good our model is. Specifically, we'd like to know how *certain* we are of the generated forecast. We can gain this knowledge via uncertainty quantification.

This broad category of quantification can be applied in many ways. Temporal uncertainty quantification can tell us how certain we are of temporal measurements such as periodicity or seasonality. In other words, we can generate confidence intervals surrounding mean periodicity of a given time series. Narrow bands can correspond to periodicities with high certainty. Spatial uncertainty can point to regions with high uncertainty. When applying models to spatiotemporal data, we'd like to know which regions the model can predict with certainty. By having spatial uncertainty quantification, we can gain an understanding of which regional forecasts to rely on. For example, forecasts of regions with lower uncertainty have the potential to have more accurate long-term forecasts.

1.5 Organization of Thesis

In this thesis, we introduce several ways in which time series forecasting can be applied to various sets of data. We start with a brief background on heart data and resuscitation. In Chapter 3, we demonstrate forecasting using signal processing and machine learning techniques on ECG data that has labels. As mentioned previously, oftentimes, time series data does not have an associated label, making supervised learning moot. In Chapter 4, we develop a method called Bagging, Optimized dynamic Mode Decomposition (BOP-DMD) that produces enables production of diagnostics, forecasting, and uncertainty quantification when applied to time series without labels. To showcase this method's abilities and robustness, we apply this method to a variety of datasets such as vortex shedding and sea surface

temperature data over the span of several years. In Chapter 5, we apply this method to noisy flu datasets for city, state and worldwide levels and compare its performance to those of other DMD models such as exact DMD [88] and optDMD [9].

Chapter 2

BACKGROUND: HEART DATA AND RESUSCITATION

2.1 The Electrocardiogram

The heart is both a mechanical and electrical dynamical system. It relies on electrical activity to induce shifts in potential throughout the heart tissue, causing the physical contraction and relaxation of various heart chambers. The contraction and relaxation of various heart chambers enables the heart to mechanically pump oxygenated blood throughout the body in a synchronized manner. Together, the combination of mechanical and electrical activity is crucial for a heart to work independently.

Voltage activity over time is measured by an electrocardiogram, or ECG. These measurements are taken by placing electrodes on a person's chest and are (in a working heart) periodic series of cardiac cycles. During a cardiac cycle, the heart sends electrical signals through its tissue (depolarization and repolarization), causing electrically-inducible cells within the heart tissue to contract or relax. These currents cause a shift in heart potential over time, which can be monitored in an ECG.

In Figure 2.1, we see the hallmark features of an ECG with very little noise. The P wave corresponds to atrial contraction, while the QRS wave corresponds to ventricular contraction. Finally, we see the T wave which corresponds to ventricular relaxation.

This electrical activity is a pre-requisite for mechanical heart activity. That being said, presence of electrical activity does not imply presence of a pulse, or sufficient mechanical action to pump blood throughout the body. Unlike electrical activity, mechanical activity cannot be monitored in an ECG and requires manual measurement by placing two fingers over a major artery.

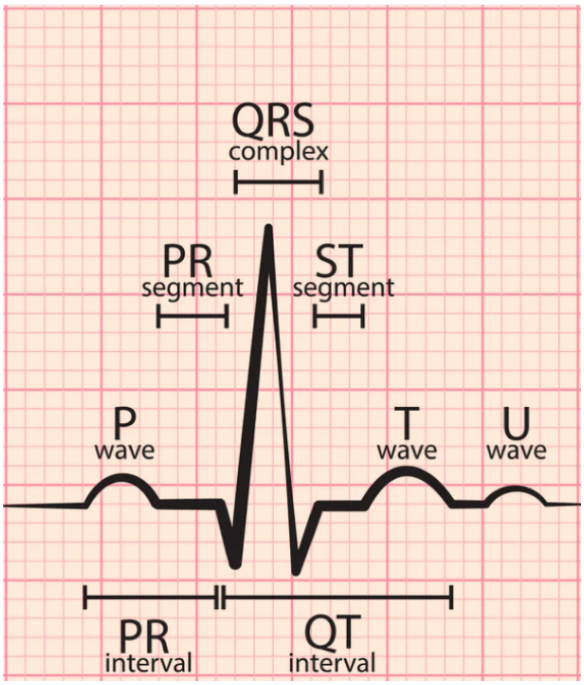

Figure 2.1: Sample ECG with labeled P, QRS, and T waves [82]

2.2 Resuscitation

However, sometimes the heart can fail to pump blood throughout the body, resulting in cardiac arrest. Out of hospital cardiac arrest occurs in over 300,000 people in the U.S. [14, 74], with a small percentage of patients resulting in survival. In order to help restore vital heart functions, rescuers and medical professionals provide a specialized therapy comprised of carefully administered defibrillation shocks, medication, and cardiopulmonary resuscitation (CPR) to the patient undergoing cardiac arrest.

2.2.1 Heart Rhythms

Using readings from the ECG, rescuers identify the underlying heart status and decide on the best course of action. Heart rhythms fall in three major categories: Ventricular fibrillation, Asystole, and Organized rhythms. Figure 2.2 shows a comparison of these three classes of rhythms. The first two categories are heart arrythmias, or conditions in which

the heart is not functioning properly, while the latter represents a heart with synchronized, periodic electric activity.

Ventricular fibrillation is a condition where the ventricles 'fibrillate' in an asynchronous manner. This uncoordinated activity amongst these chambers can be detrimental to overall heart health as it compromises the hearts ability to efficiently pump blood and even results in the loss of blood [59]. Figure 2.2 shows VF in the top panel, with aperiodic, low amplitude waves.

Asystole, or 'flatlining', is a condition where the heart does not produce sufficient electrical activity. Lack of electrical activity results in an ability to pump blood throughout the body and requires a series of chest compressions, or CPR, to manually restore blood flow throughout the body [59].

Organized rhythm is a rhythm with synchronized, periodic activity. However, this rhythm does not necessarily imply that the heart is pumping blood efficiently. In some cases, the heart demonstrates electrical activity, but not mechanical activity. In other words, the heart is going through the correct motions electrically, but is not able to physically pump blood throughout the body. This is called Pulseless Electrical Activity (PEA) and can only be verified by a manual pulse check.

2.2.2 Resuscitation Therapies

As mentioned previously, identification of heart rhythms informs rescuers' subsequent actions. These actions include defibrillation shocks, medication, CPR, manual pulse checks. We will focus on the latter two therapies, as they are central themes of Chapter 3.

CPR is a series of chest compressions and artificial respirations administered to a patient undergoing cardiac arrest. Administered at a rate of 100-120 compressions per minute [5], the goal of CPR is mimic the beat of a working heart and manually circulate blood throughout the body. As it only generates a third of a working heart's cardiac output, CPR is no replacement for original heart activity and is solely a therapy to prevent loss of blood from the brain and heart.

CPR can severely corrupt the output ECG [5]. As a result, it can be hard to assess

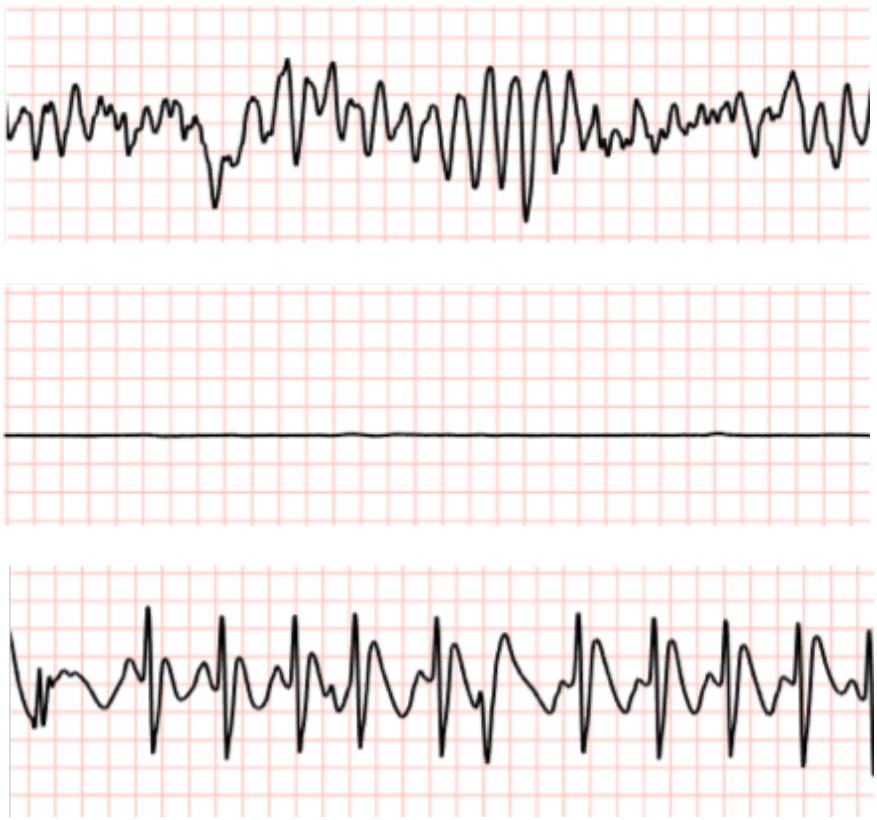

Figure 2.2: Examples of Major Heart Rhythms(from top to bottom): Ventricular Fibrillation (VF), Asystole, Organized Rhythm.

how well the heart is responding to the provided therapy. In order to assess heart status, the rescuer must pause CPR. In the case of organized rhythm, current protocols suggest that the rescuer manually assess pulse, or mechanical heart activity, by placed two fingers over the patient's carotid artery [70]. Presence of a pulse implies that the heart is able produce sufficient electrical and mechanical activity independently, resulting in the cessation of resuscitation therapy.

2.2.3 Tradeoff of the CPR Pause

While this pause in CPR can inform rescuers of the underlying heart status, it can break momentum in therapy, especially if the heart is responding well to it. As these pauses to assess rhythm can take up to 20 seconds, these interruptions in CPR to assess pulse can greatly decrease chance of patient survival [25,101]. In other hand, knowledge of cases where CPR is not beneficial to the patient can result in the early cessation of CPR and change in therapy, improving chances of survival. This tradeoff between the pausing to evaluate progress and impairing patient survival is one of the main issue in resuscitation practices. Therefore, real-time assessment of patient status *during CPR* can be an invaluable resource to the field of resuscitation and can potentially improve patient outcomes and survival.

However, implementing such practices is quite difficult as CPR artifact frequency range greatly overlaps that of vital heart activity. As a result, it can be difficult to parse out noise from signal. Eliminating these frequency bands with noise can efface crucial data. On the other hand, parsing out too little noise can result in too much measurement noise in the model, introducing potential bias the model.

2.3 Preprocessing Techniques

A number of filtration techniques have been used to eliminate the presence of CPR artifact. The goal of these filtration techniques is to not only remove this CPR artifact, but also remove extraneous noise caused by movement, powerline interference, and friction from defibrillator pads. A wide array of digital signal processing techiques have been implemented in this field ranging from small order Butterworth filters to Kalman filters [42]. We briefly discuss filtration techniques that are commonly used in the field of resuscitation.

A common filtration tool is the bandpass filter, or a notch filter. The bandpass filter operates by attenuating frequencies outside of a pre-specified frequency band. The order of the filter corresponds to how sharply these frequencies are attenuated. This type of filter requires no training and is computationally inexpensive, making it an ideal contender for preprocessing ECGs. However, vital heart activity typically occurs within the 1-40 Hz range and greatly overlaps CPR artifact which can occur below the 20 Hz cutoff [27, 43]. A filter with the passband of 20-40 Hz eliminates most of this artifact, but also eliminating crucial heart information. On the other hand, a passband of 1-40 Hz preserves vital heart activity, but still retains CPR artifact. To avoid removing important information, many studies [65] err on the side of caution and use the latter passband range.

Another digital signal processing technique used in this field is adaptive filtration. This type of filtration denotes a corrupted signal, $x(n)$ as a sum of pure signal, $s(n)$ and noise, $a(n)$:

$$x(n) = s(n) + a(n), \tag{2.1}$$

where n is a given time point in the ECG. The coefficients in this filter are optimized by using the least means squares method [42].

However, unlike the notch filter which simply needs a pre-specified frequency passband, this type of filtration method requires the training of the transfer function to parse out the corruption. As a result, this tool requires more computational energy and time, making it less ideal for real-time resuscitation methods.

Other pre-processing techniques in the field of resuscitation involve using Kalman filters and Independent Component Analysis (ICA) to parse out signal from noise [42]. However, in order to achieve high levels of accuracy, Kalman filters require large amounts of training data, which often isn't available. ICA can be used, but requires a pre-specified number of components, which may differ from patient to patient.Specifically, the heterogeneity of the data and lack of uniformity make Independent Component Analysis difficult to fit. Moreover, while these filtering techniques have the potential to be quite powerful and eliminate CPR artifact, these methods can also introduce imperceptible bias while doing so.

Thus, experts in this field often opt for the simple digital preprocessing tool, the bandpass filter [65].

Chapter 3

MACHINE LEARNING AND FEATURE ENGINEERING FOR PREDICTING PULSE PRESENCE DURING CHEST COMPRESSIONS

Adapted from: **Sashidhar, Diya**, Kwok, Heemun, Coult, Jason, Blackwood, Jennifer, Kudenchuk, Peter J., Bhandari, Shiv, Rea, Thomas D. and Kutz, J. Nathan 2021. "Machine learning and feature engineering for predicting pulse presence during chest compressions". R. Soc. open sci.8210566210566.

3.1 Introduction

Machine learning (ML) and artificial intelligence (AI) algorithms are transforming the scientific landscape [22, 44]. From self-driving cars and autonomous vehicles to digital twins and manufacturing, there are few scientific and engineering disciplines that have not been profoundly impacted by the rise of machine learning and AI methods. Medicine is no exception, with a significant growth of machine learning and AI methods developed for applications ranging from imaging [38, 99] to personalized medicine [53]. We specifically developed a novel ECG-based ML algorithm that demonstrates the potential to improve resuscitation by predicting presence of a spontaneous pulse without pausing cardiopulmonary resuscitation (CPR).

ECG data and AI/ML algorithms are ideally poised to better inform resuscitation efforts in patients with cardiac arrest. Hundreds of thousands of people suffer an out-of-hospital cardiac arrest (OHCA) each year [14]. Patients who suffer cardiac arrest require time-critical, life-saving interventions for successful resuscitation. These interventions include CPR, which is a combination of chest compressions, ventilations, and advanced airway management. Additional interventions, such as electrical defibrillation and drug administration, depend upon knowledge of the cardiac rhythm and the presence of a pulse. Ideally, the rhythm and presence of a mechanical pulse would be monitored continuously during

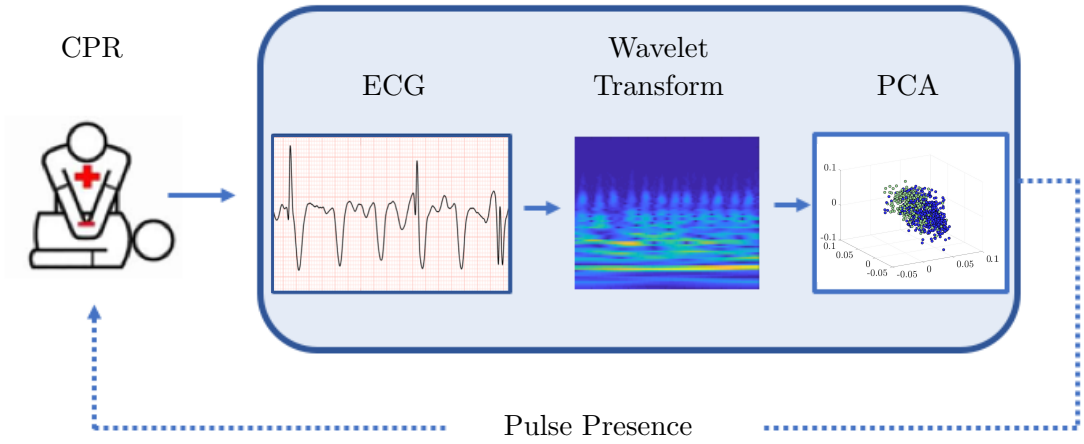

Figure 3.1: Overview of algorithmic architecture for real-time pulse presence classification. Time-series from ECG data is transformed to a feature space generated from wavelet analysis. Dominant correlated features in this space are used for assessment of pulse presence.

resuscitation to help guide treatment decisions. However, chest compressions preclude accurate assessment of rhythm or a *spontaneous* pulse, because compressions cause artifact in the ECG and can generate some measure of pulse themselves regardless of the underlying rhythm. Conversely, pausing CPR to assess these parameters comes at the cost of depriving the patient, if actually pulseless, of needed hemodynamic support and can adversely affect outcome. Although these pauses are intended to be brief in duration, interruptions in CPR to evaluate rhythm and pulse may take up to 20 seconds or longer, with increasing duration inversely associated with chances of survival [25, 101]. Thus, current guidelines reflect a balance between interrupting CPR to manually assess a patient's pulse presence versus performing continuous chest compressions in recommending an interruption in chest compressions for rhythm and pulse assessment only once every 2 minutes [60, 70]. The ability to detect a spontaneous pulse in real time during ongoing CPR would better inform care providers of the patient's clinical status throughout resuscitation and afford more

time-sensitive, better-informed treatment decisions to achieve improved clinical outcomes.

In the current study, we developed a scalable approach that integrates automated feature selection with various machine learning techniques (see Figure 3.1). This approach uses wavelet transforms followed by principal component analysis, allowing for the potential capture of subtle ECG morphologies while reducing feature dimensionality. Using only three dimensions of the reduced wavelet transform, we evaluated a diverse number of ML models, including deep learning architectures, that predict the pulse presence of organized rhythm segments both with and without CPR. We hypothesized that this approach would provide accurate detection of spontaneous pulse during ongoing CPR.

3.2 Methods

3.2.1 Description of Data

This retrospective observational study evaluated a convenience sample of 383 patients who experienced OHCA presenting with one of the following rhythms: ventricular fibrillation (VF), pulseless electrical activity (PEA), or asystole and underwent attempted resuscitation by emergency medical services (EMS) in King County, WA between 2006 and 2014 and had an available electronic defibrillator recording. While patients may have had an initial rhythm of VF, PEA, or asystole, all patients achieved an organized rhythm at some point. An organized rhythm is defined as discrete repetitive electrical complexes on the ECG (indicative of ventricular depolarization) at a rate that would be expected to produce a discernible pulse, but may or may not do so. Only organized clips were used to train and test the algorithm in study. EMS protocols followed American Heart Association guidelines and included the provision of manual CPR [70]. The defibrillator recording from each case included continuous measurement of the ECG and thoracic impedance, and an audio channel containing EMS narration of events during the course of resuscitation. Each defibrillator recording was reviewed by experts to identify two-minute rhythm/pulse checks from the impedance signal and to annotate the rhythm at each check. [67], [26] , [58] Organized rhythms were further categorized as either mechanical pulse present or absent (i.e., pulseless electrical activity) by reviewing the audio recording, as EMS rescuers verbalize the results

of manual pulse assessment and confirm any positive pulse assessment by blood pressure measurement per protocol. [65]. The study was reviewed and approved by the Review Committees of the University of Washington and Public Health – Seattle & King County.

Patients were randomized into training (60% of patients) and test (40% of patients) groups for algorithm development. Note that we have randomized the data on a patient level in order to test generalizability to patients in a withheld test set. We developed and evaluated the algorithm on adjacent ECG segments at each pulse assessment with an organized rhythm. Specifically, for each pulse check, a 10-second ECG segment was collected during the period of ongoing CPR just prior to the pulse check, immediately followed by a 5-second adjacent segment was collected without CPR that was collected during the pulse check. Figure 3.2 shows the 10-second segment with CPR along with the 5 second segment without CPR after filtration with the 4th order Butterworth filter was applied. Note that even after this filtration process, the segment with CPR is still quite artifacted and looks very different from the segment without CPR. We assumed pulse presence was unlikely to have changed during this very short interval. The number of pulse checks depended upon the course of resuscitation and was therefore variable across patients. For example, a patient who achieved ROSC quickly may have had only one pulse check, while another patient who did not ultimately survive may have had many pulse checks. To prevent the overrepresentation of data from any single patient, we set a maximum of three randomly selected 15-second periods from each category (*Pulse* or *Pulseless*) per patient. Table 3.3 also shows the median number of samples (including pulse and pulseless) for each patient for training and test sets respectively.

3.2.2 Data Preprocessing and Noise Removal

Data were collected from MRx and Forerunner 3 (Philips Healthcare, Bothell, WA), and Lifepak 12 and Lifepak 15 (Physio-Control, Redmond, WA) automated biphasic defibrillators, with ECG sampling rates ranging from 125 to 250 Hz. We homogenized all ECG data to a sampling rate of 250 Hz. To ensure that the underlying method would be applicable across this range in sampling rates, all data was low-pass filtered at 40 Hz after resampling

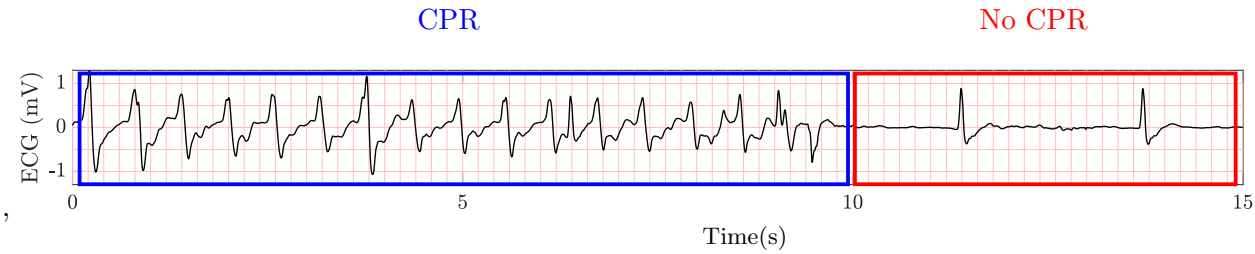

Figure 3.2: ECG after using 4-tap Butterworth filter process.A 10s segment during CPR is highlighted in blue (left), and a 5-s segment during CPR pause for pulse check is highlighted in red (right).

all data to a common sampling rate of 250 Hz. Since the lowest original sampling rate is 125 Hz (with a Nyquist frequency of 62.5Hz), we assume that the 40 Hz cutoff frequency assures the same maximum frequency content across all devices, eliminating the possibility that the algorithm would utilize frequencies above 40 Hz that may not be as well-resolved in devices with lower sampling rates. We applied filtering to remove high-frequency electrical noise and low-frequency drift using a fourth-order Butterworth bandpass filter from 1-40 Hz implemented with a forwards-backwards implementation to preserve linear phase. Of note, this filter was not designed to remove artifact from CPR, which is concentrated primarily at the frequency of manual chest compressions (2 Hz) but can exhibit harmonic artifact (e.g. up to 20 Hz) that can overlap with underlying heart activity [27, 43]. Other studies have attempted to reduce CPR artifact using adaptive filters, Kalman filters, and notch filters [6, 43].

In contrast, rather than attempt to remove CPR artifact via filtering as in prior studies [43] , the current method employs a combination of time-frequency analysis and machine learning to allow data-driven exclusion of frequencies confounded by CPR artifact.

The heart rate was estimated using a higher order (8th order) Butterworth bandpass filter in order to emphasize QRS complexes. This order was chose in order to emphasize sharp peaks in the ECG clip and reduce the presence of smaller waves such as P and T waves as well as artifact. Since these complexes are correlated with heart rate, we multiplied the number of QRS complexes in a 5 or 10 second clip by factors 12 and 6 respectively to get the heart rate.

3.2.3 Algorithm Development

Wavelet Transforms

We used wavelets as a decomposition method for analysis and representation of time-frequency signals [28,72]. As the mathematical structure for constructing the multi-resolution analysis of data such as the ECG time series or spatial data such as images [63,72], wavelets are foundational in the representation of data that have multiscale spatiotemporal features. Leading image compression technologies such as JPEG 2000 are wavelet-based encoding schemes which leverage multi-resolution analysis to sparsely represent important image features for the compression process [69]. Importantly, wavelets provide a recursive architecture which extracts dominant time-frequency features within prescribed time-bins, much like the windowed Fourier transform or Gabor transform [63].

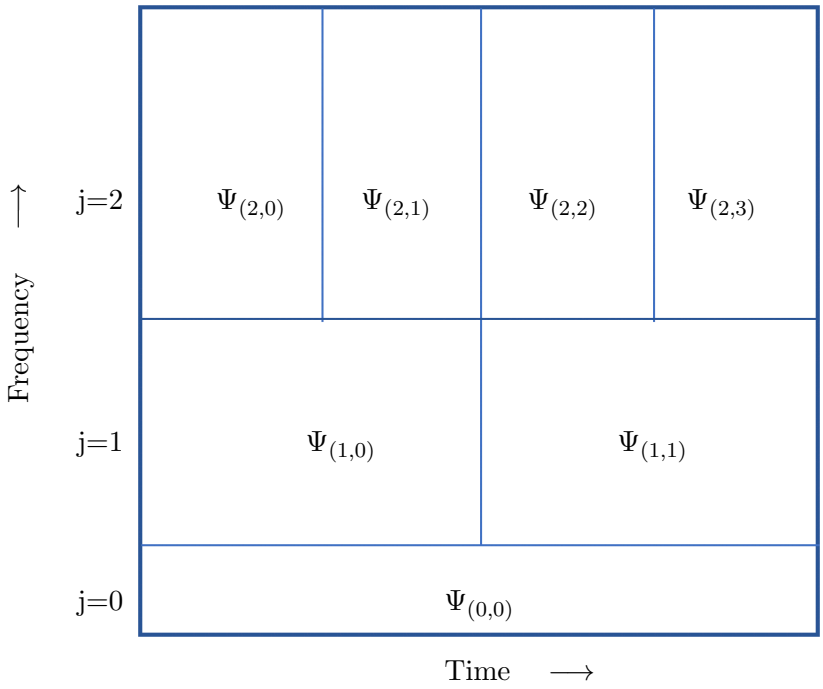

Figure 3.3: Wavelet Transform template.

In the current investigation, the wavelet transform was used to analyze the time-frequency structure of the filtered ECG by constructing the ubiquitous scalogram, which is equivalent

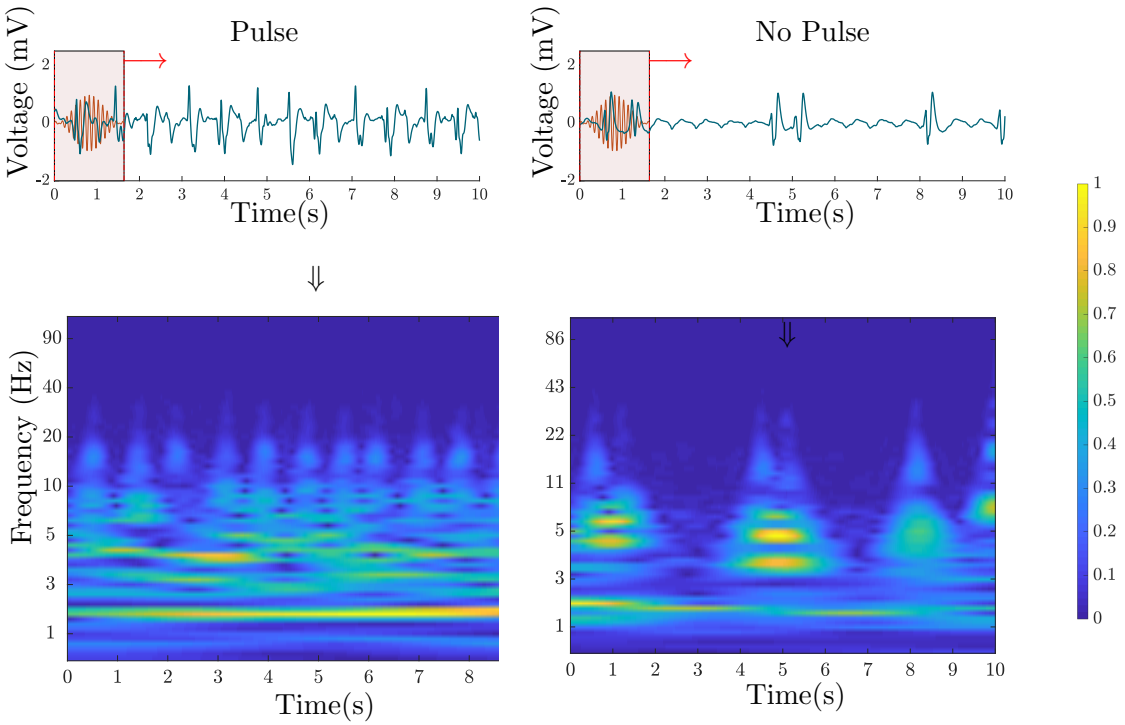

Figure 3.4: Top Left: Application of bump wavelet to segment of ECG (during CPR) with ROSC label, Top Right: Application of bump wavelet to segment of ECG (during CPR) with No Pulse label, Bottom Left: Scalogram generated by convolving various scales of bump wavelet with respective ECG segment with ROSC label, Bottom Right: Scalogram generated by convolving various scales of bump wavelet with respective ECG segment with ROSC label

to the spectrogram created using Gabor transforms [63]. This scalogram provided a visual representation of the time and frequency content of the ECG data and a critical feature space to extract meaningful portions of the signal that may predict pulse.

The wavelet transform was computed by convolving a wavelet function with the time-series recording from the ECG. The wavelet window, whose temporal width is recursively cut in half (or doubled) hierarchically, was translated across the entire ECG time series as its time-frequency information was extracted (Figure 3.3). The wavelet is represented by the functional form

$$\Psi_{j,k}(t) = \frac{1}{a_j}\Psi\left(\frac{t-b_k}{a_j}\right) \tag{3.1}$$

where a_j denotes the wavelet dilation parameter, b_k is the center position of the wavelet window, t is the current time point, and Ψ is the mother wavelet. The parameters b_k is chosen so that the wavelet window translates across the entire time history of the ECG data segment (Figure 3.5). The wavelet transform convolves (1) with the ECG signal f(t) and gives a new representation that is parametrized by both a_j and b_k:

$$W_\Psi(f)(a_j, b_k) = \int_0^T f(t)\bar{\Psi}_{j,k}(t)dt \tag{3.2}$$

where the ECG signal is recorded for $t \in [0, T]$. The value of $W_\Psi(f)(a_j, b_k)$ for each a_j and b_k gives the spectral content, or energy, at a specific time-frequency location. By plotting the entire time-frequency plane, the scalogram is constructed from equation (2). Specifically, the energy of a signal (E) in a window corresponding to dilation parameter a and centered at b is given by

$$E(a, b) = |W_\Psi(f)(a, b)|^2. \tag{3.3}$$

To calculate the scalogram, we integrated these values across all discrete values of a_j and b_k used when discretizing the time-series signal.

Since wavelets can extract localized features in the time-frequency domain, we applied the wavelet transforms to the detrended ECG time-series measurements (see Figure 3.4).

Depending on the wavelet type and sifting window size, the wavelet transform can capture both the morphology of sharp peaks such as the QRS wave as well as the nuances of the P wave. The choice of the wavelet depends on the nature of the signal. In order to assess the ideal wavelet transform for feature extraction, we used a subset of the dataset with 120 ECG strips (96 training) with the pulse label and 120 (96 training) ECG strips with the pulseless label. For each of these wavelets, we looked at various combinations of the modes and discriminants and generated the output of the combination with the highest AUC using this subset of the training data. The morphologies of these wavelets enable

Wavelet	Modes	Discriminant	AUC (CPR)	Modes	Discriminant	AUC (no CPR)
coif3	1-3	LDA	75%	2-3	QDA	77%
mexhat	1-3	LDA	77%	2-3	QDA	75%
db6	1-6	LDA	77%	1-6	LDA	79%
bump	1-3	LDA	81%	2-3	LDA	81%
morlet	2,4	LDA	75%	2-3	LDA	79%
morse	1-3	LDA	75%	1-3	LDA	77%
rbiorth3.3	1-5	LDA	79%	1-5	LDA	81%
db1	2-4	LDA	73%	1-2	QDA	75%
db2	1-3	LDA	77%	1-6	QDA	75%
fk22	1-6	LDA	77%	1-3	LDA	81%
dmey	1-5	QDA	75%	1-2	LDA	83%
sym2	1-3	LDA	77%	1-2	QDA	75%
bior1-1	2-5	LDA	75%	1-2	QDA	75%
coif1	1-2	LDA	75%	1-6	QDA	73%

Table 3.1: Combinations of modes and discriminants with highest AUC for each type of wavelet

their corresponding transform to capture and emphasize various morphologies found in the electrocardiogram when convolved with the time series. Table 3.1 shows the top 14 combinations of wavelet, modes, and discriminants. The following combinations were found using a grid search with various combinations of the first ten modes, mother wavelets, and discriminants. For simplicity, these discrimants included linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and support vector machine (SVM).

Out of 14 different wavelets and combinations of modes and discriminants, the bump wavelet had the highest model performance (Table **??**) in both CPR and non-CPR segments. While there were mother wavelets with comparable performance such as the Discrete Meyer (dmey) wavelet and the Biorthogonal wavelet (rbiorth3.3), the bump wavelet achieves the

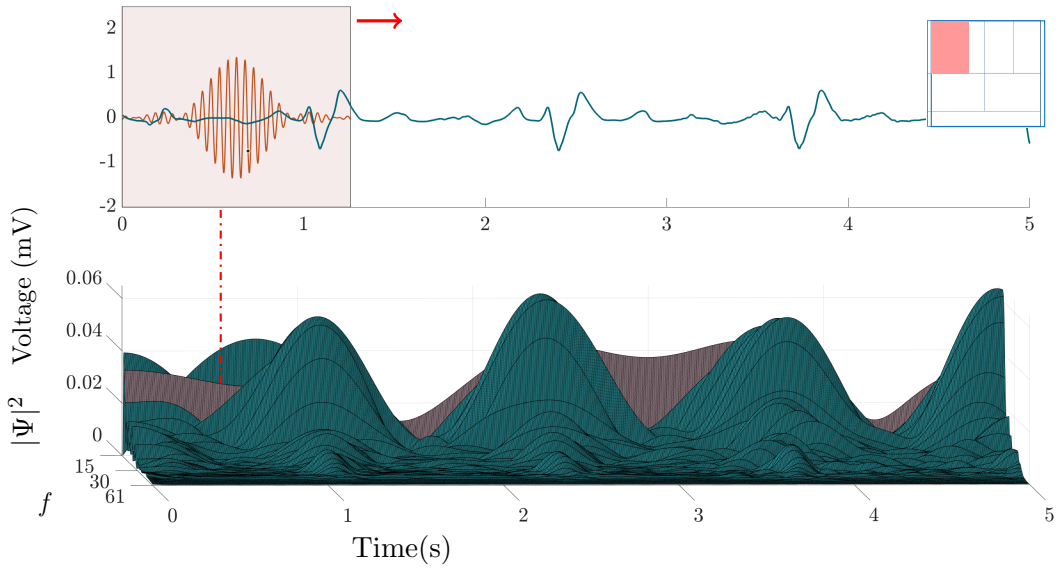

Figure 3.5: Scalogram is calculated by convolving the bump wavelet with the filtered ECG as the window translates across the time series. The highlighted region in the bottom panel corresponds to product after convolving the ECG (blue line) in the shaded red box with the mother wavelet (orange line)

highest performance across all wavelets during CPR while accomplishing similar accuracy without CPR. In addition, since ECGs are periodic and have oscillatory bursts as seen in QRS complexes, it makes sense that the bump wavelet was ideal for the extraction of our time-frequency features.

The oscillatory shape of bump wavelet enables it to capture both the morphology and sharpness of the QRS wave and the dome-like structures of the P and T waves. Importantly, the wavelet transform can produce scalograms which will be used as the feature space for identifying the pulse presence of the heart.

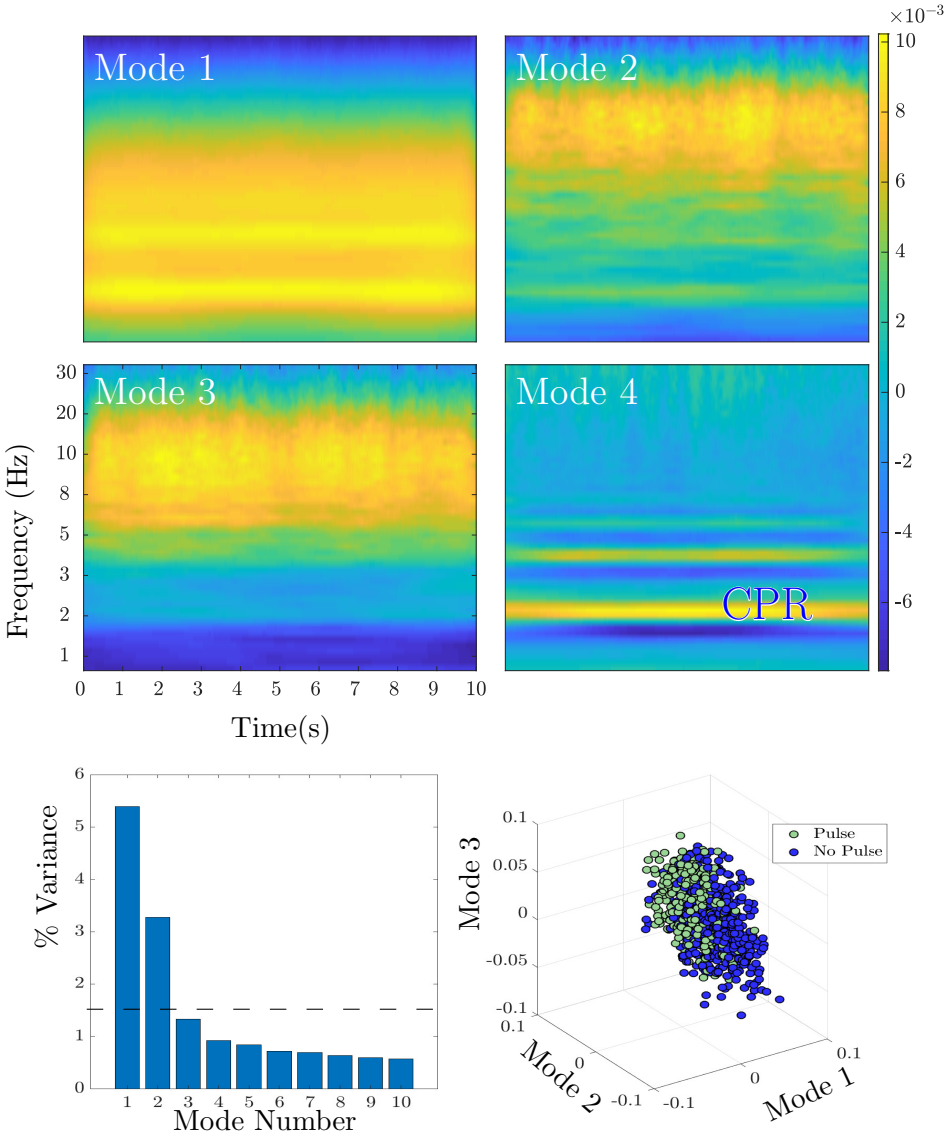

Figure 3.6: Top: First four spatial modes of the system of reshaped scalograms (during CPR). Bottom Left: Normalized Singular Values of the system of reshaped scalograms. The dottle line represents the variance cutoff. Bottom Right: Various combinations of temporal modes of system of reshaped scalograms.

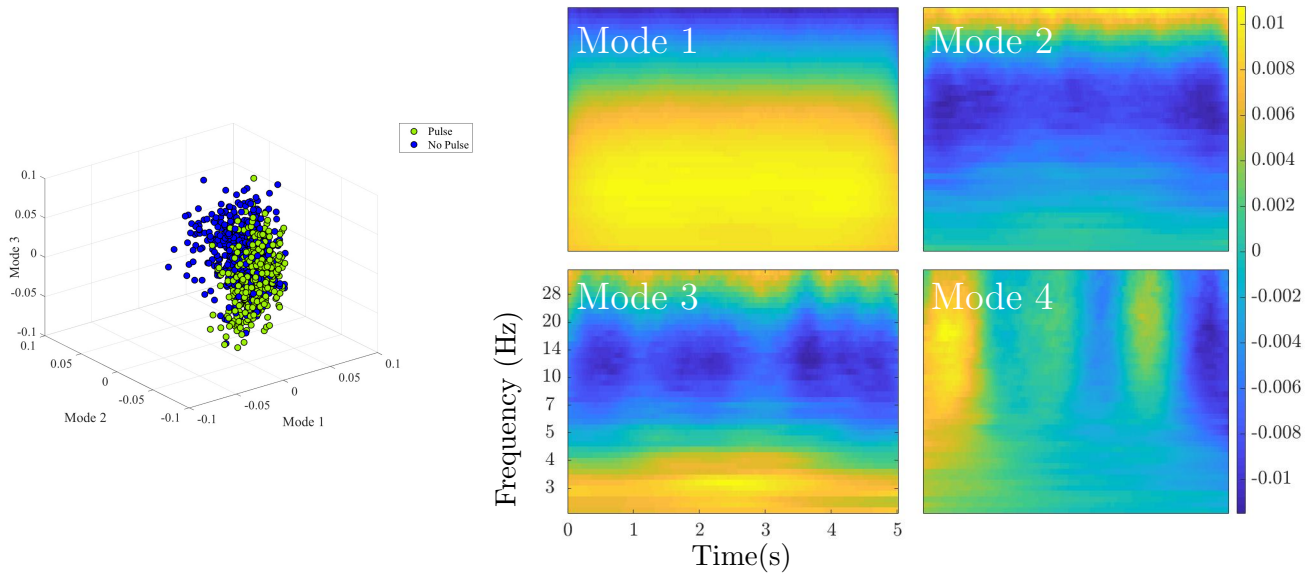

Figure 3.7: Temporal and Spatial Modes for PCA of reshaped scalograms (without CPR).

Principal Component Analysis

The scalograms produced using the wavelet transforms were typically high-dimensional (pixel space) images, making it difficult to both process and directly identify key signatures in the data. We used *Principal Component Analysis* (PCA) to reduce the dimensionality of the data and to identify dominant features in the scalograms. To help illustrate this concept, we first introduce the *Singular Value Decomposition* (SVD) on which PCA is based.

Suppose there is a matrix $A \in \mathbb{C}^{m \times n}$ with n columns of vectorized scalograms (a_i for $i = 1...n$) of size m, where $m >> n$. The SVD can be used to decompose this single matrix into a product of three matrices.

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ a_1 & \dots & a_n \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ 0 & \ddots & 0 \\ \dots & 0 & \sigma_n \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ v_1 & \dots & v_n \\ \vdots & \vdots & \vdots \end{bmatrix}^T$$

where $\mathbf{U} \in \mathbb{C}^{m \times n}$ is a unitary matrix, $\mathbf{\Sigma} \in \mathbb{C}^{m \times n}$ is a diagonal matrix, and $\mathbf{V} \in \mathbb{C}^{n \times n}$ is a unitary matrix.

The columns of $\mathbf{U}$ and $\mathbf{V}$ are known as the left and right singular vectors, respectively. The columns can be thought of as "modes" of the system, where each mode represents a dominant feature in the data. Thus, this decomposition has the ability to rank and analyze each feature produced by correlations in the data. Specifically, the left singular vectors, or the spatial modes when reshaped into matrices, show specific regions in the scalogram that exhibit the most variance. The right singular vectors, or the temporal modes, show how each scalogram projects onto these modes. Thus the SVD ultimately provides a feature engineering tool which allows for clustering and classification tasks.

The matrix Σ is comprised of descending diagonal elements called singular values. These singular values, which are associated with the columns of $\mathbf{U}$ and $\mathbf{V}$, capture the variance of the data in each principal direction. In other words, the SVD creates a hierarchy of dominant features and computes mode "weights" using singular values.

This hierarchy helps create a low-rank approximation of the data, $\tilde{A}$. In other words, the first r modes can produce the best low-rank feature representation of the data in a least-squares sense.

$$\tilde{A}_{m \times r} \approx \tilde{\mathbf{U}}_{m \times r} \tilde{\mathbf{\Sigma}}_{r \times r} \tilde{\mathbf{V}}^T_{r \times r} \tag{3.4}$$

This gives a significantly reduced matrix that is easier to manipulate and interpret. This concept of SVD is the foundation of PCA. Specifically, PCA involves taking the SVD of matrix A with normalized and centered data columns [63].

We considered a maximum of 10 PCA modes for inclusion as classification features, and the choice of r involved a balance of accuracy and simplicity. As there was only a marginal increase in performance by the successive addition of modes, we chose the Pareto optimal solution by picking the more parsimonious model [63]. Specifically, to remove extraneous signal in the data while preserving model simplicity, we imposed a percent variance cutoff of 1% as a criteria for selection of modes to include as features 3.6). We show the first four spatial modes taken from the reduced U matrix in Figure 3.6. From these four spatial

modes we can see which frequency bands high variance over time. Specifically, the 2-10 Hz frequency bands have high variance in Mode 1 while the 10-20 Hz bands have high variance in Mode 2 over the 10 second interval. While only the first three modes make the 1% variance cutoff, we show the fourth mode as it is able to capture the frequency of CPR (1.6 to 2 Hz) as well as its harmonics at 3 Hz and 5.5 Hz. As there is CPR in both pulse and pulseless segments, it is not surprising that this CPR band is of minimal variance. Note that only three modes are used in the analysis as cross-validation suggests these to be generalizable without over-fitting. The slow decay of the singular value spectrum is often handled by thresholding techniques which separate low-rank signal from noise [41]. The advantage of the three modes is also associated with interpretability and the visualization capabilities that three dimensions afford. Specifically, the figures presented show a clear pattern of clustering without producing high-dimensional space which is beyond our capabilities to visualize. The modal patterns associated with the three modes also allow for an interpretation of the time-frequency signatures that dictate our ability to comprehend the ECG time-series recordings. In any case, cross-validation alone suggest that the three modes are appropriate to use for the clustering analysis that follows.

The first 3 modes each had greater than than 1% variance, and together captured much of the variance of the system (see Figure 3.6). The distributions of modes 1-3 in the ECG segments with and without CPR, stratified by pulse presence, (see Figure 3.8) provided insight into their usefulness as features. In the CPR histogram, Modes 1 and 2 had slightly-overlapping, yet distinct distributions. While the distributions in Mode 3 showed more overlap, there were still slightly separated, indicating that the addition of Mode 3 possibly provided information that was not provided in Modes 1 and 2. Similarly, the model without CPR had distinct distributions for Modes 1 and 2 with minimal overlap. The third mode also showed more overlap between the pulse and pulseless distributions. Together, these histograms show that in both models, the combination of all three modes prove to be more informative than each individual mode.

Similarly, Figure 3.7 shows distinct clustering of temporal modes with and without pulse. Similar to Figure 3.6, there is high variance between the 0 to 10 Hz band. Interestingly, the second and third modes are drastically different, with high variance in the 20 to 30 Hz band

and 0 to 3 Hz band, respectively. Unsurprisingly, the 1.5, 3, and 5 Hz frequency bands have little variance as CPR is not administered on these samples.

Clustering and Classification

In order to classify ECG segments as pulse present or pulse absent, we considered the three modes derived from the PCA analysis, as well as a fourth variable, heart rate (in beats/min), as candidate features. Heart rate is a variable with accepted clinical importance, and to estimate heart rate, we first preprocessed the ECG with a higher-order (8th order) Butterworth filter and a passband of 10-40 Hz to emphasize QRS complexes. We then applied a simple peak-finding algorithm to calculate heart rate from QRS complex locations observed within the ECG segment.

Using the training data, we explored a variety of classification models. In projecting the three PCA modes onto a three dimensional space, we observed two distinct clusters: one with *Pulse* labels, the other with *No Pulse* labels. The formation of clusters in a low-dimensional feature space suggested amenability to discriminant analysis for which a number of classification models were suitable: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and support vector machine (SVM) models. SVM model parameters were hand-tuned using a grid search [15, 22]. These models fall under the broader aegis of supervised machine learning and are critical in automating data-driven discovery processes.

In addition to the above discriminant models, we also explored Neural Networks (NN), Logistic Regression (LR), Gaussian Mixture Models (GMM), and Random Forests (RF) with the aformentioned modes and heart rate as features. As we did not have enough samples for a deep Neural Network, we used a two-layer neural network to avoid overfitting, as there is much variability across patients. The number of trees in the random forest algorithm was optimized using out-of-bag (OOB) error on the training set. Similar to finding the optimal number of modes in the discriminant analyses (LDA and QDA), the number of modes used for the Gaussian Mixture Model was optimized using a grid search on the training data with AUC as a metric of performance. It was found that 3 modes was the optimal number

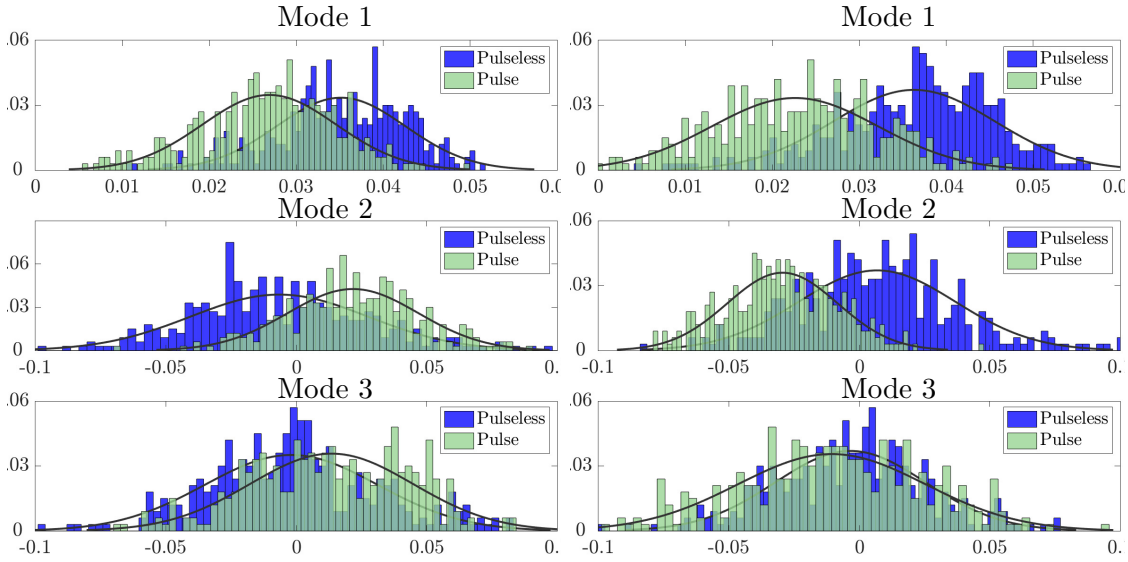

Figure 3.8: Histograms of Modes 1,2,and 3 for each label with (left) and without CPR (right) using the training data. Green histogram corresponds to ECG segments with a pulse label and Blue histogram corresponds to ECG segments with a No pulse label.

of modes for this model. The logistic regression model used the values from the three PCA modes as dependent variables with zero interaction terms. Finally, as an alternative to classification involving the PCA modes, we used scalogram matrices as input images for a Convolutional Neural Network (ConvNN). Convolutional Neural Networks are used heavily in image recognition and can sometimes detect nuances in images that are imperceptible to the human eye [63]. However, many more scalograms are needed for this model to have comparable performance.

Separate classification models were fit for ECG segments with CPR and those without CPR. To select the optimal model, we assessed the performance of candidate models on training data by calculating the area under receiver operating characteristic curve (AUC) with a 5-fold cross validation procedure.

Assessment of Classification Performance

Based on the training performance, the best model was then tested on the external validation (test) set of ECG segments with and without CPR separately. Performance was characterized by AUC values, whose 95% confidence intervals were calculated using bootstrapping [81], and sensitivity and specificity for pulse detection. As there are not any clinically defined standards for sensitivity and specificity, we selected these values using the Optimal Operating Point [46]. In addition, we compared this method to a previous algorithm which also classified organized rhythm segments during CPR. [65] That method used a logistic combination of three manually-designed features: a measure of QRS rate and two measures of QRS amplitude and morphology (median magnitudes within two frequency bands centered at 6.25 and 25 Hz). AUCs of the two methods were compared with the Delong method for correlated data [31].

3.3 Results

Demographic, clinical and outcome characteristics of the study cohort are provided in Table I. There were 230 patients with 540 segment pairs (with and without CPR) in the training set and 153 patients with 372 segment pairs in the test set. The percentage of *Pulse* vs. *Pulseless* segments were similar in the training (39% *Pulse*) and test (37% *Pulse*) sets.

We observed that the classifiers had comparable performance on the training data with and without CPR (Table 3.2). While these models have similar AUC performance, they vary in complexity. Complex models like Neural Networks and Convolutional Neural Networks have a tendency to overfit and are less ideal due to the relatively small amount of data. As significantly more data is collected (specifically, by several orders of magnitude), it is expected that deep learning would provide a method to advance the state-of-the-art beyond the simpler regression methods presented. In order to reduce the risk of overfitting, we selected the simplest model, LDA, as the final model, which had an AUC of 79% during CPR and 87% without CPR (Table 3.4). When considering the effect of estimated heart rate as a feature, the LDA model had a 0.79 AUC with confidence intervals (0.76, 0.83) without estimated heart rate and 0.80 AUC with confidence intervals (0.76, 0.84) with estimated

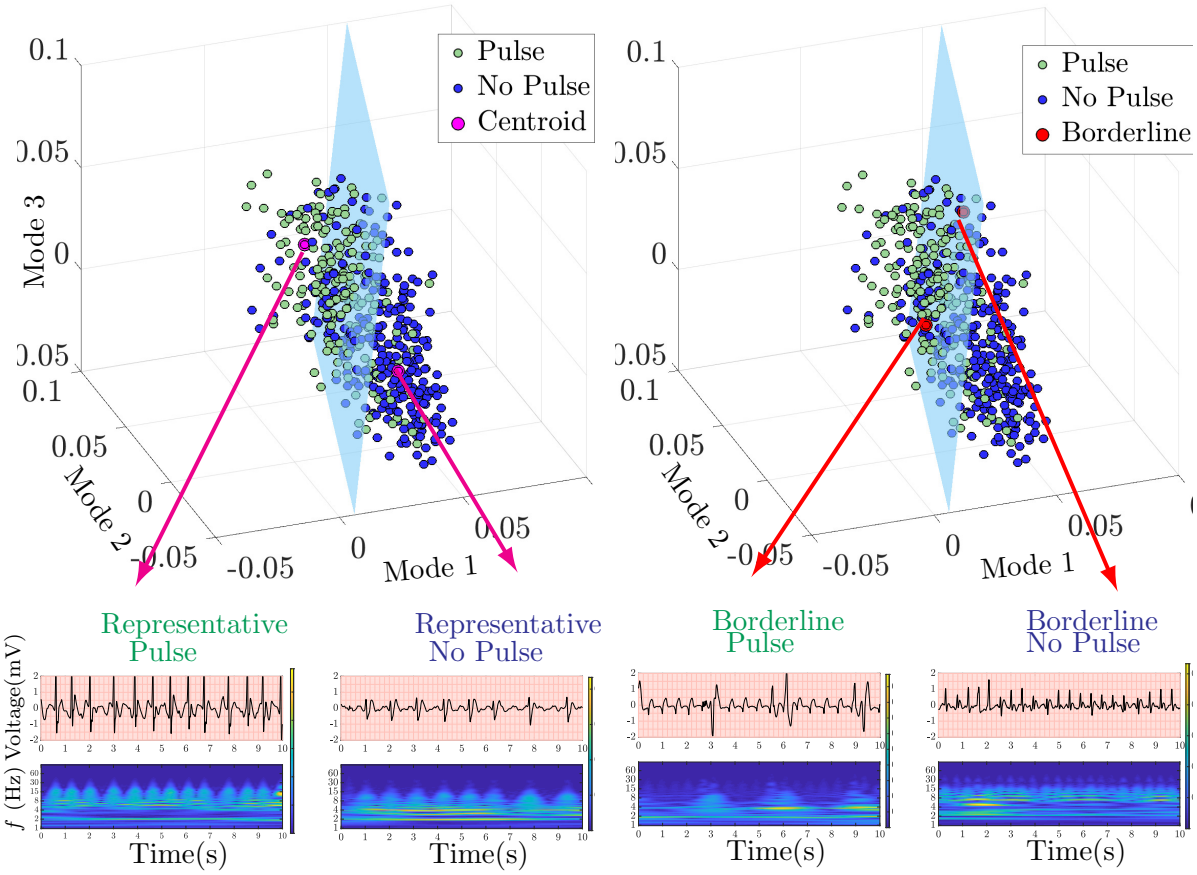

Figure 3.9: Left: Centroids of Cluster for Pulse and No Pulse labels respectively, Right: Borderline points for Pulse and No Pulse respectively

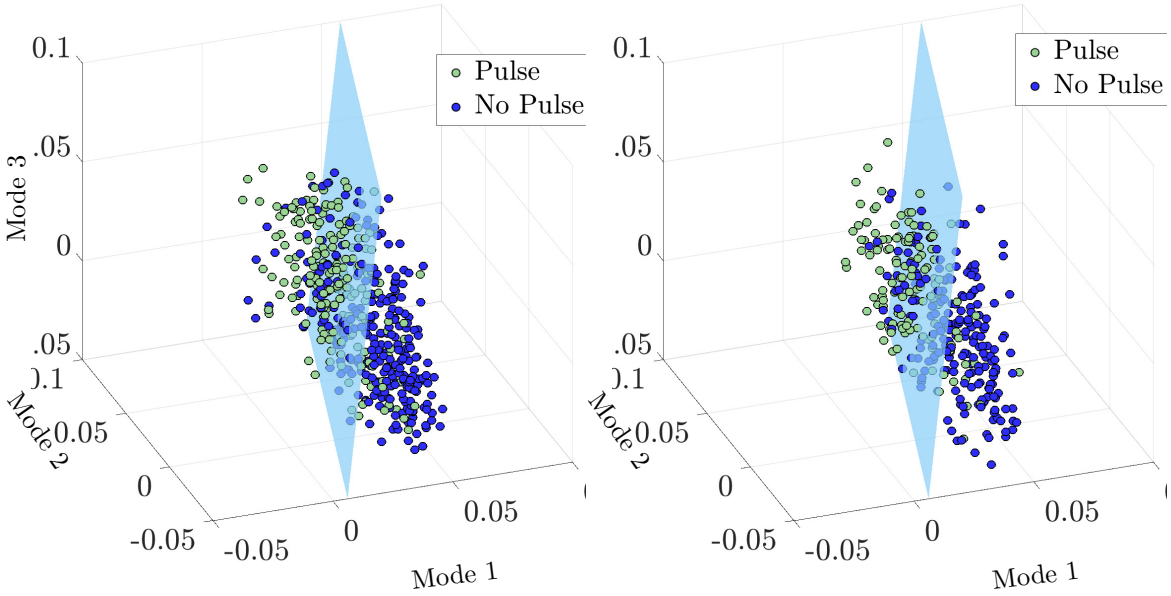

Figure 3.10: Left: Labeled training set with hyperplane. Hyperplanes correspond to the discriminant generated after applying Linear Discriminant Analysis to PCA modes 1 - 3 using training data (during CPR). Right: Labeled test set with the same hyperplane generated during training. Green cluster corresponds to transformed ECG segments with a Pulse label while blue cluster corresponds to transformed ECG segments with a No Pulse label.

	CPR		No CPR	
Classifier	Modes 1-3	Modes 1-3, HR	Modes 1-3	Modes 1-3, HR
LDA	0.79 (0.76,0.83)	0.80 (0.76,0.84)	0.87 (0.84,0.90)	0.87 (0.84,0.90)
QDA	0.80 (0.77,0.84)	0.81 (0.78,0.85)	0.87 (0.84,0.93)	0.88 (0.85,0.91)
SVM	0.79 (0.75,0.83)	0.77 (0.73,0.81)	0.87 (0.84,0.90)	0.87 (0.84,0.90)
GMM	0.77 (0.73,0.81)	0.67 (0.63,0.72)	0.85 (0.82,0.88)	0.86 (0.83,0.89)
LR	0.79 (0.75, 0.83)	0.80 (0.76, 0.83)	0.86 (0.84,0.89)	0.87 (0.84, 0.90)
RF	0.80 (0.76, 0.83)	0.81 (0.76,0.83)	0.87 (0.84,0.90)	0.87 (0.84, 0.90)
NN	0.80 (0.76,0.83)	0.83 (0.79,0.86)	0.87 (0.84,0.89)	0.89 (0.86,0.91)
ConvNN	0.78 (0.75,0.81)	–	0.77 (0.74,0.81)	–

Table 3.2: Comparison of Training AUC values and 95% confidence intervals for various models with and without heartrate (HR), where LDA : Linear Discriminant Analysis, QDA :Quadratic Discriminant Analysis, SVM: Support Vector Machine, GMM: Gaussian Mixture Model, LR: Logistic Regression, RF: Random Forest, NN: Neural Network, ConvNN: Convolutional Neural Network.

heart rate. In the model without CPR, performance was exactly the same, with 0.87 AUC and confidence intervals (0.84, 0.90). Furthermore, we observed that the addition of heart rate to PCA modes 1-3 in the LDA model did not significantly improve AUC, and therefore, the final model incorporates PCA modes 1-3 as the sole features (Table 3.2). In some cases such as the Gaussian Mixture Model, performance actually decreased with the addition of this feature.

AUC to discriminate between Pulse and Pulseless segments was 0.84 (95% CI (0.80, 0.88)) with CPR and 0.89 (95% CI (0.86,0.92)) without CPR on test data. The sensitivity and specificity was 0.68 (95% CI (0.58, 0.77)) and 0.87 (95% CI (0.79, 0.93)) with CPR and 0.76 (95% CI (0.66, 0.84)) and 0.85 (95% CI (0.77, 0.91)) without CPR (Table IV).

When tested on the same dataset, this algorithm had marginally higher AUC values than the previous algorithm [65], but the AUC difference was not statistically significant ($p = 0.44$) during CPR. This algorithm performed significantly better than the previous algorithm without CPR ($p = 0.005$).

Patient Characteristics	Training (N=230)	Test (N=153)	All (N=383)	Significance
Initial Rhythm				
VF	192(83.5)	128(83.7)	320(83.6)	0.96
Asystole	11(4.7)	5 (2.2)	16(4.2)	0.47
PEA	26(11.3)	20(13.1)	46(12.0)	0.6
Indeterminate	1(0.4)	0(0.0)	1(0.2)	*
Female, n (%)	63(27.4)	42(27.5)	105(27.4)	0.72
Age, median (IQR)	64(53, 74.5)	62(50.5, 73)	63(52, 74)	0.62
Cardiac etiology, n(%)	201(87.4)	128(83.7)	329(85.9)	0.37
Location, n(%)				
Home	155(67.4)	99(64.7)	254(66.3)	0.53
Public	61(26.5)	43(28.1)	104(27.2)	0.84
Nursing Home	13(5.7)	12(7.8)	25(6.5)	0.4
Arrest before EMS arrival, n(%)	215(92.6)	141(92.2)	356(93)	0.73
Witnessed, n(%)	164(71.3)	100(65.4)	264(68.9)	0.19
Bystander CPR, n(%)	151(65.7)	97(63.4)	248(64.8)	0.52
EMS Response (minutes), median (IQR)	5.1(4.3, 6.9)	5.2(4.4, 6.8)	5.2(4.4, 6.9)	0.47
Total shocks, median (IQR)	3(1, 7)	2(1, 3.5)	2(1, 5)	0.06
ROSC, n(%)	156(67.8)	109(71.2)	265(69.2)	0.58
Admit to hospital, n(%)	164(71.3)	110(71.9)	274(71.5)	0.42
Survive to discharge, n(%)	108(47)	69(45.1)	177(46.2)	0.51

Table 3.3: Patient Demographics. VF: Ventricular Fibrillation, PEA: Pulseless Electrical Activity. Note: Significance was calculated using Mann-Whitney U test for continuous variables and chi-square for proportions

	Classifier	CPR	No CPR
Training			
	LDA	0.79 (0.76,0.83)	0.87 (0.84,0.90)
	QDA	0.80 (0.77,0.84)	0.87 (0.84,0.93)
	SVM	0.79 (0.75,0.83)	0.87 (0.84,0.90)
	GMM	0.77 (0.73,0.81)	0.85 (0.82,0.88)
	LR	0.79 (0.75, 0.83)	0.86 (0.84,0.89)
	RF	0.80 (0.76, 0.83)	0.87 (0.84,0.90)
	NN	0.80 (0.76,0.83)	0.87 (0.84,0.89)
	ConvNN	0.78 (0.75,0.81)	0.77 (0.74,0.81)
Test			
	LDA	0.84 (0.80,0.88)	0.89 (0.86,0.92)

Table 3.4: Comparison of Training AUC values and 95% confidence intervals for candidate models and test performance for final model, where LDA : Linear Discriminant Analysis, QDA :Quadratic Discriminant Analysis, SVM: Support Vector Machine, GMM: Gaussian Mixture Model, LR: Logistic Regression, RF: Random Forest, NN: Neural Network, ConvNN: Convolutional Neural Network.

3.4 Discussion

An understanding of the ECG rhythm and its physiological consequences provides the basis for using ECG to predict pulse. The most common rhythms observed during cardiac arrest are ventricular fibrillation, asystole, and organized rhythms. Ventricular fibrillation is characterized by chaotic, disorganized electrical activity on the ECG; and asystole by the absence of any electrical activity. Neither produces a pulse and their presence on the ECG obviates the need for a pulse check. Organized rhythms are defined by coordinated electrical activity on the ECG, but may or may not be associated with cardiac muscle contraction and a pulse (the latter commonly referred to as pulseless electrical activity). Thus when observed on the ECG, organized rhythms require a pulse assessment as to the continued need for CPR and resuscitative efforts.

A variety of modalities to detect spontaneous circulation during resuscitation have been investigated, but at present, automated tools to reliably identify a spontaneous pulse during CPR are not available [78]. The study of photoplethysmography is limited to pre-clinical data and pilot studies [100], and end-tidal carbon dioxide monitoring can be confounded by other interventions. Ultrasound is generally not available in the pre-hospital setting and can be operator dependent. ECG and thoracic impedance monitoring are universally available on defibrillators, and a few prior methods to automatically detect pulse have been reported [34, 35, 71, 83]. However, those methods were not designed for real-time classification of pulse presence during CPR. In recent work, we developed an algorithm which predicts pulse presence in real time from the ECG during CPR among patients with underlying organized rhythms [65]. Ultimately, the optimal method to real-time assessment of a patient's physiological status and in turn the best choice of therapy may involve the integration of multiple technology-based measures. The ECG may be considered foundational given its universal use and fundamental role in resuscitation.

Although the method described in the current investigation and the recent prior method [65] performed similarly during CPR, the current method integrates machine learning and signal processing techniques (see Fig 3.1) and presents a more scalable alternative to the recent prior method for pulse detection during CPR [65]. A limitation of the previous

approach [65] was that it did not provide a framework to model more subtle morphologic characteristics nor to discover new feature spaces. In contrast, the components of the algorithm developed in the current method are modular and flexible; thus, new feature spaces and clustering techniques can be readily integrated and explored within the algorithmic structure proposed. As larger quantities of data become available, data-based approaches such as deep neural networks and more robust convolutional neural networks could easily be integrated into the current method to further-improve performance. Importantly, the current algorithm can be applied in a real-time manner during resuscitation, as the signal processing and feature extraction steps are not computationally intensive. Specifically, wavelet transforms may be implemented using FFTs which modern embedded hardware is well-optimized to compute the transformations required. Computation time of PCA greatly relies on the length of the clip and the number of samples, both of which are relatively small. PCA is O(mn^2+n^3) [96], where m is the length of the clip (2500 for CPR, 1250 for no CPR) and n is the number of samples used. Given that we have small clips and relatively small sample sizes (compared to those used for NN), this method has the potential to be done in real time. In addition, the LDA model has already been trained, so features simply need to be projected on the LDA subspace to classify the rhythm. The LDA subspace with the corresponding hyperplane is portrayed in Figure 3.10 for the CPR model. Values projected to the right of this hyperplane will be classified with Pulseless labels while values projected to the left will be classified with Pulse labels. As seen in this figure, many of segments the training and test data have been classified correctly. Those that are incorrectly classified (i.e. pulseless segments that are classified as pulse segments or vice versa) are 'borderline' data points and are often near the hyperplane. This minimization of computational energy could allow the algorithm to be incorporated into devices currently in production and enable real-time feedback to rescuers during CPR. As this algorithm has the priority of scalability and in-field assessment, other manually-designed and spectral features used in previous studies [34] were not used in this algorithm. The addition of such features could add computational complexity which would inevitably compromise scalability.

The algorithm described in the current study incorporates three spatial modes of the reduced wavelet scalogram, which enables some limited interpretability of the underlying

variance of the two classes of data. Examination of the spatial modes demonstrates a band in most of the modes around approximately 1.5-2 Hz (Fig. 3.6). This fundamental band and its harmonics may represent some effects of CPR (guidelines recommend CPR rates of 100-120 compression per minute) [5] , but could also represent underlying heart rate. Heart rate may be intrinsically incorporated in these three modes, which could account for why the addition of heart rate as a separate feature did not appreciably improve model performance versus use of modes 1-3 alone. In modes 1 to 3, there is also a significant amount of energy at higher frequencies. However, in contrast, mode 4 exhibits minimal energy above the 2 Hz band other than the first and second harmonics (approximately 4 and 6 Hz). Because each mode represents a linear combination of the high-dimensional scalogram, there is not a definitive relationship between the modes and conventionally-understood ECG characteristics (e.g, amplitude, QRS width, QRS rate, and rhythmicity). However, further insight can be gained by visualizing representative cases of the temporal modes of this analysis. For example, a case that maps to the cluster centroid of the Pulse classification has QRS complexes that are more narrow and larger in amplitude than the cluster centroid of the No Pulse classification, as well as a faster rate (see Fig. 3.9). Narrower QRS complexes could be indicative of increased myocardial conduction velocity, and thus may potentially be represented by increased broad-band high-frequency content such as that observed in modes 1-3.

3.4.1 Limitations

In order to be deployed in clinical practice, the algorithm needs to be integrated with a function that can distinguish rhythms during CPR [2, 66, 85], because in application it assumes that the underlying rhythm is organized. In addition, further study is necessary to determine how this algorithm might be implemented into clinical practice to affect rescuer actions and resuscitation metrics such as CPR fraction and drug administration. Although the ascertainment of pulse presence used information from real-time audio and measured blood pressure, pulse detection can be clinically difficult and result in misclassification of its presence or absence that could attenuate the accuracy of the algorithm.While the

study cohort included patients with initial rhythms of ventricular fibrillation, asystole, and pulseless electrical activity, the initial rhythm distribution was weighted to include excess ventricular fibrillation given the greater likelihood of spontaneous pulse and the added analytical power. Moreover cases came from a single EMS system with generally good outcomes. These characteristics (initial rhythm distribution and EMS system) may affect algorithm performance and hence the generalizability of the results.

3.4.2 Future Directions

There are many ongoing challenges and promising directions that motivate future research in this area. The raw ECG data-streams remain noisy and have frequent artifacts. Machine learning architectures can potentially be integrated at the front end of methods in order to help provide more robust filtering of the ECG time-series data. Further, as recording technologies improve and more data is available, more powerful neural network architectures can be used in the classification step to drive accuracies even higher. Although the wavelet decomposition proved to be highly advantageous for extracting key features of the pulse presence, there is a potential for improving the feature space used for extraction of dominant spatio-temporal signals responsible for accurate clustering and thus improving model performance. Regardless, the data-driven discovery pipeline proposed here provides a baseline model for which improvements can be readily integrated and executed.

3.5 Conclusion

In summary, we developed a machine learning framework for automating the accurate prediction of pulse presence from the ECG time-series. The proposed algorithmic architecture integrates and leverages three mathematical architectures: (i) time-series analysis through a wavelet decomposition, (ii) feature engineering through dominant correlated spatio-temporal patterns, and (iii) clustering and classification methods. We demonstrated moderate predictive ability in identifying pulse presence even while CPR is being performed on a patient, which has potential to provide essential feedback to rescuers during uninterrupted CPR. The mathematical approach provides a data-driven framework that can be applied widely across current technology, giving a generalizable approach for clinical pulse prediction to

potentially improve outcomes following cardiac arrest. Indeed, this approach suggests an opportunity to improve resuscitation through real-time identification of pulse presence to minimize CPR interruptions and inform treatment decisions.

Chapter 4

BAGGING, OPTIMIZED DYNAMIC MODE DECOMPOSITION (BOP-DMD) FOR ROBUST, STABLE FORECASTING WITH SPATIAL AND TEMPORAL UNCERTAINTY-QUANTIFICATION

Adapted from: **Sashidhar, Diya** and J. Nathan Kutz. Bagging, optimized dynamic mode decomposition (BOP-DMD) for robust, stable forecasting with spatial and temporal uncertainty-quantification. Philosophical Transactions of the Royal Society A. 2021.

4.1 Introduction

The data-driven modeling and control of complex systems, which includes the ability to produce accurate and robust forecasting algorithms, is a rapidly evolving field with the potential to transform the engineering, biological, and physical sciences. The combination of high-fidelity measurements from modern sensor technologies and numerical simulations has ensured that while data is often abundant, accurate and computationally efficient models for forecasting remain challenging. *Dynamic mode decomposition* (DMD) provides a data-driven regression architecture for adaptively learning linear dynamics models over snapshots of temporal data. Although often used for the discovery of approximate dynamical systems from high-dimensional, spatio-temporal data (e.g. fluid flows), it can equally applied to simple time series measurements to produce best-fit models. DMD has been broadly used in the scientific community due to its ease of use, interpretability and adaptive nature [62]. However, bias induced by measurement noise has significantly limited the forecasting and reconstruction performance of DMD algorithms, as illustrated in the example measurements of atmospheric chemistry dynamics of Fig. 4.1. As shown in Fig. 4.2, using an optimized DMD formulation, we show that leveraging the statistical method of bagging (**b**ootstrap **agg**regat**ing**) [18] significantly improves DMD robustness and accuracy for forecasting while also providing *uncertainty quantification* (UQ) in its predictions and reconstructions.

Dynamic mode decomposition originated in the fluid dynamics community. Introduced

as as algorithm by Schmid [88,89], it has rapidly become a commonly used data-driven analysis tool and the standard algorithm to approximate the Koopman operator from data [84]. In the context of fluid dynamics, DMD was used to identify spatio-temporal coherent fluid structures from high-dimensional time-series data. The DMD analysis offered an alternative to standard dimensionality reduction methods such as the *proper orthogonal decomposition* (POD), which highlighted low-rank features in fluid flows using the computationally efficient *singular value decomposition* (SVD) [61]. The advantage of using DMD over SVD is that the DMD modes are linear combinations of the SVD modes that have a common linear (exponential) behavior in time, given by oscillations at a fixed frequency with growth or decay. Specifically, DMD is a regression to solutions of the form

$$\mathbf{x}(t) = \sum_{j=1}^{r} \boldsymbol{\phi}_j e^{\omega_j t} b_j = \boldsymbol{\Phi} \exp(\boldsymbol{\Omega} t)\mathbf{b}, \tag{4.1}$$

where $\mathbf{x}(t)$ is an r-rank approximation to a collection of state space measurements $\mathbf{x}_k = \mathbf{x}(t_k)$ $(k = 1, 2, \cdots, n)$. The algorithm regresses to values of the DMD eigenvalues ω_j, DMD modes $\boldsymbol{\phi}_j$ and their loadings b_j. The ω_j determines the temporal behavior of the system associated with a modal structure $\boldsymbol{\phi}_j$, thus giving a highly interpretable representation of the dynamics. Such a regression can also be learned from time-series data [68]. DMD may be thought of as a combination of SVD/POD in space with the Fourier transform in time, combining the strengths of each approach [24, 62]. DMD is modular due to its simple formulation in terms of linear algebra, resulting in innovations related to control [79], compression [21, 37], reduced-order modeling [3], and multi-resolution analysis [23, 64], among others.In a related line of work, the spectral POD (SPOD) method and its modes are optimally averaged DMD modes obtained from an ensemble DMD problem for stationary flows [**?**]. The work explicitly connects the SPOD to the exact formulation of DMD.

Because of its simplicity and interpretability, DMD has been applied to a wide range of diverse applications beyond fluid mechanics, including neuroscience [19], disease modeling [80], robotics [39,40], video processing [36,47], power grids [92,93], financial markets [73], and plasma physics [57, 95]. The regression to (5.1) shows the immediate value of DMD for forecasting. Specifically, any time t^* can be evaluated to produce an approximation to the state of the system $\mathbf{x}(t^*)$. However, despite its introduction more than a decade ago,

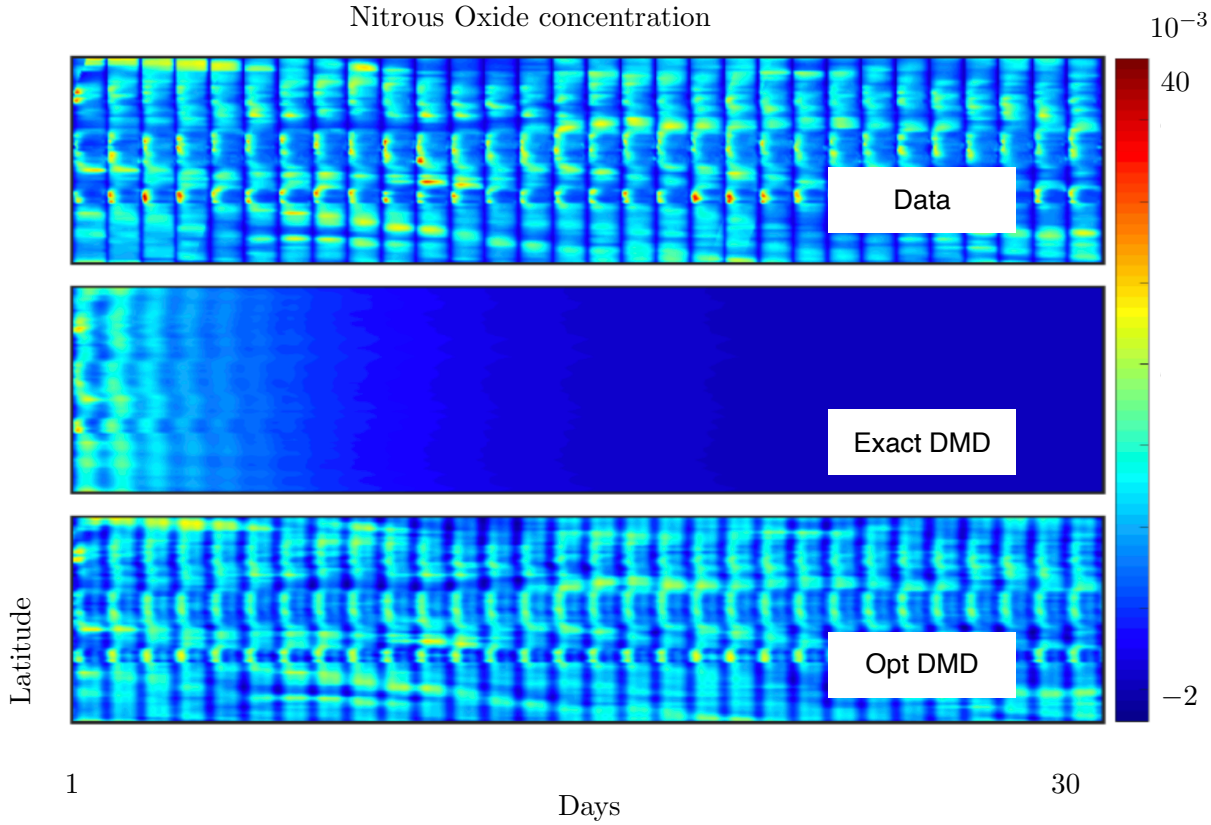

Figure 4.1: Canonical bias effect in DMD algorithms for noisy (normalized) data (top panel) for a specific longitude and elevation. In this example of atmospheric chemistry from Velagar et al [98], thirty days of nitrous oxide (NO) are measured and a DMD model regression is used to fit the data which has been normalized [98]. The bias of the exact DMD algorithm (middle panel) shows that the solution almost immediately tends to zero while optimized DMD (bottom panel) is able to correctly approximate the chemical dynamics.

DMD is rarely used for forecasting and/or reconstruction of time-series data except in cases with high-quality (noise-free or nearly noise free) data. Indeed, practitioners who work with DMD and noisy data know that the algorithm fails not only to produce a reasonable forecast, but also often fails in the task of reconstructing the time-series it was originally regressed to. Thus in the past decade, the value of DMD has largely been as an important *diagnostic* tool as the DMD modes and frequencies are highly interpretable. Indeed, from Schmid's [88, 89] original work until now, DMD papers are primarily diagnostic in nature

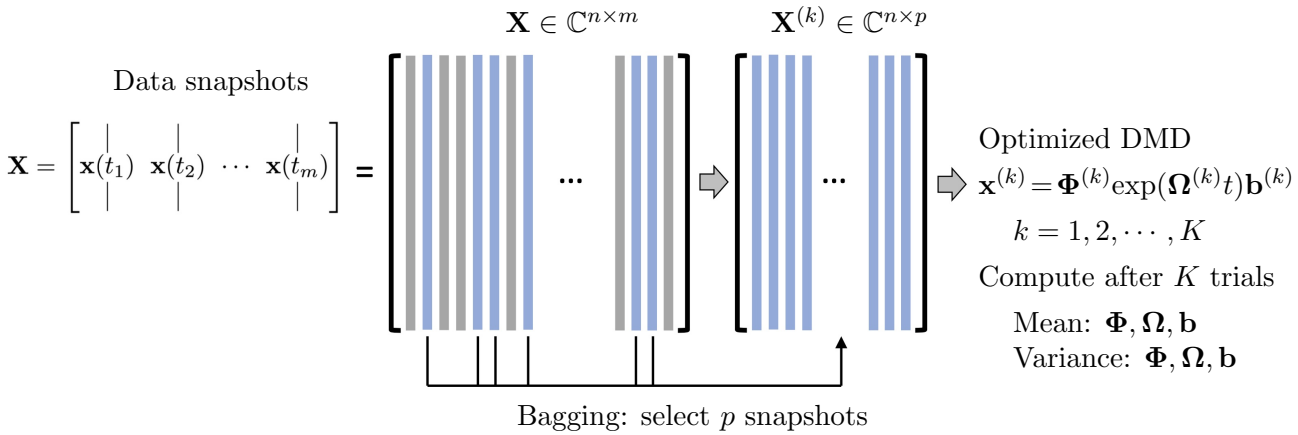

Figure 4.2: Summary of the BOP-DMD architecture. The data snapshots $\mathbf{x}(t_k)$ are collected over m snapshots into the matrix $\mathbf{X}$. Columns of $\mathbf{X}$ are randomly sub-selected into the matrix $\mathbf{X}^{(k)}$ to build an optimized DMD model. Each DMD model $\mathbf{x}^{(k)} = \mathbf{\Phi}^{(k)} \exp(\mathbf{\Omega}^{(k)} t)\mathbf{b}^{(k)}$ is used to compute the statistics (mean and variance) of the DMD parametrizations $\mathbf{\Phi}, \mathbf{\Omega}, \mathbf{b}$ which are used in building a the BOP-DMD ensemble solution with UQ.

with the key figures of any given paper being the DMD modes and eigenvalues. In cases, where DMD is used on noise-free data, such as for producing reduced order models from high-fidelity numerical simulation data [3, 12], then the DMD solution (5.1) can be used for reconstructing and forecasting accurate representations of the solution.

The origins of the DMD's poor performance for reconstruction and forecasting is due to bias-induced effects from noisy measurements [8, 12, 29, 49]. Indeed, based on observations of the DMD spectrum from Bagheri et al [12] and Degennaro et al [30], Hemati et al. [49] identified the explicit origins of this noise-driven bias effect on the eigevalue distribution and proposed methods that could circumvent it. Following [29, 49], Askham and Kutz [8] produced the *optimized* DMD (optDMD) algorithm which is a nonlinear optimization enabled by variable projection techniques which provides optimal de-biasing for a given signal-to-noise ratio. Indeed, optDMD is the standard for producing the most accurate and precise DMD modes and eigenvalues in the presence of noise. However, the variable projection method often fails to converge, limiting its usefulness. In what follows, we empower

the optimized DMD algorithm with three critical new features: (i) an initialization procedure allowing for stabilizing the convergence of the variable projection algorithm, (ii) a statistical bagging scheme [18], whose objective is to reduce variance and stabilize models, for improving the accuracy and robustness of the regression, and (iii) uncertainty quantification (UQ) in the DMD model, both temporal and spatial. Our algorithm, termed the *bagging, optimized dynamic mode decomposition* (BOP-DMD) is highlighted in Fig. 4.2 and is characterized against current DMD algorithms, demonstrating its performance for not only diagnostics, but also reconstruction and probabilistic forecasting. It is further applied to a number of example spatio-temporal systems. It should be noted that alternatives to improve DMD models include choosing appropriate windows of time for sampling data [50, 64] and ensembling DMD regressions on short burst of data, which was developed by Scandurra, Tezzele and Louiseau for the pyDMD package (Tutorial 8) [32]. These sampling strategies can also be used to great effect with the BOP-DMD algorithm, with the ensembling method being similar in spirit, but formalized and improved here with bagging and optimization. Bagging also provides a robustification technique whereby corrupt data is often not included in a down-selected set of snapshots used for ensembling the DMD. Thus the impact of data outliers on the DMD model is greatly reduced, as is standard in the bagging paradigm. BOP-DMD also easily integrates other methods for improving DMD models, including centering data [51] and time-delay embedding data [7, 20, 52, 56]. Importantly, BOP-DMD can be used on top of these algorithms instead of the exact DMD algorithm for improving accuracy, stability and performance while generating UQ metrics, reducing variance and preventing over-fitting through bagging [18].

4.2 Algorithms for Dynamic Mode Decomposition

The algorithmic construction of the DMD method can be best understood from the so-called *exact* DMD [97]. Indeed, this exact DMD is simply a least-square fitting procedure. Specifically, the DMD algorithm seeks a best fit linear operator $\mathbf{A}$ that approximately advances the state of a system, $\mathbf{x} \in \mathbb{R}^n$, forward in time according to the linear dynamical

system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \tag{4.2}$$

where $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, and Δt denotes a fixed time step that is small enough to resolve the highest frequencies in the dynamics. Thus, the operator $\mathbf{A}$ is an approximation of the Koopman operator $\mathcal{K}$ restricted to a measurement subspace spanned by direct measurements of the state $\mathbf{x}$ [84]. In the original DMD formulation [88], uniform sampling in time was required so that $t_k = k\Delta t$. The exact DMD algorithm [97] does not require uniform sampling. Rather, for each snapshot $\mathbf{x}(t_k)$ there is a corresponding snapshot $\mathbf{x}(t'_k)$ one time step Δt in the future. These snapshots are arranged into two matrices, $\mathbf{X}$ and $\mathbf{X}'$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_m) \\ | & | & & | \end{bmatrix} \tag{4.3a}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t'_1) & \mathbf{x}(t'_2) & \cdots & \mathbf{x}(t'_m) \\ | & | & & | \end{bmatrix}. \tag{4.3b}$$

In terms of these matrices, the DMD regression (4.2) is

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}. \tag{4.4}$$

The exact DMD is the best fit, in a least-squares sense, operator $\mathbf{A}$ that approximately advances snapshot measurements forward in time. Specifically, it can be formulated as an optimization problem

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger \tag{4.5}$$

where $\|\cdot\|_F$ is the Frobenius norm and † denotes the pseudo-inverse. The pseudo-inverse may be computed using the SVD of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ as $\mathbf{X}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*$. The matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V}^{m\times m}$ are unitary, so that $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^*\mathbf{V} = \mathbf{I}$, where * denotes complex-conjugate transpose. The columns of $\mathbf{U}$ are known as POD modes. Often for high-dimensional

data, the DMD leverages low-rank structure by first projecting $\mathbf{A}$ onto the first r POD modes in $\mathbf{U}_r$ and approximating the pseudo-inverse using the rank-r SVD approximation $\mathbf{X} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^*$:

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{A} \mathbf{U}_r = \mathbf{U}_r^* \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1}. \tag{4.6}$$

The leading spectral decomposition of $\mathbf{A}$ may be approximated from the spectral decomposition of the much smaller $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \mathbf{\Lambda}. \tag{4.7}$$

The diagonal matrix $\mathbf{\Lambda}$ contains the *DMD eigenvalues*, which correspond to eigenvalues of the high-dimensional matrix $\mathbf{A}$. The columns of $\mathbf{W}$ are eigenvectors of $\tilde{\mathbf{A}}$, and provide a coordinate transformation that diagonalizes the matrix. These columns may be thought of as linear combinations of POD mode amplitudes that behave linearly with a single temporal pattern given by the corresponding eigenvalue λ.

The eigenvectors of $\mathbf{A}$ are the *DMD modes* $\mathbf{\Phi}$, and they are reconstructed using the eigenvectors $\mathbf{W}$ of the reduced system and the time-shifted data matrix $\mathbf{X}'$:

$$\mathbf{\Phi} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \mathbf{W}. \tag{4.8}$$

Tu et al. [97] proved that these DMD modes are eigenvectors of the full $\mathbf{A}$ matrix under certain conditions. As already shown in the introduction, the DMD decomposition allows for a reconstruction of the solution as (5.1). The amplitudes of each mode $\mathbf{b}$ can be computed from

$$\mathbf{b} = \mathbf{\Phi}^\dagger \mathbf{x}_1., \tag{4.9}$$

however, alternative and often better approaches are available [8, 24, 55]. These alternative approaches compute the vector $\mathbf{b}$ by regressing across all the snapshots, not just a single time instance as given by Equation 4.9 Thus, the data matrix $\mathbf{X}$ may be reconstructed as

$$\begin{aligned} \mathbf{X} &\approx \mathbf{\Phi} \mathrm{diag}(\mathbf{b}) \mathbf{T}(\boldsymbol{\omega}) \\ &= \begin{bmatrix} | & & | \\ \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \\ | & & | \end{bmatrix} \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_r \end{bmatrix} \begin{bmatrix} e^{\omega_1 t_1} & \cdots & e^{\omega_1 t_m} \\ \vdots & \ddots & \vdots \\ e^{\omega_r t_1} & \cdots & e^{\omega_r t_m} \end{bmatrix}. \end{aligned} \tag{4.10}$$

Bagheri [12] first highlighted that DMD is particularly sensitive to the effects of noisy data, with systematic biases introduced to the eigenvalue distribution [11, 29, 33, 48]. As a result, a number of methods have been introduced to stabilize performance, including total least-squares DMD [48], forward-backward DMD [29], variational DMD [10], subspace DMD [94], time-delay embedded DMD [7, 20, 52, 56] and robust DMD methods [8, 87]. However, the *optimized DMD* algorithm of Askham and Kutz [8], which uses a variable projection method for nonlinear least squares to compute the DMD for unevenly timed samples, provides the best and optimal performance of any algorithm currently available. Indeed, this optimal performance is mathematically guaranteed by the exponential fitting procedure of Askham and Kutz [8]. This is not surprising given that it actually is constructed to optimally satisfy the DMD problem formulation. Specifically, the optimized DMD algorithm solves the exponential fitting problem directly:

$$\underset{\boldsymbol{\omega},\boldsymbol{\Phi}_\mathbf{b}}{\operatorname{argmin}} \|\mathbf{X} - \boldsymbol{\Phi}_\mathbf{b}\mathbf{T}(\boldsymbol{\omega})\|_F, \tag{4.11}$$

where $\boldsymbol{\Phi}_b = \boldsymbol{\Phi}\mathbf{diag}(\mathbf{b})$. This has been shown to provide a superior decomposition due to its ability to optimally suppress bias and handle snapshots collected at arbitrary times. The disadvantage of optimized DMD is that one must solve a nonlinear optimization problem, often which can fail to converge.

4.3 BOP-DMD: Bagging, Optimized DMD

BOP-DMD leverages Breiman's statistical bagging sampling strategy [18] shown in Fig. 4.2 along with the optimized DMD algorithm. Bagging is designed to produce an ensemble of models, thereby reducing model variance and suppressing over-fitting by design. Not only is ensembling improve DMD, it also is effective in deep neural network regressions [4]. Further innovations here include stabilizing the variable projection technique used by optDMD so that it converges consistently to an optimal solution. Its ability to converge is often dependent upon a suitable initial guess for the DMD eigenvalues and eigenvectors. More precisely, we have observed through simulations that the initial guess of the eigenvectors has the most pronounced effect on the reliable convergence of variable projection. In its standard form,

Algorithm 1 BOP-DMD

Input: Input ($\mathbf{X}$, p, K)

procedure BOPDMD($\mathbf{X}$, p, K)

Compute $\mathbf{\Phi}_0, \mathbf{\Omega}_0, \mathbf{b}_0$ — optDMD regression

for $k \in \{1, 2, \cdots, K\}$ **do** — Compute K optDMD models.

Choose p of m snapshots ($p < m$) — Bagging

opt-DMD $\mathbf{\Phi}_k, \mathbf{\Omega}_k, \mathbf{b}_k$ — Initialize with $\mathbf{\Omega}_0$

Update $\mathbf{\Phi}_0, \mathbf{\Omega}_0, \mathbf{b}_0$ — Use improved $\mathbf{\Phi}, \mathbf{\Omega}, \mathbf{b}$

end for

Compute mean $\boldsymbol{\mu} = \{\langle \mathbf{\Phi} \rangle, \langle \mathbf{\Omega} \rangle, \langle \mathbf{b} \rangle\}$

Compute variance $\boldsymbol{\sigma} = \{\langle \mathbf{\Phi}^2 \rangle, \langle \mathbf{\Omega}^2 \rangle, \langle \mathbf{b}^2 \rangle\}$

return $\boldsymbol{\mu}, \boldsymbol{\sigma}$ — Return optDMD parameters.

end procedure

no initial guess is given and failure to converge is common for complex, noisy data.

The BOP-DMD algorithm accounts for the initialization process and further provides the optimal solutions to linear models by using opt-DMD as the regression architecture. Algorithm 1 shows the algorithmic structure of BOP-DMD, highlighting the bagging, initialization and ensembling of the DMD models to produce an ensemble, probabilistic DMD model. The initialization of DMD is accomplished by first constructing an optDMD model approximation, whose eigenvalues and eigenvectors $\mathbf{\Phi}_0$ can be used to seed the BOP-DMD.

Algorithm 1 provides a robust method for producing DMD models. As will be shown in the next section, BOP-DMD outperforms the opt-DMD algorithm which has already been shown to be the gold standard for producing DMD models [8, 10]. In addition to the DMD model itself, the output of the algorithm can be used to produce a forecast with UQ metrics as shown in Algorithm 2. Specifically, the BOP-DMD model is used in a Monte-Carlo fashion to produce a probabilistic forecast for a given system. This forecasting is accomplished by simply drawing from the probability distributions of the $\mathbf{\Phi}, \mathbf{\Omega}, \mathbf{b}$ computed from BOP-DMD. The mean and variance of the forecast are produced, giving uncertainty

Algorithm 2 Forecasting BOP-DMD

Input: Input $(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{T}, K\ )$

procedure FORECASTBOPDMD$(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{T}, K)$

for $k \in \{1, 2, \cdots, K\}$ **do** — Compute K Forecasts.

Generate $\langle \boldsymbol{\Phi} \rangle, \boldsymbol{\Omega}_k, \mathbf{b}_k$ — Drawn from $(\boldsymbol{\mu}, \boldsymbol{\sigma})$

$\mathbf{x}_k(\mathbf{T}) = \langle \boldsymbol{\Phi} \rangle \exp(\boldsymbol{\Omega}_k \mathbf{T}) \mathbf{b}_k$ — Solution for times $\mathbf{T}$

end for

Compute mean $\langle \mathbf{x}(\mathbf{T}) \rangle$

Compute variance $\langle \mathbf{x}^2(\mathbf{T}) \rangle$

return $\langle \mathbf{x}(\mathbf{T}) \rangle$, $\langle \mathbf{x}^2(\mathbf{T}) \rangle$ — Return optDMD forecast

end procedure

quantification with the forecasting prediction. Both Algorithms 1 and 2 are demonstrated in what follows.

4.4 A Simple Example

We first tested the performance of the BOP-DMD algorithm on a simple example with additive noise. Algorithm performances were assessed by comparing the computed eigenvalues using optDMD and BOP-DMD to the analytical eigenvalues, as highlighted in previous works [8, 10, 29, 48].

4.4.1 Generation of Data

The data used in this example was generated using the following equation:

$$f(x,t) = g_1(x)e^{\lambda_1 t} + g_2(x)e^{\lambda_2 t} + g_3(x)e^{\lambda_3 t} + \sigma\epsilon \tag{4.12}$$

where $\epsilon \sim N(0,1)$ is normally distributed noise and

$$g_1(x) = \sin(x)$$

$$g_2(x) = \cos(x)$$

$$g_3(x) = \tanh(x).$$

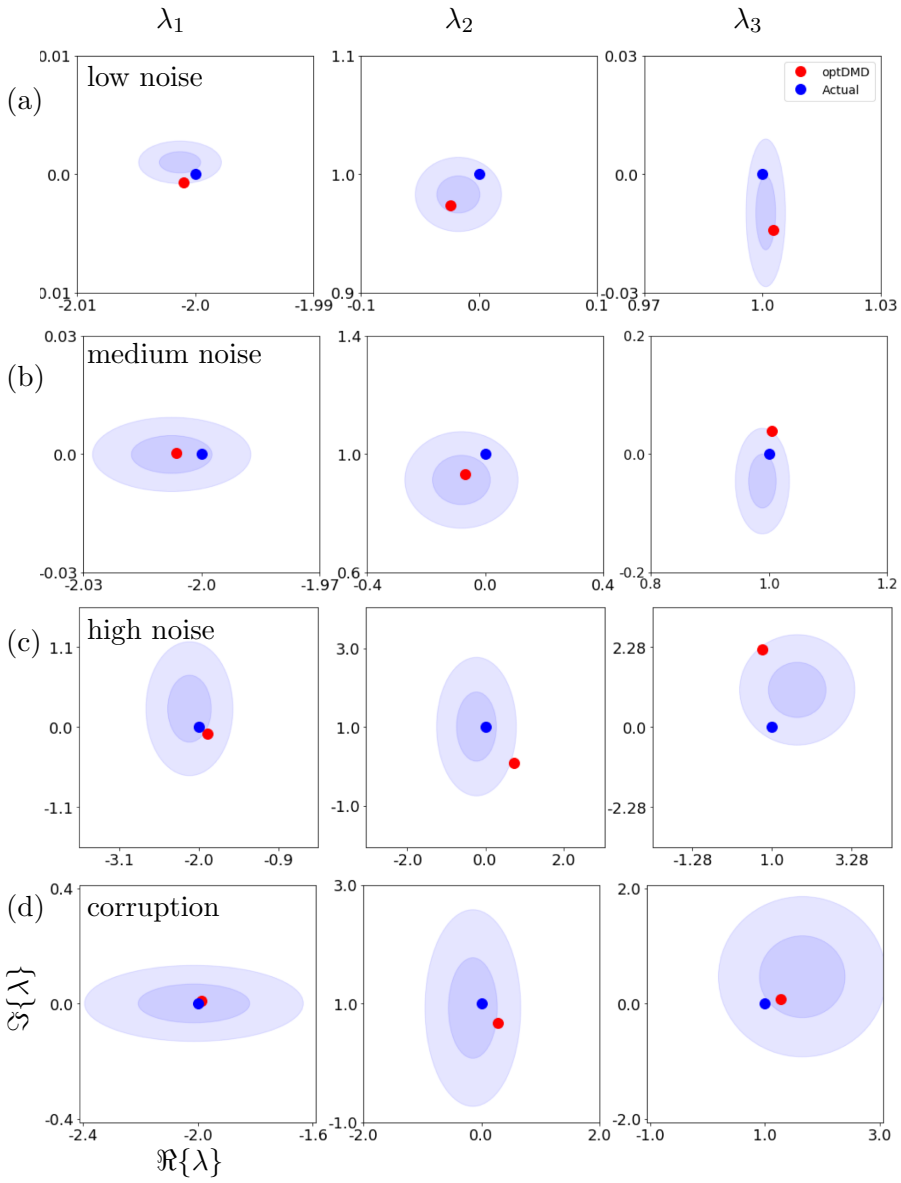

Figure 4.3: Uncertainty quantification for computed eigenvalue magnitudes for 4 separate noise realizations (a - d) for eigenvalues $\lambda_1 = -2$, $\lambda_2 = i$, $\lambda_3 = 1$. For comparison, the magnitude of the true eigenvalue is depicted by a solid blue dot, the magnitude of the computed eigenvalue using optDMD is depicted by a solid red dot. The two concentric circles represent the range of eigenvalue magnitudes within one and two standard deviations, respectively. These results were generating using k = 100 trials of the BOP-DMD algorithm with a randomly selected subset of size p = 20 for low and medium noise settings (a and b) and p = 50 for higher noise settings (c and d).

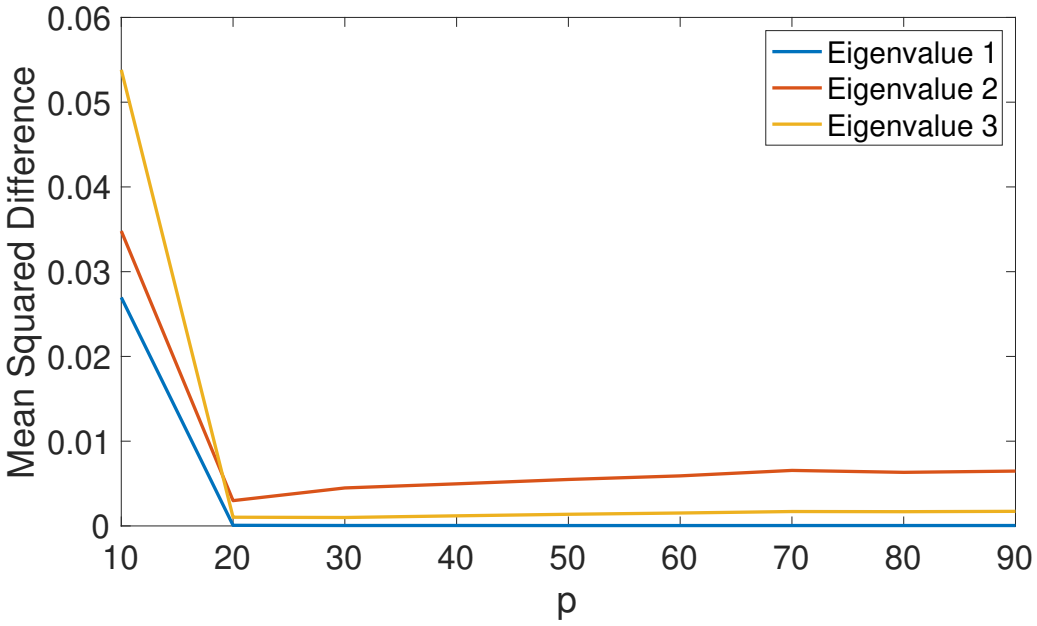

Figure 4.4: Evaluation of the bagging sub-selection of data with medium noise. In this case, there are $m = 100$ data snapshots and p randomly sub-selected snapshots for bagging. The mean squared difference between the true value and mean across trials for various p are shown. Note that only $p = 20$ sub-selected snapshots are required to achieve convergence.

Here, σ ranges from 0.001 to 0.05 (depending on noise setting), $x \in [0, 1]$, and $t \in [0, 1]$. Specifically, σ was set to 0.001, 0.005, 0.05 for low, medium, and high noise settings. Corruption was introduced into the dataset by adding sharp spikes to the dataset with high noise at randomly selected spatiotemporal coordinates. To create the data matrix $X \in \mathbb{R}^{mxn}$, Equation 4.12 was evaluated on a gridspace comprised of equally spaced spatial points, x_k, for $k = 1...n$, and equally spaced time points, t_k, for $k = 1...m$. These sampling points are used to construct an individual state space estimate $\mathbf{x}_k = \mathbf{x}(t_k)$ and the full data matrix $\mathbf{X} \in \mathbb{C}^{n \times m}$.

4.4.2 Methods and Results

Like many iterative methods, the choice of initial conditions dictates model trajectory. In this case, we used the original optDMD algorithm to generate initial eigenvectors for the

BOP-DMD algorithm. In order to create an initial seed for the BOP-DMD algorithm, the original optDMD algorithm was run on the full data matrix, $\mathbf{X}$, without initial conditions. The eigenvalues outputted by this algorithm, $\mathbf{\Omega}_{opt}$, were used as initial conditions for our algorithm. For k trials, eigenvalues of various subsets were numerically computed using the aforementioned initial conditions. As shown in Figure 4.2, these subsets, $\mathbf{X}^{(k)} \in \mathbb{R}^{n \times p}$, for $k = 1...100$, were generated by concatenating p randomly selected column indices of the original data matrix, $\mathbf{X}$.

Figure 4.3 shows the BOP-DMD distributions of the computed eigenvalues for each of these subsets. These distributions were generated for various noise realizations (a-c). The ellipses represent the standard deviation along the imaginary and real axes for the k = 100 trials. As the resulting distributions are symmetric Gaussian for both imaginary and real components (with different standard deviations), we depict the uncertainty quantification in terms of elliptical rings. The center of the ellipse is the mean value of BOP-DMD after ensembling while the darker ring represents the region of values one standard deviation away from the mean and the lighter ring represents two standard deviations away from the mean. Not surprisingly, as the added noise increases, the uncertainty or width of the ellipses in this figure increases as well. The actual and optDMD value when conducted on the full dataset with no initial conditions are also included in this figure for comparison. In most of the distributions, the mean computed eigenvalue using bagging performed better than the original optDMD algorithm. More importantly, in each of the three noise realizations,the actual value is captured within one standard deviation of the mean value of BOP-DMD ensembles, highlighting the robustness and efficacy of this algorithm.

Figure 4.4 shows the squared difference in error between the mean computed eigenvalue and true eigenvalue for this particular noise setting ($\sigma = 0.005$). This process of finding the optimal p is repeated for each noise setting and dataset in order to reach a Pareto optimal solution [22]. As the batch sizes, p, increases, this error appears to plateau. Out of m = 100 snapshots, only 20 randomly selected snapshots are required for convergence. More importantly, optimal performance is actually achieved with p = 20 for this particular noise setting. This process of finding the optimal p is repeated for each noise setting and dataset in order to reach a Pareto optimal solution. The choice of k relies on the complexity of the

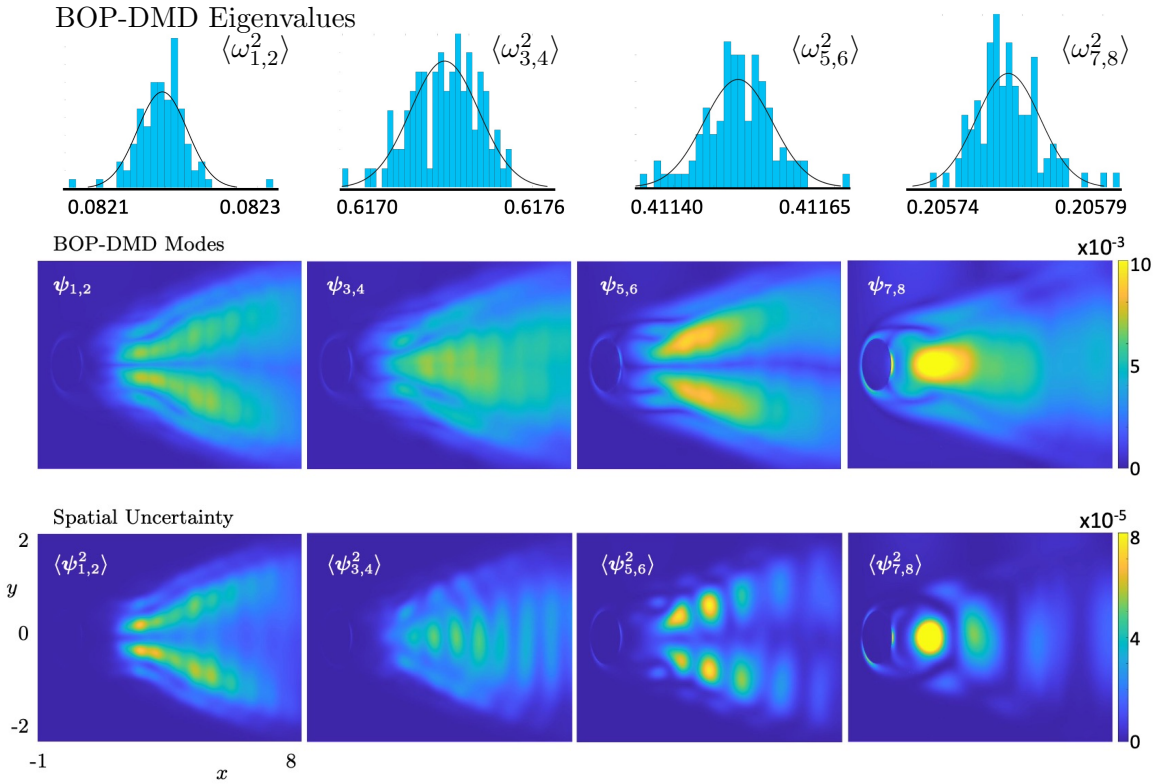

Figure 4.5: Vortex Shedding Example with noise and corruption (first 8 paired modes): Top panel: Temporal uncertainty quantification for eigenvalues corresponding to modes. The black lines represent a least-square fit of a normal distribution. Middle panel: Mean of the BOP-DMD modes generated after bagging with $K = 100$ trials. Bottom panel: Spatial variance of the uncertainty over the $K = 100$ ensemble. Areas with yellow shading indicate high model uncertainty, while areas with blue shading indicate low model uncertainty.

dataset. As this is a smaller dataset (m = 100, n = 200) with relatively simple dynamics, we used k = 100 trials. However, in larger and more complex datasets, a larger k is required in order to fully capture dynamics.

4.5 BOP-DMD Applications

In what follows, we apply the BOP-DMD algorithm to two spatio-temporal systems: flow around a cylinder and sea-surface temperature data. Both provide a platform for the evaluation of the algorithm's performance, including its ability to produce UQ metrics in both

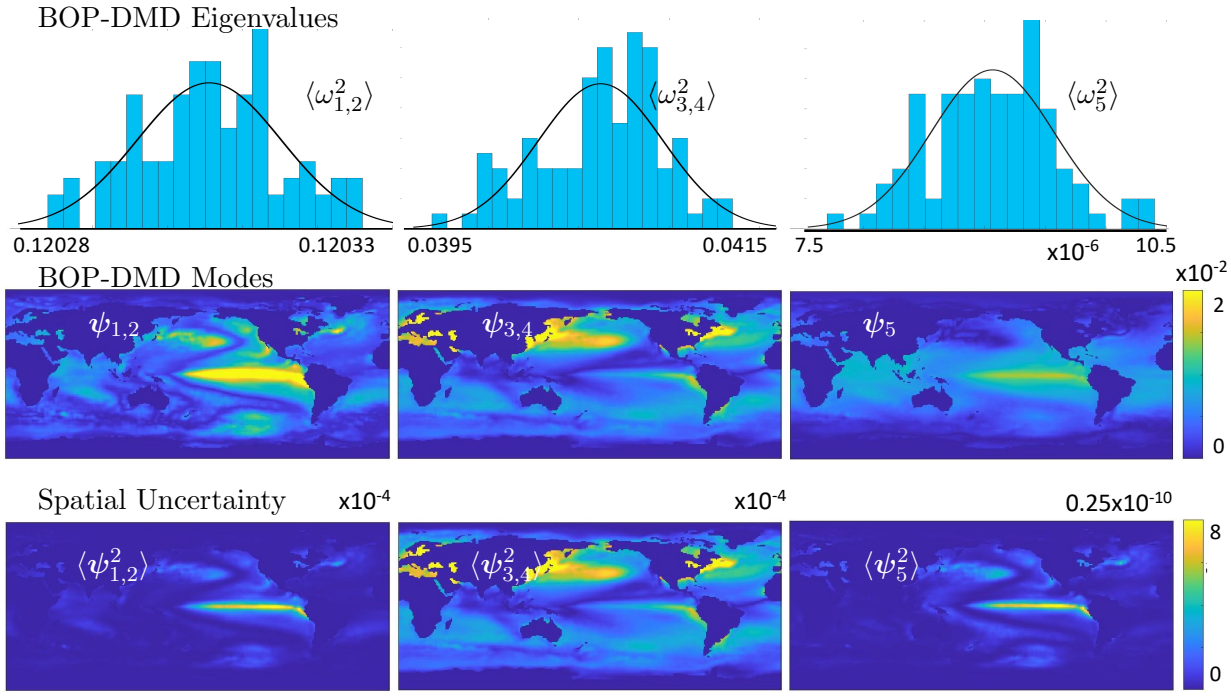

Figure 4.6: SST data (first 5 modes): Top panel: Uncertainty quantification for eigenvalues corresponding to modes. The black line represents a least-square fit of a normal distribution. Middle panel: Mean of the DMD modes generated after bagging with $K = 100$ trials. Bottom panel: Spatial variance of the BOP-DMD modes over the ensemble of models. Areas with yellow shading indicate high model uncertainty, while areas with blue shading indicate low model uncertainty.

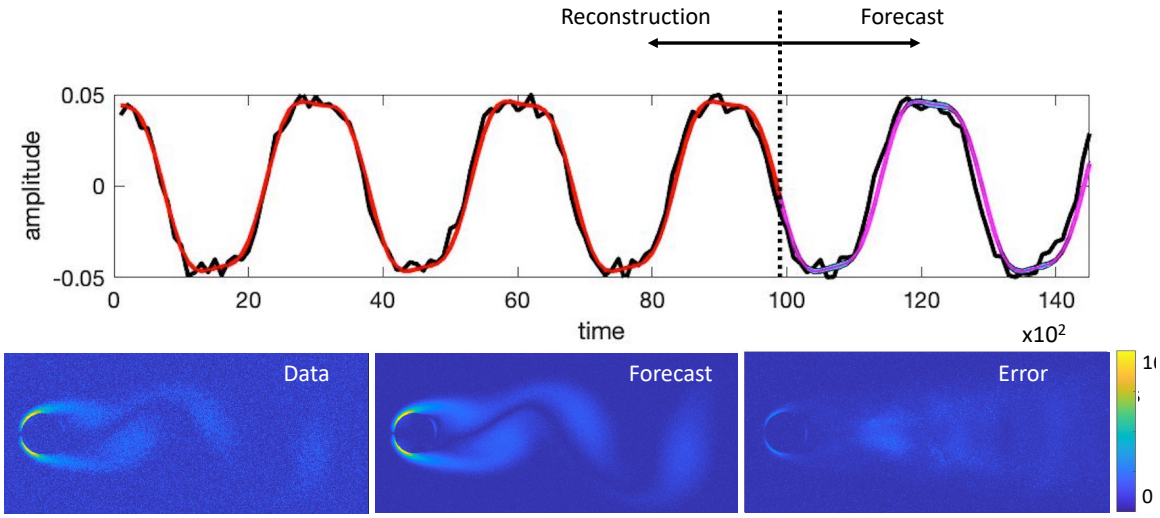

Figure 4.7: Forecasting skill of BOP-DMD for flow around cylinder with noise as illustrated in Fig. 4.5. The BOP-DMD model is trained over $t \in [0, 100]$ and forecast in the time range $t \in [100, 150]$. For this data set, the BOP-DMD model provides an accurate and stable prediction with nearly vanishing variance. The top panel shows the average vorticity data (black line) as a function of time along with the BOP-DMD reconstruction (red line) and forecast (magenta line). The slightly visible cyan line in the forecast shows the variance around the average of the magenta forecast line.

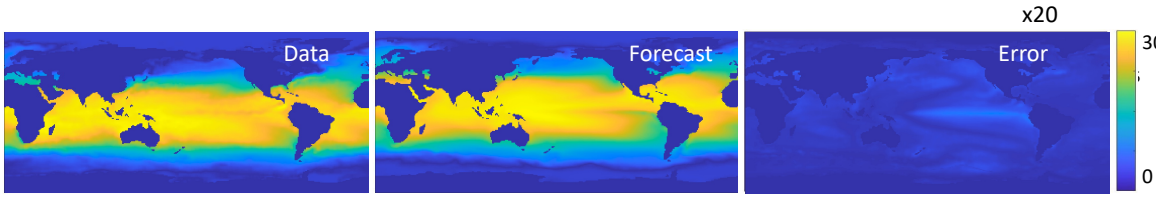

Figure 4.8: Forecasting skill of BOP-DMD for global sea-surface temperature. The BOP-DMD model is trained over a 19.2 year period (1000 snapshots) and forecast to the next 7.6 years (400 snapshots). For this data set, the BOP-DMD model provides an accurate and stable prediction with nearly vanishing variance with five modes. The error in the right panel is multiplied by 20 for visibility on the same color scale.

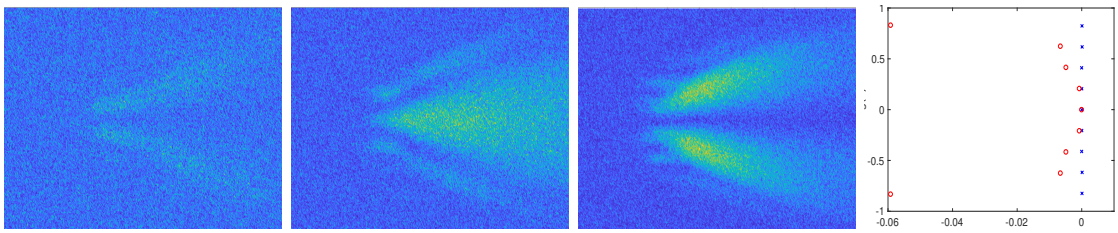

Figure 4.9: Exact DMD spatial modes when applied to Vortex Shedding Dataset. Note that eigenvalues for BOP-DMD are along imaginary axis, while exact DMD eigenvalues have a real component, introducing bias into the model.

space and time.

4.5.1 Flow Around a cylinder

As a common DMD example [12], we simulate the fluid flow past a circular cylinder with the two-dimensional, incompressible Navier-Stokes equations at $Re = 100$:

$$\nabla \cdot \boldsymbol{u} = 0, \quad \partial_t \boldsymbol{u} + (\mathbf{d}u \cdot \nabla)\mathbf{d}u = -\nabla p + \frac{1}{Re}\Delta \boldsymbol{u} \tag{4.13}$$

where $\boldsymbol{u}$ is the two-component flow velocity field in 2D and $\boldsymbol{p}$ is the pressure term. For Reynold's number $Re = Re_c \approx 47$, the fluid flow past a cylinder undergoes a supercritical Hopf bifurcation, where the steady flow for $Re < Re_c$ transitions to unsteady vortex shedding [13]. The unfolding gives the celebrated Stuart-Landau ODE, which is essentially the Hopf normal form in complex coordinates. This has resulted in accurate and efficient reduced-order models for this system [76, 77].

Figure 4.5 shows the first eight normalized BOP-DMD modes constructed from $K = 100$ trials using noisy, non-dimensional data. These modes contain distinct periodic behavior with periodicity of 76.5, 10.17, 15.2, 30.5 time units. While we see distinct spatial dynamics in each of these modes, together these spatial modes both reconstruct and forecast vortex shedding with high accuracy when combined. The data set can be accessed on the github repository for this algorithm (See Code Availability). Specifically, we added Gaussian noise to each of the spatiotemporal points in this dataset. The modes are cleanly produced and the spatial UQ, as illustrated in the bottom panel of the figure, show the spatial patterns where

DMD is most uncertain in its ability to provide an accurate reconstruction. With $K = 100$ ensembles, BOP-DMD is able to both produce UQ metrics for the DMD eigenvalues and their temporal uncertainty as well as a spatial uncertainty map for the variance of each of the modes. The spatial and temporal UQ provide a framework for Monte Carlo forecasting since the probability distributions of both the spatial and temporal fields can be drawn from to estimate an ensemble of future state predictions.

Although flow around a cylinder is a well known example, it helps illustrates the key concept that BOP-DMD stabilizes the prediction of the DMD modes and eigenvalues by ensembling them. More than that, the UQ can be explicitly computed for all spatial points and every DMD frequency extracted. In addition, BOP-DMD is able removing DMD bias induced by noise which leads to issues as illustrated in Fig. 4.1.

4.5.2 Sea-surface temperature

Another common example used in the DMD and POD literature is sea-surface temperature data. We consider the NOAA_OISST_V2 global ocean surface temperature data set spanning 1990–2016. The data are publicly available online [1]. The following dataset is comprised of weekly sensor and satellite measurements. Each snapshot is a 360 by 180 matrix with average sea surface temperature. These grid cells correspond to the latitude and longitude values. The first 1000 snapshots, which ends in 2007 (19.2 years) are used for producing BOP-DMD modes whose resolution is 360x180 pixels. In the next section, these BOP-DMD modes are used for forecasting.

The normalized BOP-DMD modes are constructed from $K = 100$ trials of the data with batch sizes of p = 500. The number of modes to select was optimized using a grid search on the training data with various combinations of modes. It was found that five modes was the optimal number of modes to be used on this dataset. Figure 4.6 shows the first five BOP-DMD modes along with their DMD eigenvalues. The first four modes are two conjugates pairs and the last mode, is a growth mode with only a real component. The first two modes have a mean periodicity of 52 weeks with the main variance occurring along the equator. The second pair of eigenmodes has a mean periodicity of approximately three years, with

the main variance occuring along the coast of East Asia and north of the equator (near the North Equatorial ocean current) as well as near the north east coast of North America, near the Gulf Stream. The fifth mode corresponds to an eigenvalue with a very small real component. This eigenmode has the highest variance north of the equator, east of South America.

In contrast to exact DMD modes (see Figure 4.9), the BOP-DMD modes are cleanly produced and have eigenvalues with mostly imaginary components. Specifically, the modes found using exact DMD appear to retain noise found in the data whereas the modes using BOP-DMD appear to have denoised the data, creating sharper, more refined modes. This lack of noise found in the modes themselves enables the potential for cleaner forecasts. As opposed to the exact DMD eigenvalues, the mean eigenvalues of BOP-DMD lie along the imaginary axis with no real component, enabling long term forecasting with minimal exponential growth or decay. While exact DMD eigenvalues also have an imaginary component, they also have a non-trivial real component. In this case, these negative real components contribute a general exponential decay in the series as time progresses, resulting in poor long-term forecasting abilities.

The spatial UQ, as illustrated in the bottom panel of the Figure 4.6, show the spatial patterns where DMD is most uncertain in its ability to provide an accurate reconstruction. In each of these spatial uncertainty maps, the uncertainty quantification is at several magnitudes lower than that of the mean. Note the highest spatial uncertainty along the El Niño Southern Oscillation (ENSO) focused near the equator. The phenomenon is also captured in the growth mode, or fifth eigenmode. The BOP-DMD algorithm also quantifies the temporal uncertainty by allowing for a Gaussian fit to the eigenvalue distribution.

4.6 Forecasting with BOP-DMD

The BOP-DMD algorithm can produce an approximation to the solution (4.10) using just a few modes. This solution gives the capability to forecast future states of temporal systems, including the examples considered here of flow around a cylinder and sea-surface temperatures. Moreover, by using the BOP-DMD forecasting algorithm, Monte Carlo simulations can produce UQ for the future state by drawing from the spatial and temporal distributions

of the DMD modes and eigenvalues as shown in Fig. 4.5 and 4.6.

The forecasting capabilities for the two examples shown in the previous section are shown in Fig. 4.7 and 4.8. For the flow around a cylinder, the average vorticity over the domain is illustrated in the top panel for the next 50 time points. It should be noted that the model was trained using only 100 time points. The magenta line depicts areas in the forecast with low uncertainty whereas the cyan line depicts time points with higher uncertainty. These areas are typically in the peaks and troughs of this oscillating time series. The forecast in this data set has very little uncertainty as the dynamics is quite low-dimensional and BOP-DMD has no problem extracting the future state. Interestingly, the forecast itself is much cleaner than the original data. The added noise is clearly visible in the actual data snapshot whereas the forecasted snapshot generated with BOP-DMD provides a sharper, more refined version of the vortex shedding at that time point. Thus, BOP-DMD is not only able to forecast with accuracy using only 2/3 of the available data, but it also *denoises* the data while doing so.

For the sea-surface temperature data, The first 19.2 years (1000 snapshots) are used to predict the next 7.6 years (400 snapshots). The forecast at the 7.6 year mark is illustrated along with the error. Despite this long forecasting period, the method is able to preserve much of the dynamics found in the real data itself. In looking at Figure 4.8, we see that the model is able to capture the general dynamics of sea surface temperature despite the inherent noise found in the sensor and satellite measurements. Note that the largest error (Fig. 4.8) occurs where the spatial UQ metric is largest as shown in Fig. 4.6. Only five modes are used to forecast the sea-surface temperature data.

4.7 Conclusions

In conclusion, the *dynamic mode decomposition* has emerged as a flexible, adaptive, robust, and general purpose data-driven method for the analysis and characterization of a broad range of scientific applications [62]. The regression to a linear dynamical system also provides an interpretable, data-driven framework for modal decompositions whose temporal dynamics are exponentials. The sheer multitude of variants of the DMD regression framework highlights the broad interest from the community in making the method more stable,

robust, and predictive. The BOP-DMD algorithm outlined here integrates and leverages many advantageous mathematical techniques, including variable projection for the optimization and statistical bagging for creating an ensemble of models. When combined, these two widely used techniques are able to construct a model which is less prone to over-fitting and reduces model variance. Indeed, the ensemble of BOP-DMD provides both spatial and temporal UQ metrics for reconstruction and forecasting that are currently unavailable in other methods.

It is important to be specific about what we mean by improving performance using BOP-DMD. As shown in our comparisons, optDMD can at times perform as well or better than BOP-DMD. It is also clear that optDMD can be worse than BOP-DMD. The opt-DMD algorithm is by construction a single shot model which provides no feedback on whether the former or later cases hold (better or worse performance). Nor is there any information about how variable the models can be when working with a given set of snapshots. BOP-DMD overcomes this not only by providing uncertainty metrics explicitly, but also by stabilizing a model for a given set of snapshots by ensemble averaging. Thus BOP-DMD mitigates the risk of randomly generating a poor model using optDMD. It is important to note that in the comparisons made here for optDMD and BOP-DMD, the underlying model was known. In practice, the data-driven applications do not have an underlying truth model to calibrate the validity of the optDMD model, thus BOP-DMD provides a framework for cross-validating the model and determining the underlying volatility of the model from a given set of snapshots. The ensemble averaging thus results in a stabilized model for forecasting and reconstruction even under volatile conditions.

Demonstrated in this manuscript is the BOP-DMD algorithm on a number of applications. The algorithm not only stabilizes the DMD method and consistently removes noise bias, but also provides both a spatial and temporal uncertainty quantification of the linear model. Thus it improves on the performance of optimized DMD while providing additional and critical insights into the dynamics and its spatio-temporal variability. The method is adaptive, requires very little data, and provides the capability to produce stable, probabilistic forecasts. Moreover, it does this within the framework of an interpretable modal decomposition that reveals dominant spatio-temporal features in a given data set.

As data-driven methods continue to be developed across the engineering, physical and biological sciences, it is imperative that modeling strategies be capable of handling real data that is often noisy and high-dimensional. With its ease of use, minimal data requirements and interpretability, DMD provides an exceptional practical alternative to data-driven techniques, such as neural network models. The ensembling aspect of BOP-DMD also provides a Bayesian perspective on the linear model fitting by providing estimates of the probability distributions for the DMD eigenvalues and DMD eigenfunctions, both of which allow for estimates of the method's forecasting horizon and capabilities. The open source code provided allows for reproducible and broad usage across the sciences. Moreover, the underlying algorithm can be used with established DMD pre-processing innovations, including time-delay embeddings, data centering and multi-resolution analysis.

Code Availability: The code used in this paper can be found at https://github.com/dsashid/BOP-DMD. Use of this code requires installation of the initial optDMD framework which can be found at https://github.com/duqbo/optdmd.

Chapter 5

DIAGNOSTICS AND FORECASTING OF INFLUENZA TRANSMISSION USING OPT- AND BOP-DMD FRAMEWORKS. AN ANALYSIS OF THREE SPATIAL RESOLUTIONS

Adapted from: **Sashidhar, Diya**, Mauricio Santillana, and J. Nathan Kutz. Forecasting and Uncertainty Quantification of Influenza Transmission using Bagging Optimized Dynamic Mode Decomposition (in preparation)

5.1 Introduction

Understanding the transmission dynamics and predicting potential outbreaks of a given disease remains mathematically challenging, especially as the severity and timing of many diseases are highly-sensitive to a great number of climatic, environmental, geographical, social and political factors. Influenza (flu) is one of the most common viral infections in the world, having a significant yearly cycle which impacts the entire world population. Despite decades of tracking data, modeling and predicting flu transmission activity remains difficult due to the significant variability in the severity and initiation of the outbreak each season. We develop a data-driven algorithm that provides stable and robust long-term forecasts that enable geographical and temporal uncertainty quantification.

Flu forecasting methods and diagnostics tools aim to predict the characteristics of influenza seasons before disease activity occurs and is captured by the Center of Disease Control's (CDC) influenza surveillance system. However, the ability to forecast oftentimes does not imply a true understanding of model dynamics. While it may seem relatively trivial, such an understanding can be an invaluable asset towards model intuition and long term forecasting. This in turn, can guide policy implementation, vaccination intervention strategies, and decision making capabilities. In this work, we implement a data-driven forecasting algorithm based upon the recently developed *bagging, optimized dynamic mode decomposition* (BOP-DMD) [86]. While the original dynamic mode decomposition (DMD)

was recently applied as a diagnostic tool for epidemiology datasets [80], it lacked the ability to reconstruct or forecast flu time series data due inherent biasing and instability in the algorithm. The critical innovation leveraged in the BOP-DMD algorithm is the concept of statistical (ensemble) bagging [17] along with exponential fitting [8]. This probabilistic formulation produces a stable, adaptive, interpretable and accurate flu forecasting model with uncertainty quantification (UQ), both in space and in time. Fundamentally, the DMD framework is a decomposition of spatio-temporal flu data into dominant correlated geographical structures (modes) along with their temporal cycles. The discovery of these modes and their time dependence is stabilized by the BOP-DMD algorithm.

In the United States, it is estimated by the CDC the common flu effects on average 8% of the population each year. Flu is a contagious respiratory illness caused by influenza viruses that infect the nose, throat, and sometimes the lungs. It typically causes only mild illness, but does occasionally lead to severe illness, and sometimes death. Thus, yearly vaccination is recommended by the CDC. Influenza has regional, national and city-wide effects, all of which can be quite different from each other over the course of a season. The ability to accurately forecast flu outbreaks while understanding underlying geographical and temporal transmission dynamics can lead to improved vaccination strategies and to the potential suppression of major outbreaks.

The ubiquitous nature of forecasting in epidemiology has engendered significant and diverse mathematical developments for time-series forecasting methods [91]. These include classical time-series methods from statistics, such as ARIMA and its variants, to more recent neural network based approaches [45], such as LSTMs (long-term, short-term memory), GRUs (gated recurrent units), and ECNs (Echo-state networks). These diverse mathematical strategies regress to models fit to historical training data, often making assumptions that the data is generated from a stationary process with Gaussian distributed variability. However, climatic, environmental, social, geographical and political factors severely undermine such assumptions, making forecasting exceptionally difficult and requiring re-training of models as new data becomes available. Thus adaptive models that can be easily updated and trained for national, regional, and city-level prediction are desirable, along with the ability to produce uncertainty metrics in the forecast at different geographical scales.

DMD originated in the fluid dynamics community as data-driven algorithm for diagnostics by Schmid [88]. In this context, it was used to identify interpretable spatio-temporal coherent fluid structures from high-dimensional time-series data. The DMD analysis offered an alternative to standard dimensionality reduction methods which highlighted low-rank features in fluid flows using the computationally efficient *singular value decomposition* (SVD) [61]. The advantage of using DMD over SVD is that the DMD modes are linear combinations of the SVD modes that have a common linear (exponential) behavior in time, given by oscillations at a fixed frequency with growth or decay, both of which are highly-interpretable quantities. Specifically, DMD is a regression to solutions of the form

$$\mathbf{x}(t) = \sum_{j=1}^{r} \boldsymbol{\phi}_j e^{i\omega_j t} b_j = \boldsymbol{\Phi} \exp(i\Omega t)\mathbf{b}, \tag{5.1}$$

where $\mathbf{x}(t)$ is a r-rank approximation to a collection of state space measurements $\mathbf{x}_k = \mathbf{x}(k\Delta t)$ $(k = 1, 2, \cdots, n)$. The optimized DMD algorithm uses variable projection to regress to values of the DMD eigenvalues ω_j, DMD (spatio-temporal) modes $\boldsymbol{\phi}_j$ and their loadings b_j. The optimized DMD algorithm is critical for debiasing the DMD model (5.1) which is generally unstable, i.e. solutions go to infinity or zero. BOP-DMD improves on this algorithm by statistical bagging, thus further improving predictions in the presence of noise [86]. Note that the ω_j in (5.1) determines the temporal behavior of the system associated with a modal (geographical) structure $\boldsymbol{\phi}_j$, thus giving a highly interpretable representation of the dynamics, both in terms of spatial patterns and their temporal evolution. DMD may be thought of as an ideal combination of PCA (principal components analysis) in space with the Fourier transform in time, combining the strengths of each approach [62].

Equation (5.1) provides a forecast and reconstruction of time-series data as shown in Fig 5.1. It is easily computed from data snapshots alone, giving an equation-free approach for estimating future states of the system. It also can be easily re-computed with the addition of new data to update the model, providing flexibility and adaptibility. While DMD has been previously advocated in epidemiology as a valuable diagnostic tool [80], the DMD algorithm used was incapable of producing reconstructions, uncertainty metrics, or forecasts of the noisy data due to bias in the regression [8]. Indeed, it is of note that the majority of research using DMD provides only a diagnostic analysis since reconstruction and

forecasting fail due to bias from noisy measurements. The BOP-DMD algorithm, however, removes bias and allows for a stable and robust forecasting technique. Indeed, as we will show, it provides strong forecasting capabilities with geographical and temporal UQ metrics. Moreover, it produces interpretable features, including geographical patterns that dictate the dominant dynamics of the flu, along with their temporal evolution.

5.2 Data and Methods

Data was collected from CDC FluView and WHO Flu databases [insert source]. Three datasets were used in this study: a worldwide set comprised of 168 countries, a statewide comprised of 49 U.S. states, and a citywide comprised of 336 U.S. cities. We used weighted influenza like illness (wILI) using the ratio of positive to processed influenza tests for each geographic region:

$$\text{wILI} = \frac{\text{Number of positive influenza Tests}}{\text{Number of processed influenza Tests}} \tag{5.2}$$

5.2.1 Worldwide Dataset

The worldwide dataset was taken from WHO Flu databases for all countries from years 2014 to 2019. This yearly range was selected in order to optimize the amount of available data within a span of consecutive years. As missing data was a significant issue in this set, we eliminated countries that had more than half of their data missing. In addition, countries with significant outliers were removed to avoid skewing the model. Even after this elimination, many of the remaining countries had a significant amount of missing points. To resolve this issue, we computed monthly wILI rates by taking monthly averages of the weekly wILI rates (Equation 5.2). Finally, any country that had a missing data point in its averaged monthly time series was eliminated from the dataset. These rates were then standardized for each country.

5.2.2 Statewide Dataset

The statewide dataset was taken from the Center for Disease Control and Prevention's FluView database for years 2010 to 2020. Similarly, weekly wILI rates were computed using

Equation 5.2. These rates were then standardized for each state.

5.2.3 Citywide Dataset

The citywide dataset was taken from [insert source] for years 2001 to 2010. Similar to the statewide dataset, weekly wILI rates were computed using Equation 5.2. Cities with significiant missing data were excluded from the dataset. After this exclusion, 336 major U.S. cities were used in this analysis. These rates were then standardized for each city.

5.2.4 Optimized Dynamic Mode Decomposition

We apply optimized Dynamic Mode Decomposition [9] with carefully curated initial conditions to create a data-driven model. This algorithm is based off of exact DMD which finds the best fit linear operator A that maps a point at time j, $\mathbf{x}_j = \mathbf{x}(t_j)$, to a consecutive point $\mathbf{x}_{j+1} = \mathbf{x}(t_{j+1})$ for $j = 1...m-1$. Specifically, we are looking for a linear operator A such that the following expression is minimized:

$$\|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F,$$

where X is comprised of vectorized snapshots, $\mathbf{x}_i = \mathbf{x}(t_i)$, for $j = 1...m-1$ time points. and $\mathbf{X}'$ are the same snapshots offset by a constant time increment, Δt, as shown below:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}$$

While DMD is a powerful computational tool, it is sensitive to noise found in complex datasets. Large amounts of noise can lead to biasing of DMD eigenvalues, resulting in a less accurate model. To circumvent this issue, we use optimized Dynamic Mode Decomposition (optDMD), which builds off of this algorithm while reducing bias due to noise. Using a variable projection algorithm, optDMD finds the best fit for α and B in the following nonlinear least squares problem:

$$\min \left\|\mathbf{X}^T - \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B}\right\|_F \qquad \text{over } \boldsymbol{\alpha} \in C^k, \mathbf{B} \in C^{lxn}$$

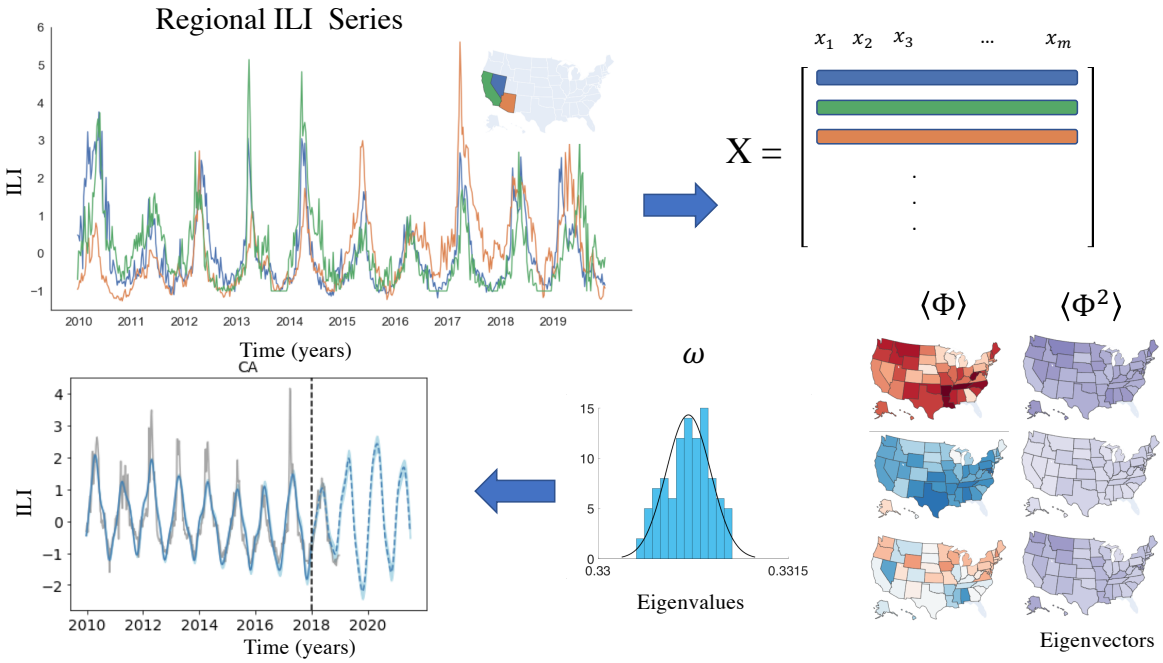

Figure 5.1: Overview: ILI time series for n regions (including California, Nevada, and Arizona) over m time points is stored in rows of matrix $\mathbf{X}$, where $x_i = x(t_i)$ for $i = 1...m$. The BOP-DMD algorithms are applied to $\mathbf{X}$ to produce $\hat{\alpha}$and$\hat{B}$, which are manipulated to get eigenvalue distributions and mean eigenvectors which are used to reconstruct and forecast the ILI series of any of the n regions. The gray time series in the bottom left figure is the actual time series while the blue time series is the reconstructed and predicted time series (before and after the dotted line).

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m]$ is comprised of vectorized snapshots over m time points and $\mathbf{\Phi}(\boldsymbol{\alpha}) = e^{\boldsymbol{\alpha} t}$ is Unlike exact DMD (see Figure 4.9), optDMD avoids issues like biased eigenvalues and damping without compromising interpretability and predictive power.

Initialization

As with many iterative methods, the convergence of the optDMD algorithm relies greatly on initial conditions. While DMD has limited predictive abilities when applied to noisy data, its eigenvectors still provide a wealth of information. We use the eigenvectors as initial conditions for optDMD as a guide to the true solution. As we are interested in studying the oscillatory dynamics (seasonality) of disease transmission over time, we constrain eigenvalues to the imaginary axis. While there is some degree of growth in flu dynamics over time, we choose to eliminate the real component of eigenvalues altogether. Over time, this real component can cause exponential growth in overall flu dynamics, leading to a less accurate model.

Mode Extraction

Upon convergence, optDMD algorithm outputs eigenvalues and eigenmodes (eigenvectors) of the dynamical system. The eigenmodes can provide information regarding regions that exhibit high variance over time while the eigenvalues can provide insight on the periodicity and magnitude of these eigenmodes. As we constrained the eigenvalues to the imaginary axis, the eigenvalues will have solely imaginary components. Thus, the eigenvalues will convey information regarding the periodicity, but not the overall growth of the system.

We can obtain these values by using the optimized parameters, $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{B}}$. The optDMD eigenvalues are given by $\lambda_i = \hat{\alpha}$ and optDMD eigenmodes are defined by normalizing each column of $\hat{\mathbf{B}}^T$:

$$\boldsymbol{\varphi}_i = \hat{\mathbf{B}}^T(:, i)/\|\hat{\mathbf{B}}^T(:, i)\|_2$$

Together, these eigenmodes and eigenvalues can be combined to create a low-rank approximation of the dynamical system as given by (5.1). This can be used to forecasting k time points ahead by simply inputing $t = m + 1...m + k$.

5.2.5 Reconstruction and Forecasting

As mentioned previously, we can reconstruct and forecast the time series, x(t) for $t = 1...m+k$ using Equation 5.1 with optDMD eigenvectors and eigenvalues. Reconstruction of time series used time points prior to the time point cutoff, $t = 1...m$, while forecasting used time points past the training time point cutoff, $t = m+1...m+k$. For reconstruction and forecasting with BOP-DMD, we use Algorithms 1 and 2 found in Chapter 4 for $t = 1...m+k$.

5.2.6 Uncertainty Quantification

With BOP-DMD (see Chapter 4), we can also gain an understanding of spatial and temporal uncertainty of these extracted modes by examining individual eigenvector and eigenvalue distributions, respectively. We find the optimal batch size, p, using a grid search with K = 100 trials. Via ensembling, we create distributions for each eigenvector Φ_i, λ_i, and b_i. Mean modes were created taking the mean of each Φ_i distribution while the mean lambda (periodicity after being converted) was computed by taking the mean of each λ_i distribution. The spatial uncertainty maps were created by taking the standard deviation of each Φ_i distribution and projecting these values onto the corresponding domain. Temporal uncertainty can also be obtained by computing the standard deviations of each $lambda_i$ distributions.

5.3 Results

The DMD modes of Equation 5.1, can be projected onto the maps for worldwide, state, and city data, forming 'eigenmaps'. These eigenmaps can be looked at individually and together, enabling both interpretability and reconstruction.

5.3.1 Interpretability

We first look at each eigenmap individually in order to see spatial patterns and gain insight of these dynamics. To further understand the oscillatory dynamics of these modes, we calculate the periodicity of each mode using imaginary components of the eigenvalues. The periodic nature of these modes is demonstrated in Figure 5.2 for the worldwide dataset.

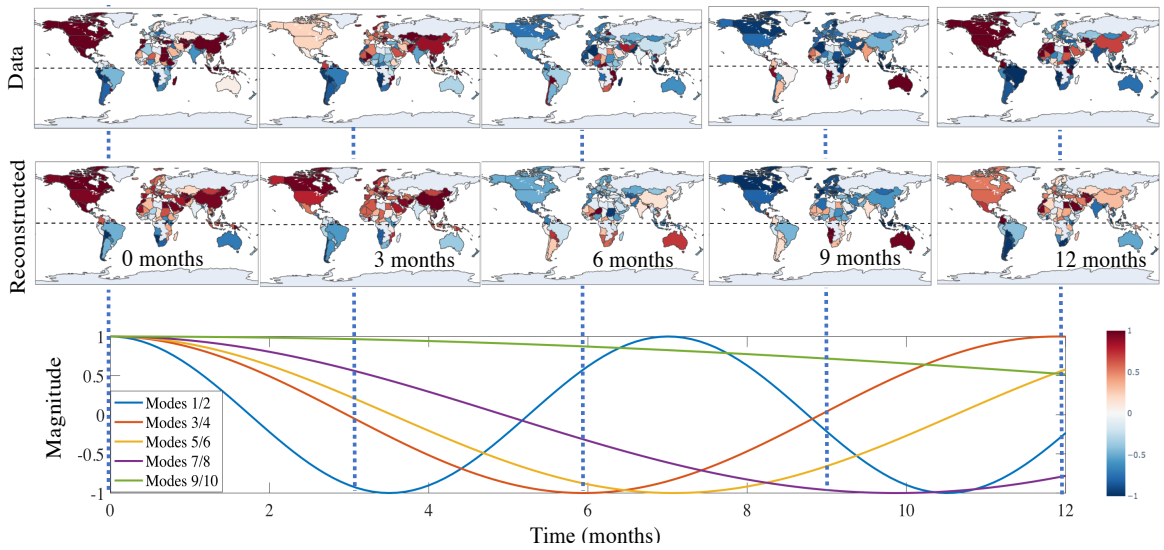

Figure 5.2: Reconstructed Modes for Worldwide Series. Top panel: Original standardized dataset for months 0 - 12. Spatial regions shaded red denote areas with a positive normalized ILI, while spatial regions shaded blue denote areas with a negative normalized ILI. Countries with large amounts of missing data in their time series were not included in the model and are shaded off-white. Middle panel: Reconstructed dataset for months 0-12. Bottom panel: Magnitude of modes as time progresses. Periodicities correspond to imaginary component of each paired eigenvalue. Each sinosoidal wave corresponds to a pair of conjugate modes.

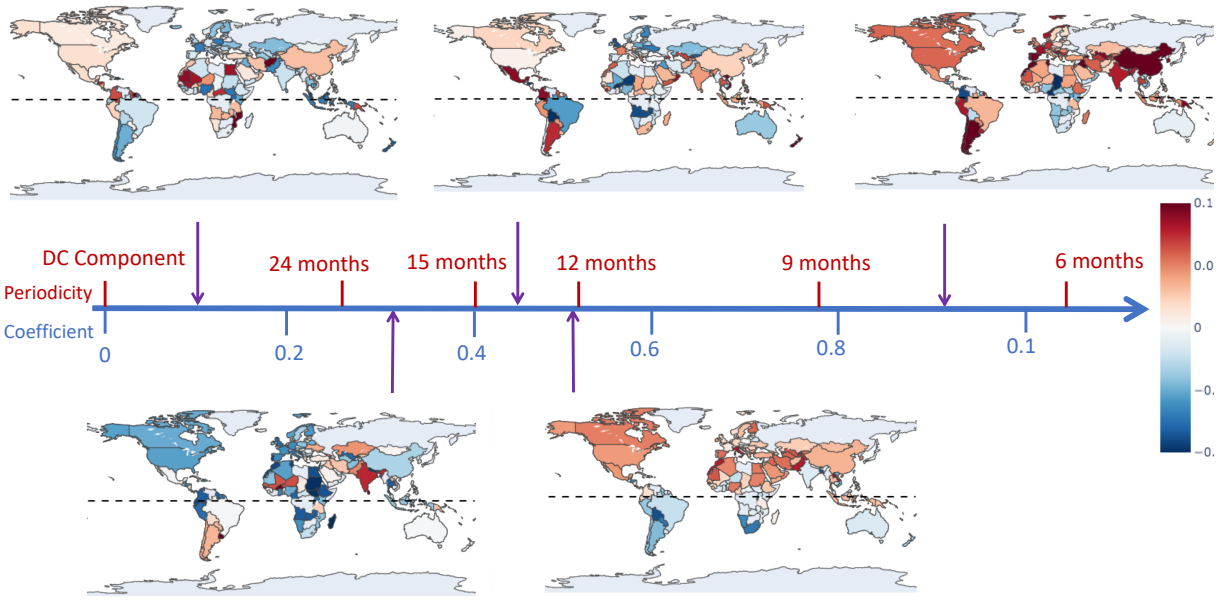

Figure 5.3: Periodicity of modes for the worldwide dataset. Each eigenmap corresponds to a spatial mode, with shading of countries corresponding to values of the eigenvector. Blue ticks on the line correspond to the coefficient, k, of the exp(ikx). Red ticks delineate the corresponding periodicity of these mode. Regions with high saturation of color are associated with high mode magnitude, or variance.

The magnitude of regions in these eigenmaps in Figures 5.3.1 - 5.3.1 correspond to variance over time. Thus, each mode portrays the oscillatory dynamic between regions of opposite signs with the corresponding periodicity.

Worldwide Data

As shown in Figure 5.3.1, the worldwide data set has 5 modes with periodicities: 7 months, 12 months, 14 months, 20 months, and 30 months. As previously mentioned, these eigenmaps show regions with high magnitude, or variance. The mode with periodicity of 7 months displays positive values of almost all countries except for those in central and southern Africa. Interestingly, in the map with periodicity of 12 months, there is a strong north-south divide. Specifically, there is a strong oscillatory dynamic between the Northern Hemisphere and the Southern Hemisphere. As the periodicity of this mode is about a year, these dy-

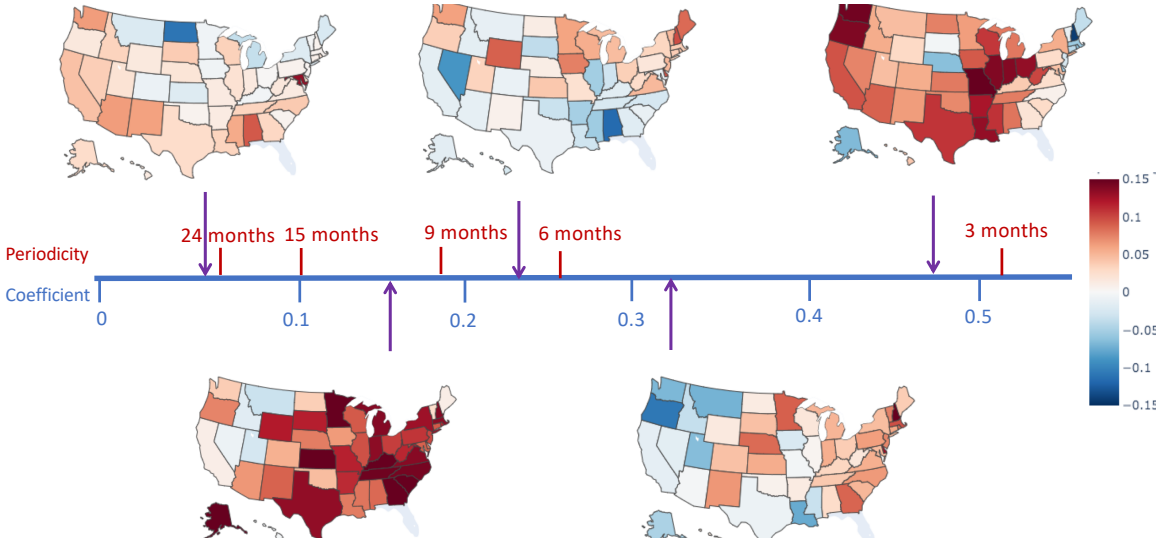

Figure 5.4: Periodicity of modes for the statewide dataset. Each eigenmap corresponds to a spatial mode, with shading of states corresponding to values of the eigenvector. Blue ticks on the line correspond to the coefficient, k, of the $exp(ikx)$. Red ticks delineate the corresponding periodicity of these mode. Regions with high saturation of color are associated with high mode magnitude, or variance.

namics could be explained by the seasonality of these regions. This pattern seems to even appear in the 20 month eigenmap. With generally negative eigenmap values in northern hemisphere and positive eigenmap values in the southern hemisphere this map appears to to complement the dynamics of those with the periodicity of 12 months. However, this spatial pattern seems to disappear in the map corresponding to 30 months. While there appears to be clusters among continents (e.g. North American countries, European countries) this general synchronicity across North Hemisphere countries is no longer preserved.

Statewide Data

In the statewide data (Figure 5.3.1), there are 5 modes of periodicities: 4 months, 5 months, 7 months, 12 months, and 25 months. In the eigenmap with periodicity of 4 months, there is a strong saturation of positive values in the Southern and Midwestern as well as in the Pacific Northwest and clustering of negative values in the Northeastern states. In the eigenmap

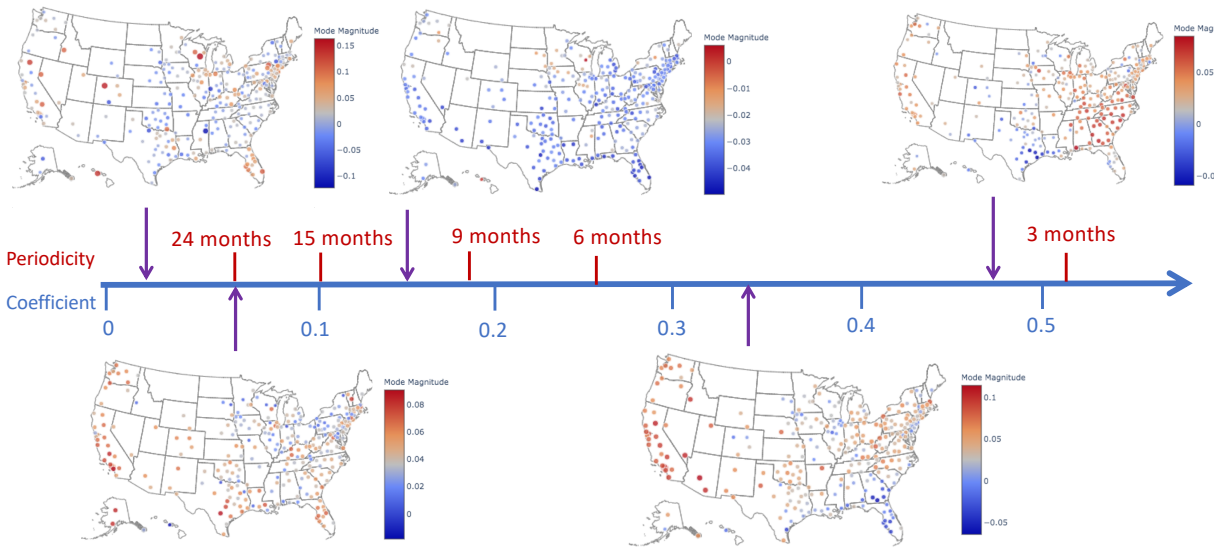

Figure 5.5: Periodicity of modes for the statewide dataset. Each eigenmap corresponds to a spatial mode, with shading of states corresponding to values of the eigenvector. Blue ticks on the line correspond to the coefficient, k, of the $exp(ikx)$. Red ticks delineate the corresponding periodicity of these mode. Regions with high saturation of color are associated with high mode magnitude, or variance.

with periodicity 5 months, there is a general deivide between eastern and western states. Specifically, we see a general clustering of states with positive values East of the Mississippi River and a general clustering of states with negative values in the western part of the U.S, particularly along the West Coast. Interestingly, in the eigenmap with periodicity of 7 months, there's a north/south divide. States within the northeast, northwest, and northern Midwestern states all have generally positive values whereas states in Southwestern and Southeastern U.S. have generally negative values. The eigenmaps with periodicities 12 and 25 months appear to complement the eigenmaps with periodicities 4 and 7 months.

Citywide Data

In the citywide data, there are 5 modes with periodicities: 4 months, 5 months, 12 months, 24 months, and 30 months. In the map with a periodicity of 4 months, there is a strong

divide between states directly west of the Mississippi and east of the Mississippi. In other words, flu transmission tends to have an oscillatory dynamic among cities east and west of the Mississippi. Similar to that of statewide data, the citywide modes corresponding to periodicities of 5 months and 24 months have a divide between the southwestern U.S. and the southeastern part of the country. In the map corresponding to periodicity of 12 months, we see a divide between a small clustering of northern Midwestern cities and the rest of the cities in the U.S. Finally, in the map corresponding to a periodicity of 30 months, we see isolated cities throughout the nation with high variance. While it is possible that these cities could be outliers in the data, it is more likely that this behavior may be due to noise in the data.

5.3.2 Reconstruction

The original maps can also be reconstructed using a combination of the aforementioned modes and eigenvalues. This reconstruction can be thought of a low rank approximation of the system. Using only 5 paired eigenvalues and eigenvectors, we can create the following reconstruction found in Figure 5.2. These maps are reconstructed for the first twelve months in three month increments. More importantly, in each of these maps we can identify the dominant modes. The oscillatory dynamic of these maps is created when the weight of these modes change over time, causing a shift in the dominant mode makeup. For example, Modes 1 and 2 (periodicity: 7 months) appear to be the dominant mode for time points 0 and 6 months, while Modes 3 and 4 (periodicity: 12 months) seem to be the dominant mode for time points 0 and 12 months.

This reconstruction is able to capture much of the major dynamics found in the data. In the zero and 12 months we see a general divide between the northern and southern hemisphere. Many of the countries in the northern hemisphere have positive normalized ILI while the southern countries have a negative normalized ILI. However, at the six month mark, we see a general reverse in dynamics, as the southern countries now have generally positive normalized ILI values while the northern countries have generally negative values. The dominant modes in all of these time points are Mode 1 and 2 with periodicity of

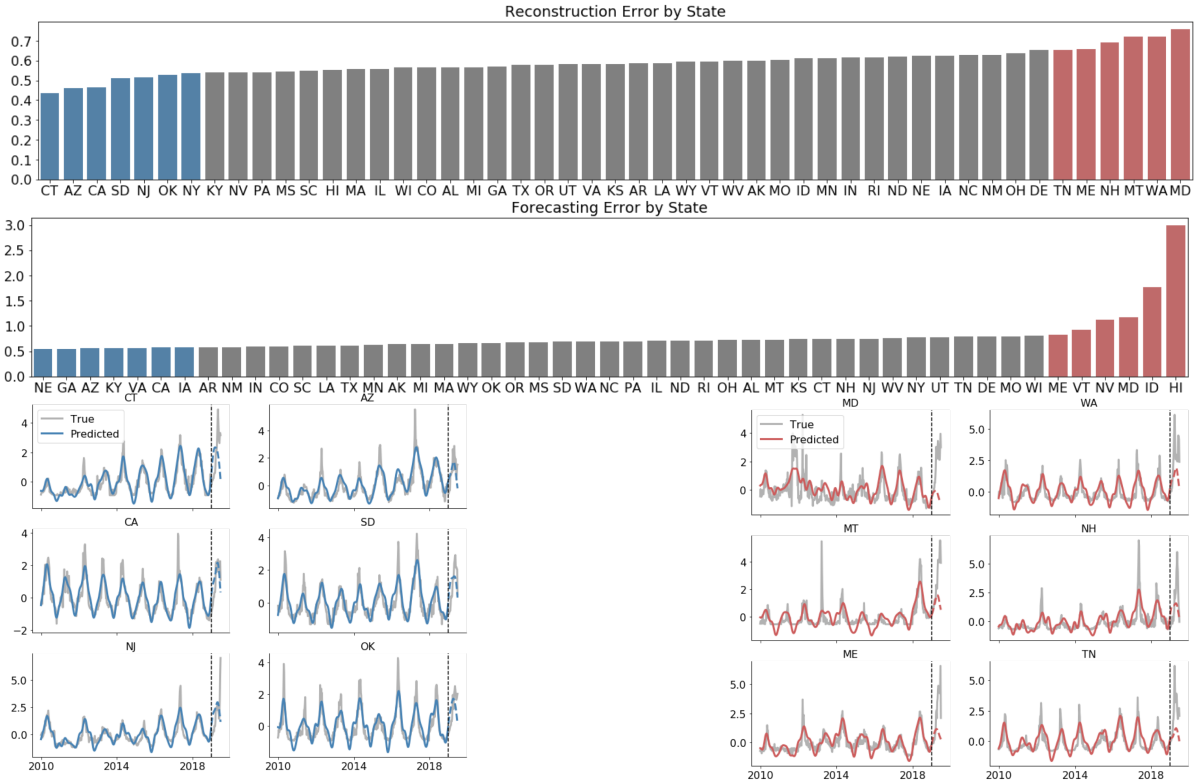

Figure 5.6: Reconstructed and forecasted plots of states with the highest (red) and lowest errors (blue). Top Panel: Errors for each of the states used in the statewide model. The six states with the lowest errors are highlighted in blue and the six states with the highest errors are highlighted in red. Bottom left: The predicted time series of states with lowest errors are portrayed in blue (solid lines for reconstructed series and dashed lines for forecasted series). The true time series are portrayed in gray.Bottom right: The predicted time series of states with highest errors are portrayed in red (solid lines for reconstructed series and dashed lines for forecasted series). The true time series are portrayed in gray.

approximately 6 months and Mode 3 and 4 with periodicity of 12 months. Importantly, the model is able capture the seasonal shift that is characteristic of flu transmission dynamics. While there are a few countries that don't fit the model well (i.e. Australia at 0 and 6 months), the model is able to reconstruct much of the dynamics.

5.3.3 Forecasting and Comparison of Models

As seen in Figure 5.10, the optDMD model does not decay over time due to bias and preserves oscillatory dynamics for the city, state, and worldwide data. The exact DMD model on the other hand, decays exponentially due to the inherent bias in its eigenvalues. The mean as well as the range of the worldwide series starts to decay about two years from the start date and continues to taper by the year 2017. A more rapid decay happens for the state and citywide data. The statewide level reconstruction decays within a year of the start date, tapering off at 2012 and continues to do so for the next seven years. Similarly, the citywide reconstruction continues to decay within a year of the start date and tapers off from 2002 to 2010.

We can extend this reconstruction of time series to forecasting. In Figure 5.6, we plot the times series (reconstructed and forecasted) for states with the lowest errors. The reconstructed time series is depicted with solid lines while the forecasted time series is depicted with dashed lines. We plot 6 states with the lowest reconstruction and forecasting error in Figure 5.6. In these states, the model is able capture the dynamics and periodicity with high accuracy. Importantly, the model is able to capture the seasonal nature for each state's transmission. While the model is unable to capture outliers like the magnitude of the nationwide 2015 outbreak, it is generally able to capture even nuances in transmission behaviors like those found in New Jersey with remarkable accuracy.

Even in the cases with high error, the model is generally able to reconstruct the time series with decent accuracy. In looking at the six states with the highest reconstruction and forecasting error, we see that the model has issues reconstructing time series with multiple outbreaks within one season like those found in Maryland during 2011. The model also has issues with unexpected outbreaks such as those found in Montana during 2013 and even

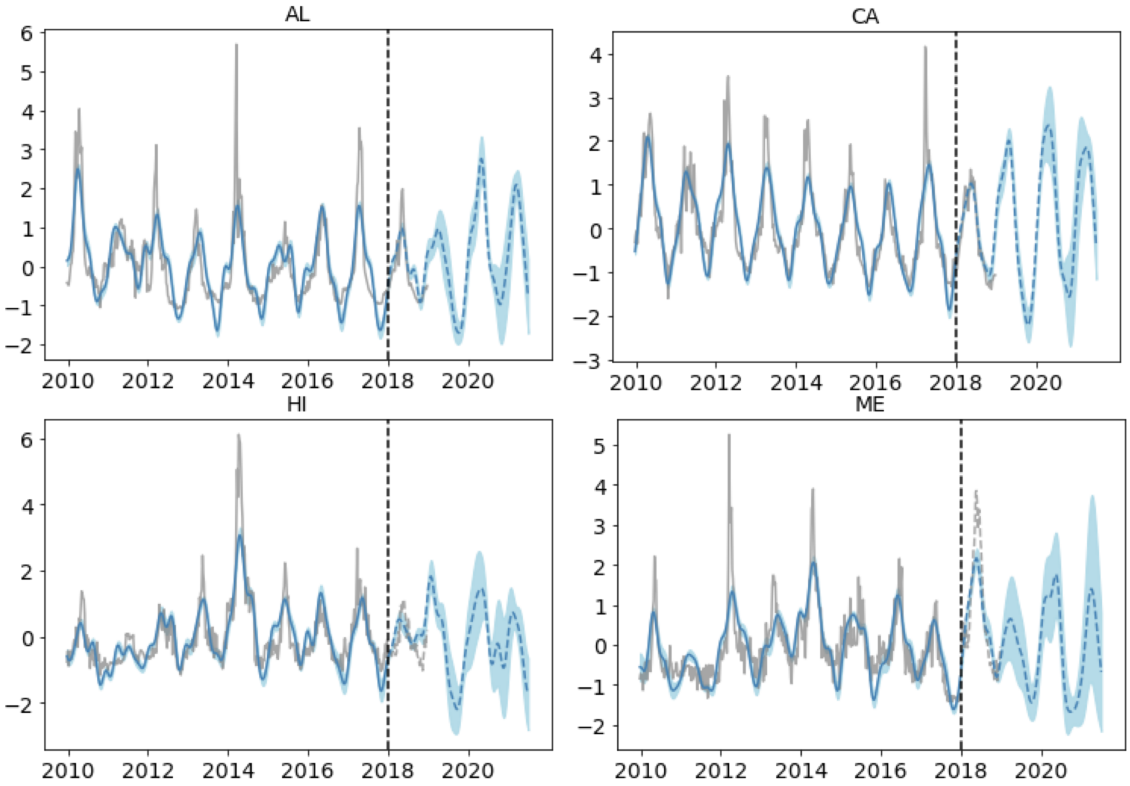

Figure 5.7: Reconstructed and projected weekly time series for four randomly selected states after ensembling optDMD with k = 100 trials for years 2010 to 2022. The blue line represents the mean values for ensembled optDMD during a given week.The shaded blue regions portray the 95% confidence intervals for the state forecast during a given week. The gray line represents the actual data. Reconstructed time series are displayed before the dotted black line (years 2010-2018). Forecasted time series are displayed after the dotted black line (2018-2022). One year of actual data (2018-2019) was not included in the reconstruction data in order to compare the forecast with the actual data.

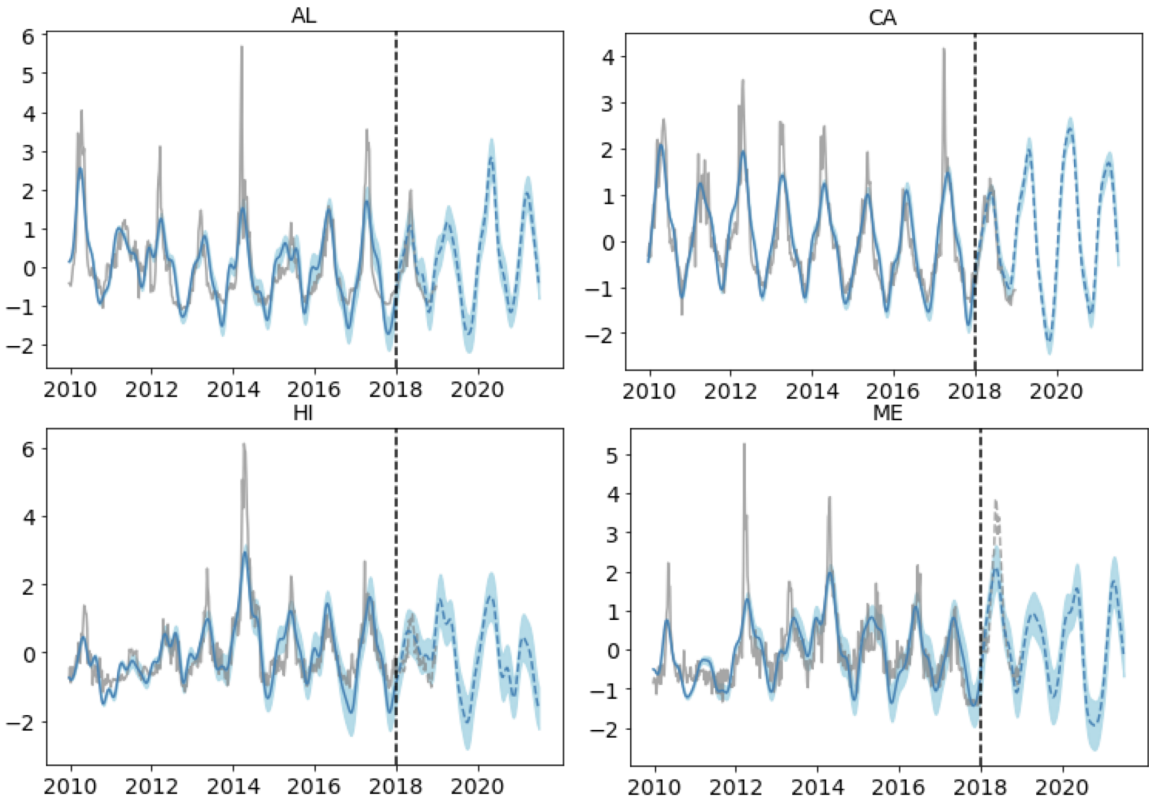

Figure 5.8: Reconstructed and projected weekly time series for four randomly selected states after applying BOP-DMD with k = 100 trials for years 2010 to 2022. The blue line represents the mean values for BOP-DMD during a given week.The shaded blue regions portray the 95% confidence intervals for the state forecast during a given week. The gray line represents the actual data. Reconstructed time series are displayed before the dotted black line (years 2010-2018). Forecasted time series are displayed after the dotted black line (2018-2022). One year of actual data (2018-2019) was not included in the reconstruction data in order to compare the forecast with the actual data. Note that this year of actual data is captured in the confidence intervals for each of these states forecasts.

those found in the forecasting dataset. However, while the reconstruction is not always able to capture the magnitude of the outbreak, it is generally able to capture the timing of the outbreak.

If we *ensemble* the optDMD model, we see that the forecasting abilities of the model generally increase. In Figure 5.7, we see that the ensembled model with k = 100 trials is not only able to continue to reconstruct the time series with high accuracy, but also forecast the first year past the training cutoff with remarkable accuracy for all four states. Of note, the time series for states, Hawaii and Maine, in Figure 5.7 have some of the highest forecasting and reconstruction errors in the optDMD model without ensembling (Figure 5.6).

To compare, we also forecast the state level time series using BOP-DMD (Chapter 4). Figure 5.8 demonstrates the forecasting and reconstruction abilities of BOP-DMD when applied to the same four states. Unsurprisingly, the mean value of BOP-DMD is quite similar to that of ensembled opt-DMD. However, the uncertainty quantification is drastically different. Specifically the 95% confidence intervals in the forecasted segments are significantly smaller in all four states. In contrast that of ensembled optDMD, the actual time series (gray) in the forecasting segments (2018-2019) are fully captured in the the confidence intervals for all for states.

In Figures 5.6 - 5.8, we see an increase in performance as bagging is integrated. While BOP-DMD proved to be the best tool for reconstruction and forecasting, we see how simply ensembling models can greatly improve general model robustness. BOP-DMD has the power to forecast without decay for long periods of time (in this case several *years*) while showing uncertainty quantification for each time point. Unsurprisingly, the confidence intervals during forecasting is much wider than those found in reconstructing the data and continue to widen as time progresses. Note that the model has relatively low uncertainty regarding the timing of the outbreak and high uncertainty regarding the magnitude of the outbreak.

5.3.4 Spatial and Temporal Uncertainty Quantification of Modes

As mentioned previously, we can apply optDMD to create the modes used to generate reconstruction and forecasting in Figures 5.2 to 5.3.1. However, even with this aforementioned

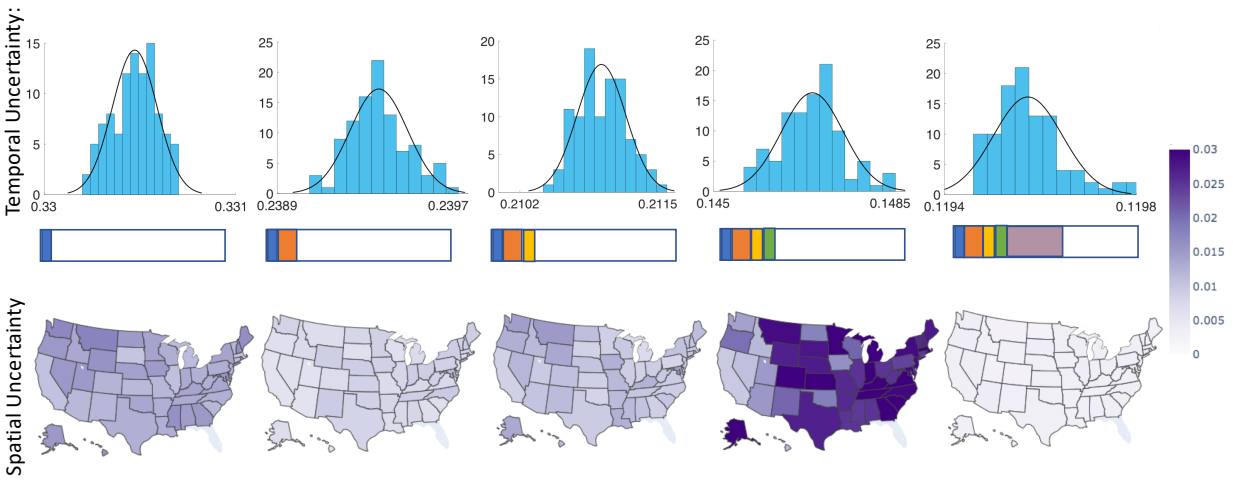

Figure 5.9: Example of Uncertainty Quantification on State-level Dataset: Top panel shows the eigenvalue distributions of the modes after ensembling. Below each mean mode is the average fraction of total amplitude. Bottom panel show variance of these modes or uncertainty of modes at a state-level. Areas with high magnitude (purple) correspond to areas with high uncertainty

initialization, the model is susceptible to convergence issues, making it difficult to assess how *certain* we are of this model.

Using BOP-DMD, we adapt the optDMD algorithm to compute an ensemble of models with an accompanying ensemble of future state predictions from a given collection of snapshots and a carefully selected initial condition (Chapter 4). In addition to forecasting, we apply the BOP-DMD algorithm to the datasets in order to compute spatial uncertainty quantification of these modes. In Figure 5.9, we show the results after applying 100 trials of ensembling to the statewide dataset. The ensembling provides a tool for estimating the variance the ensemble forecasts, thus providing variance metrics as a function of time. Moreover, we can get 95% confidence intervals allowing us to understand the degree of uncertainty. This can be directly used to compute variability in both the diagnostic and forecasting tool, giving critical UQ metrics.

As expected, Figure 5.9 shows similar behavior and periodicity in its mean modes as those found in Figure 5.3.1. The mode with periodicity of approximately 4 months has a similar concentration of eigenmap values in the Southern and Midwestern states while the

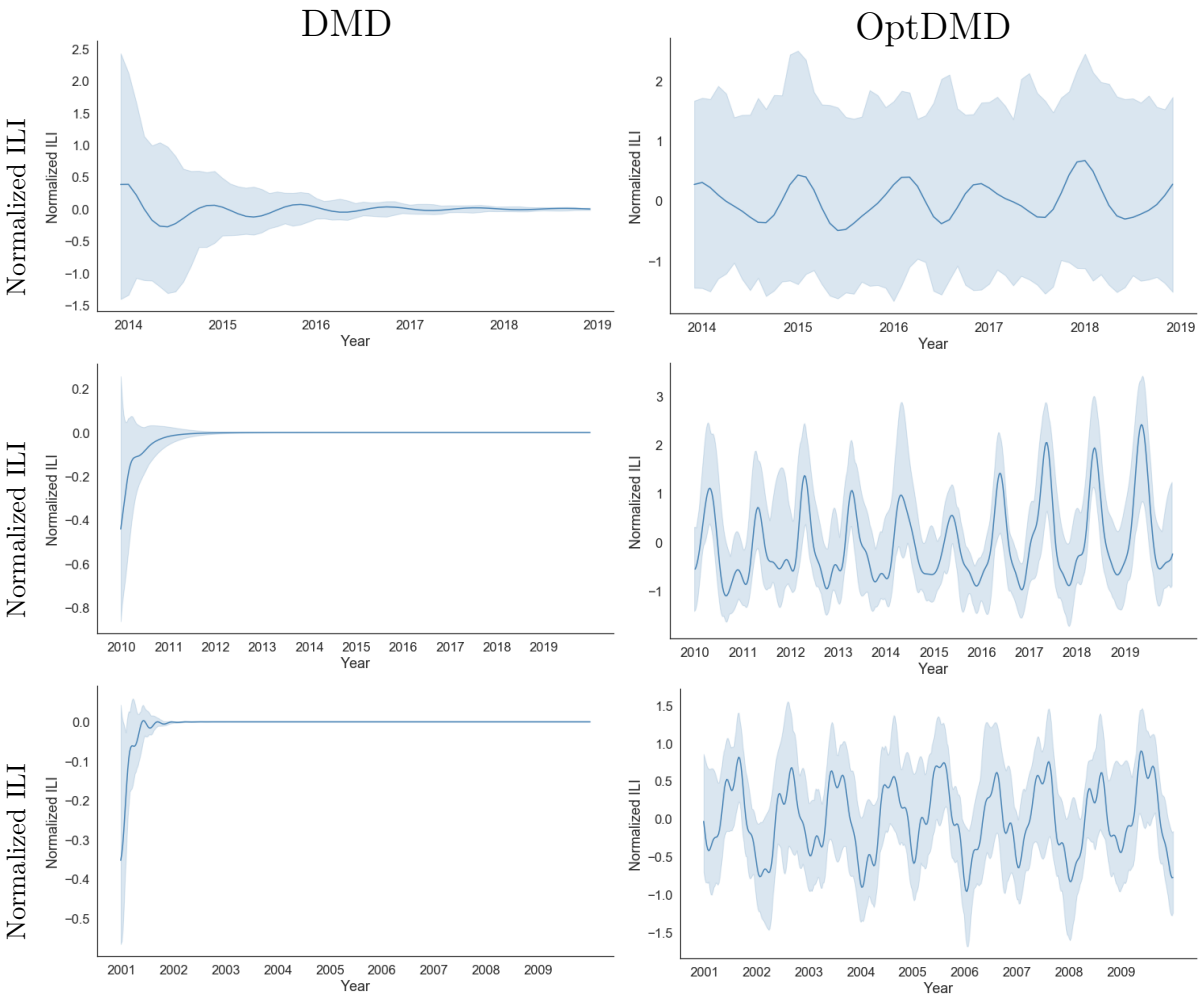

Figure 5.10: Reconstructed time series using DMD and optDMD. Solid line correspons to mean of entire time series across all states while the shaded regions correspond to the range of all states' time series. Top panel: Reconstructed worldwide ILI time series using DMD vs. optDMD. Middle panel: Reconstructed state ILI time series using DMD vs. optDMD. Bottom panel: Reconstructed city ILI time series using DMD vs. optDMD.

mode of periodicity of 6 months seems to have a similar divide between the eastern and western states. The mode with periodicity of about 7 months has a divide between the northern states and southern states. However, this divide is less distinct as that found in Figure 5.3.1. The last two modes in this figure are least similar to those found in Figure 5.3.1. However, this difference can be explained by the high uncertainty found in the eastern half of the U.S.

Interestingly, we see high uncertainty in Northern Midwestern states in the mode with periodicity of 4 months as well in the southeast portion of the U.S. in the mode with periodicity of 6 months. However, the mode with periodicity of 7 months appears to have fairly high spatial certainty. Only two states in this spatial uncertainty map had high uncertainty: Washington and Wyoming. The mode with a 22 month periodicity also has regions of uncertainty especially within the Midwest and Northeast.

5.4 Discussion

Together, optDMD and BOP-DMD are powerful and robust tools that not only have the potential to forecast and reconstruct noisy time series, but provide interpretable modes with respective peridiocities. Both of these methods create simple models by creating low rank approximations with minimal reconstruction and forecasting error, hence circumventing the issue of overfitting often found in time series modeling. In addition, BOP-DMD has the unique ability to provide uncertainty quantification both spatial and temporally. This added uncertainty quantification feature allows the user to gain perspective on which regions have high uncertainty as well as provide an idea of how *far* one can safely forecast within the bounds of a given threshold of certainty.

In this chapter, we apply these methods on three separate datasets. In each of these datasets, we generate eigenvectors or modes in order to gain an understanding of the many moving pieces in flu transmission. For example, in the worldwide data we see a mode with periodicity of 12 months that has a strong divide between Northern and Southern Hemisphere countries. With saturation of color in countries far from the equator, this mode appears to correlate with seasonal temperatures. When we put these modes together to form a time series reconstruction, it is clear that this 'seasonal' mode singlehandedly accounts

for the oscillatory dynamic we see in between the Northern and Southern Hemisphere at the 0, 6, and 12 month marks in Figure 5.2.

We also demonstrate BOP-DMD's ability to provide spatial uncertainty in Figure 5.9 using the state-wide data. Similar to the worldwide analysis, we gain perspective on the oscillatory dynamic of flu transmission across regions in the U.S. For example, we see a general asynchronicity between the South and Midwestern states with the rest of the U.S. We can then assess the certainty of these regions by examining the variance of these regions for each mode.

Finally, we showcase and compare both methods' abilities to reconstruct and forecast time series when these modes and corresponding eigenvalues and amplitudes are combined. Figures 5.7 show the increase in performance when models are simply ensembled. Peak performance and robustness however is achieved when BOP-DMD is applied to the dataset. In looking at Figure 5.8, we see that BOP-DMD has no problem forecasting states that optDMD struggled to forecast (such as Hawaii and Maine).

While DMD has been applied to epidemiological problems in the past [80], optDMD and BOP-DMD prove to be better tools for long term forecasting. In extending these reconstruction plots to several years, Figure 5.10 shows that optDMD preserves the oscillatory dynamics for all three datasets. DMD, on the other hand, starts off with similar oscillatory dynamics but exponentially decays resulting in a dampened time series for all cities, states, and countries. This dampened time series is a result of biased eigenvalues [11, 12, 29, 33, 48], an issue that optDMD (and hence, BOP-DMD) circumvents. Thus, while exact DMD proves to be a useful tool for diagnostics, it is not ideal for long term forecasting.

In this chapter, we compare the performances of optDMD and BOP-DMD when applied to three separate flu datasets. While it is much faster to run a single iteration of optDMD on the dataset, optDMD can run into convergence issues especially when applied to noisy datasets, resulting in an inaccurate forecast. More importantly, with a single measurement, it is hard to discern how *certain* we are of this forecast. BOP-DMD circumvents these issues with its bagging and ensembling approach and spatial and temporal uncertainty quantification, respectively. Specifically, its approach allows the model to exclude outliers that can skew model performance, and hence select the best model. With these features,

BOP-DMD proves to be better suited for the long-term forecasting when applied to each of these datasets.

The simplicity of the algorithm, its adaptibility, robustness to noisy, and stability of the forecast hold significant promise for the BOP-DMD method for application in epidemiology. Specifically, epidemiologial data sets are inherently noisy, thus making forecasting difficult. The incorporation of ensembling in BOP-DMD allows for the creation of stable models that can produce long-range forecasts without any significant bias generated from the measurement noise and corrupt measurements. Thus BOP-DMD has the potential to be used in a diversity of epidemiologial data, not just in the flu transmission setting. Importantly, the family of DMD algorithms also have a diversity of algorithmic variants appropriate for epidemiology, including for actuated systems [79], multi-resolution spatio-temporal behavior [64] and/or for robustness (not susceptible to corrupt data or outliers) [9]. Although not included here, they provide an interesting source of future directions for tailoring BOP-DMD in epidemiology.

5.4.1 Limitations

There are some limitations in the data collection that affect the accuracy of the algorithm. While the city and state level dataset was very comprehensive, the worldwide set had many instances of missing data. This is due to the lack of reporting and testing in some countries. Oftentimes, these countries do not have enough tests to confirm cases of influenza and run out mid-season. As a result, there are gaps in surveillance across months. Countries with large gaps in surveillance were excluded from analysis and countries with smaller gaps were averaged monthly. While this monthly averaging eliminates the need to exclude countries from the analysis, it does have the potential to remove some of the weekly dynamics. Only major cities were used for the city-wide set, which often aren't evenly distributed across the U.S. The east and west coasts have a high density of cities while the central region of the United States is relatively sparse. This non-uniform distribution of cities can affect the accuracy of model. That being said, in some cases the state-level is too coarse of a scale to measure these dynamics, especially for large states like California and Texas.

There are also some limitations regarding the implementation of BOP-DMD. Specifically, the efficacy and robustness of the BOP-DMD can be compromised when modes have a large amplitude , b. For example, 26 paired modes were required in order to model the state dataset with decent accuracy. However, although each generated distribution for the eigenvalues had a standard deviation that did not exceed $10^{-}3$ and the relative error between the mean and actual eigenvalue was 0.47% , there was a general growth in ILI as time progressed. Moreover, while this small error does not affect the general mode accuracy with modes with smaller amplitudes, the exponentiation of this small change with the multiplication by a large amplitude causes a fundamental change in model behavior and forecasting. In order to circumvent this issue, we instilled an amplitude cutoff (greater than 10^3). Modes that exceed this amplitude cutoff only used the mean amplitude and eigenvalues and did not draw from the respective distributions.

Chapter 6

CONCLUSION

In this thesis, we highlight the many ways in which forecasting can be applied to time series data. We demonstrate forecasting in the context of classification as well as predicting future time points in a spatiotemporal series. In addition, we showcase the robustness of time series forecasting methods when applied to spatiotemporal data inundated with compression-induced artifact, noise, and outliers. As mentioned in Chapter 1, the main goal in prediction and forecasting is to be able to generate $\mathbf{Y}$ from a given $\mathbf{X}$ (Equation 1.1, where

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_m) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k \times m}. \tag{6.1}$$

However, the scientific questions posed heavily depends on the nature of the data. Depending on these questions, $\mathbf{Y}$ can vary greatly. This thesis provides a comprehensive overview of time series forecasting for various forms of Y.

In Chapter 3, we provide an example of time series forecasting when X has a labeled outcome. Specifically, we apply signal processing and supervised machine learning methods in order to predict the pulse status of a given patient. In this example, $y \in \{1, -1\}$, denotes whether a given ECG of a patient corresponds to mechanical pulse. This particular classification problem is difficult to solve as data found in this field is often corrupted with severe artifact caused by chest compressions. Moreover, data in the field is often hard to obtain, limiting possibilities even in the large realm of supervised learning. We demonstrate how, even in the presence of corruption, methodologies such as Principal Component Analysis, Wavelet Transforms, Linear Discriminant Analysis can be applied to the data in order to uncover underlying dynamics. Although this compression artifact greatly overlaps crucial

heart data, these techniques are still able to classify a given patient's pulse status with high accuracy.

In Chapter 4, we demonstrate forecasting for a time series without a label. In this problem, we wish to predict the new few points in a spatiotemporal series. The outcome in this case, $\mathbf{Y}$, consists of the values for the next p time points in the time series:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_{m+1}) & \mathbf{x}(t_{m+2}) & \cdots & \mathbf{x}(t_{m+p}) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k \times p}, \tag{6.2}$$

where k,p ≥ 1 and $x(t_i)$ is the value of the series at the ith time point.

To solve this problem, we introduce a variant of the data-driven model, Dynamic Mode Decomposition. This method, BOP-DMD [86], builds upon another variant of the method, optDMD [8], by leveraging the simple, yet powerful statistical technique, bagging [16]. In addition, to resolve the common issue of convergence found in optDMD, the method carefully selects initial conditions by using DMD eigenvalues to seed BOP-DMD method. Unlike the rest of the DMD models, BOP-DMD brings uncertainty quantification to the table, enabling us to gauge how certain we are of the model. The spatial uncertainty quantification of this model provides basic diagnostics regarding which spatial regions have the most uncertainty, while the temporal uncertainty quantification of this model provides insight regarding certainty of mode periodicity. This data is first demonstrated on a simple example comprised of three eigenvalues as well as simulated vortex shedding and sea surface temperature data gathered by coastal sensors and satellites. In each of these cases, BOP-DMD not only is able to reconstruct the original data with remarkable accuracy but also forecast future time points years later.

Finally, in Chapter 5, we highlight the robustness of these methods by comparing the performances of exact DMD, optDMD, and BOP-DMD on flu datasets on three separate geographical levels, each replete with noise and missing data. When applied to datasets with high noise, exact DMD appears to fail in its abilities to forecast and even reconstruct the data. However, in each of these cases, we see that optDMD and BOP-DMD are able to

withstand the abundance of noise and outliers in the data and still parse out the general signal. Interestingly, with its added feature of ensembling and carefully selected initial conditions, BOP-DMD is able to forecast with higher accuracy states that optDMD struggled with. Moreover, unlike optDMD which provides a single forecasts, BOP-DMD provides a range of forecasts, enabling the user to assess how good their model. Despite the noisy nature of epidemiological data, BOP-DMD not only provides new tools regarding uncertainty quantification, but exhibits robustness that surpasses those of other DMD models.

BIBLIOGRAPHY

[1] (2017, july). noaa optimal interpolation (oi) sea surface temperature (sst) v2. [online]. available: https://www.esrl.noaa.gov/psd/.

[2] Roberta Affatato, Yongqin Li, and Giuseppe Ristagno. See through ecg technology during cardiopulmonary resuscitation to analyze rhythm and predict defibrillation outcome. *Current opinion in critical care*, 22(3):199–205, 2016.

[3] Alessandro Alla and J Nathan Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39(5):B778–B796, 2017.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[5] Anton Amann, Andreas Klotz, Thomas Niederklapfer, Alexander Kupferthaler, Tobias Werther, Marcus Granegger, Wolfgang Lederer, Michael Baubin, and Werner Lingnau. Reduction of cpr artifacts in the ventricular fibrillation ecg by coherent line removal. *Biomedical engineering online*, 9(1):2, 2010.

[6] Elisabete Aramendi, Sofia Ruiz [de Gauna], Unai Irusta, Jesus Ruiz, MÃÂª Fe Arcocha, and Jose Miguel Ormaetxe. Detection of ventricular fibrillation in the presence of cardiopulmonary resuscitation artefacts. *Resuscitation*, 72(1):115 – 123, 2007.

[7] Hassan Arbabi and Igor Mezić. Ergodic Theory, Dynamic Mode Decomposition, and Computation of Spectral Properties of the Koopman Operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, January 2017.

[8] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[9] Travis Askham, Peng Zheng, Aleksandr Aravkin, and J Nathan Kutz. Robust and scalable methods for the dynamic mode decomposition. *arXiv preprint arXiv:1712.01883*, 2017.

[10] Omri Azencot, Wotao Yin, and Andrea Bertozzi. Consistent dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(3):1565–1585, 2019.

[11] Shervin Bagheri. Koopman-mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.

[12] Shervin Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes, and Koopman spectrum. *Physics of Fluids*, 26(9):094104, September 2014.

[13] P.W. Bearman. On vortex shedding from a circular cylinder in the critical reynolds number regime. *Journal of Fluid Mechanics*, 37(3):577–585, 1969.

[14] Emelia J Benjamin, Salim S Virani, Clifton W Callaway, Alanna M Chamberlain, Alexander R Chang, Susan Cheng, Stephanie E Chiuve, Mary Cushman, Francesca N Delling, Rajat Deo, et al. Heart disease and stroke statistics?2018 update: a report from the american heart association. *Circulation*, 2018.

[15] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[16] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[17] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[18] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[19] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

[20] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(19):1–9, 2017.

[21] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015.

[22] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[23] Kathleen P Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019.

[24] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.

[25] Sheldon Cheskes, Robert H Schmicker, Jim Christenson, David D Salcido, Tom Rea, Judy Powell, Dana P Edelson, Rebecca Sell, Susanne May, James J Menegazzi, et al. Perishock pause: an independent predictor of survival from out-of-hospital shockable cardiac arrest. *Circulation*, 124(1):58–66, 2011.

[26] Jason Coult, Jennifer Blackwood, Thomas D Rea, Peter J Kudenchuk, and Heemun Kwok. A method to detect presence of chest compressions during resuscitation using transthoracic impedance. *IEEE journal of biomedical and health informatics*, 24(3):768–774, 2020.

[27] Jason Coult, Jennifer Blackwood, Lawrence Sherman, Thomas D Rea, Peter J Kudenchuk, and Heemun Kwok. Ventricular fibrillation waveform analysis during chest compressions to predict survival from cardiac arrest. *Circulation: Arrhythmia and Electrophysiology*, 12(1):e006924, 2019.

[28] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

[29] Scott TM Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):1–19, 2016.

[30] Anthony DeGennaro, Scott Dawson, and Clarence Rowley. Uncertainty quantification of the dynamic mode decomposition. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages R5–010, 2015.

[31] Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.

[32] N. Demo, M. Tezzele, and G. Rozza. PyDMD: Python dynamic mode decomposition. *The Journal of Open Source Software*, 3(22):530, 2018.

[33] Daniel Duke, Julio Soria, and Damon Honnery. An error analysis of the dynamic mode decomposition. *Experiments in fluids*, 52(2):529–542, 2012.

[34] Andoni Elola, Elisabete Aramendi, Unai Irusta, Javier Del Ser, Erik Alonso, and Mohamud Daya. Ecg-based pulse detection during cardiac arrest using random forest classifier. *Medical & biological engineering & computing*, 57(2):453–462, 2019.

[35] Andoni Elola, Elisabete Aramendi, Unai Irusta, Artzai Picón, Erik Alonso, Pamela Owens, and Ahamed Idris. Deep neural networks for ecg-based pulse detection during out-of-hospital cardiac arrest. *Entropy*, 21(3):305, 2019.

[36] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. *Journal of Real-Time Image Processing*, 2016.

[37] N. B. Erichson, S. Voronin, S. L. Brunton, and J. N. Kutz. Randomized matrix decompositions using R. *Journal of Statistical Softwar*, 89(11):1–48, 2019.

[38] Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, 2017.

[39] X Tan G Mamakoukas, M Castano and Todd Murphey. Local Koopman operators for data-driven control of robotic systems. In *Proceedings of "Robotics: Science and Systems 2019", Freiburg im Breisgau, June 22-26, 2019*. IEEE, 2019.

[40] X Tan G Mamakoukas, M Castano and Todd Murphey. Derivative-based Koopman operators for real-time control of robotic systems. *arXiv Preprint arXiv:2010.05778*, 2020.

[41] Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[42] Yushun Gong, Bihua Chen, and Yongqin Li. A review of the performance of artifact filtering algorithms for cardiopulmonary resuscitation. *Journal of Healthcare Engineering*, 4(2):185–202, 2013.

[43] Yushun Gong, Bihua Chen, and Yongqin Li. A review of the performance of artifact filtering algorithms for cardiopulmonary resuscitation. *Journal of Healthcare Engineering*, 4:807467, 2013.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[46] Markus Greiner, Dirk Pfeiffer, and RD Smith. Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. *Preventive veterinary medicine*, 45(1-2):23–41, 2000.

[47] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[48] Maziar S Hemati, Clarence W Rowley, Eric A. Deem, and Louis N. Cattafesta. Debiasing the dynamic mode decomposition for applied Koopman spectral analysis. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.

[49] Maziar S Hemati, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Debiasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.

[50] Nasser Heydari, Panayiotis Diplas, J Nathan Kutz, and Soheil Sadeghi Eshkevari. Modal analysis of turbulent flow near an inclined bank–longitudinal structure junction. *Journal of Hydraulic Engineering*, 147(3):04020100, 2021.

[51] Seth M. Hirsh, Kameron Decker Harris, J. Nathan Kutz, and Bingni W. Brunton. Centering Data Improves the Dynamic Mode Decomposition. *arXiv:1906.05973 [math]*, June 2019.

[52] Seth M Hirsh, Sara M Ichinaga, Steven L Brunton, J Nathan Kutz, and Bingni W Brunton. Structured time-delay models for dynamical systems with connections to frenet-serret frame. *arXiv preprint arXiv:2101.08344*, 2021.

[53] Andreas Holzinger. Trends in interactive knowledge discovery for personalized medicine: cognitive science meets machine learning. *The IEEE intelligent informatics bulletin*, 15(1):6–14, 2014.

[54] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[55] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2):024103, 2014.

[56] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz. Time-delay observables for Koopman: Theory and applications. *SIAM J. Appl. Dyn. Syst.*, 19(2):886–917, 2020.

[57] Alan A Kaptanoglu, Kyle D Morgan, Chris J Hansen, and Steven L Brunton. Characterizing magnetized plasmas with dynamic mode decomposition. *Physics of Plasmas*, 27:032108, 2020.

[58] Richard E Kerber, Lance B Becker, Joseph D Bourland, Richard O Cummins, Alfred P Hallstrom, Mary B Michos, Graham Nichol, Joseph P Ornato, William H Thies, Roger D White, et al. Automatic external defibrillators for public access defibrillation: recommendations for specifying and reporting arrhythmia analysis algorithm performance, incorporating new waveforms, and enhancing safety: a statement for health professionals from the american heart association task force on automatic external defibrillation, subcommittee on aed safety and efficacy. *Circulation*, 95(6):1677–1682, 1997.

[59] Richard Klabunde. *Cardiovascular physiology concepts*. Lippincott Williams & Wilkins, 2011.

[60] Monica E Kleinman, Erin E Brennan, Zachary D Goldberger, Robert A Swor, Mark Terry, Bentley J Bobrow, Raúl J Gazmuri, Andrew H Travers, and Thomas Rea. Part 5: adult basic life support and cardiopulmonary resuscitation quality: 2015 american

heart association guidelines update for cardiopulmonary resuscitation and emergency cardiovascular care. *Circulation*, 132(18_suppl_2):S414–S435, 2015.

[61] J. N. Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Oxford University Press, 2013.

[62] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[63] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[64] J Nathan Kutz, Xing Fu, and Steven L Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

[65] Heemun Kwok, Jason Coult, Jennifer Blackwood, Shiv Bhandari, Peter Kudenchuk, and Thomas Rea. Electrocardiogram-based pulse prediction during cardiopulmonary resuscitation. *Resuscitation*, 147:104–111, 2020.

[66] Heemun Kwok, Jason Coult, Mathias Drton, Thomas D Rea, and Lawrence Sherman. Adaptive rhythm sequencing: A method for dynamic rhythm classification during cpr. *Resuscitation*, 91:26–31, 2015.

[67] Heemun Kwok, Jason Coult, Chenguang Liu, Jennifer Blackwood, Peter J Kudenchuk, Thomas D Rea, and Lawrence Sherman. An accurate method for real-time chest compression detection from the impedance signal. *Resuscitation*, 105:22–28, 2016.

[68] Henning Lange, Steven L Brunton, and Nathan Kutz. From Fourier to Koopman: Spectral methods for long-term time series prediction. *arXiv preprint arXiv:2004.00574*, 2020.

[69] Adrian S Lewis and G Knowles. Image compression using the 2-d wavelet transform. *IEEE Transactions on image Processing*, 1(2):244–250, 1992.

[70] Mark S Link, Lauren C Berkow, Peter J Kudenchuk, Henry R Halperin, Erik P Hess, Vivek K Moitra, Robert W Neumar, Brian J O?Neil, James H Paxton, Scott M Silvers, et al. Part 7: adult advanced cardiovascular life support: 2015 american heart association guidelines update for cardiopulmonary resuscitation and emergency cardiovascular care. *Circulation*, 132(18_suppl_2):S444–S464, 2015.

[71] Heidrun Losert, Martin Risdal, Fritz Sterz, Jon Nysæther, Klemens Köhler, Trygve Eftestøl, Cosima Wandaller, Helge Myklebust, Thomas Uray, Sven O Aase, et al. Thoracic-impedance changes measured via defibrillator pads can monitor signs of circulation. *Resuscitation*, 73(2):221–228, 2007.

[72] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[73] Jordan Mann and J Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, pages 1–13, 2016.

[74] Dariush Mozaffarian, Emelia J Benjamin, Alan S Go, Donna K Arnett, Michael J Blaha, Mary Cushman, Sarah De Ferranti, Jean-Pierre Després, Heather J Fullerton, Virginia J Howard, et al. Heart disease and stroke statistics—2015 update: a report from the american heart association. *circulation*, 131(4):e29–e322, 2015.

[75] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[76] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[77] Bernd R Noack, Marek Morzynski, and Gilead Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.

[78] F Javier Ochoa, E Ramalle-Gomara, JM Carpintero, A Garcıa, and I Saralegui. Competence of health professionals to check the carotid pulse. *Resuscitation*, 37(3):173–175, 1998.

[79] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[80] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.

[81] Gengsheng Qin and Lejla Hotilovac. Comparison of non-parametric confidence intervals for the area under the roc curve of a continuous-scale diagnostic test. *Statistical Methods in Medical Research*, 17(2):207–221, 2008.

[82] ResuscitationGroup. labeledecg. Accessed: 2021-12-6.

[83] Martin Risdal, Sven Ole Aase, Jo Kramer-Johansen, and Trygve Eftesol. Automatic identification of return of spontaneous circulation during cardiopulmonary resuscitation. *IEEE Transactions on Biomedical Engineering*, 55(1):60–68, 2007.

[84] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D.S. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 645:115–127, 2009.

[85] Sofia Ruiz de Gauna, Unai Irusta, Jesus Ruiz, Unai Ayala, Elisabete Aramendi, and Trygve Eftestøl. Rhythm analysis during cardiopulmonary resuscitation: past, present, and future. *BioMed research international*, 2014, 2014.

[86] Diya Sashidhar and J Nathan Kutz. Bagging, optimized dynamic mode decomposition (bop-dmd) for robust, stable forecasting with spatial and temporal uncertainty-quantification. *arXiv preprint arXiv:2107.10878*, 2021.

[87] Isabel Scherl, Benjamin Strom, Jessica K Shang, Owen Williams, Brian L Polagye, and Steven L Brunton. Robust principal component analysis for particle image velocimetry. *Physical Review Fluids*, 5(054401), 2020.

[88] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, August 2010.

[89] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, November 2008.

[90] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[91] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.

[92] Yoshihiko Susuki and Igor Mezic. Nonlinear Koopman modes and coherency identification of coupled swing dynamics. *IEEE Transactions on Power Systems*, 26(4):1894–1904, 2011.

[93] Yoshihiko Susuki, Igor Mezić, and Takashi Hikihara. Coherent Dynamics and Instability of Power Grids. *repository.kulib.kyoto-u.ac.jp*, 2009.

[94] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, 96(3):033310, September 2017.

[95] Roy Taylor, J Nathan Kutz, Kyle Morgan, and Brian A Nelson. Dynamic mode decomposition for plasma diagnostics and validation. *Review of Scientific Instruments*, 89(5):053501, 2018.

[96] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[97] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[98] Meghana Velegar, N Benjamin Erichson, Christoph A Keller, and J Nathan Kutz. Scalable diagnostics for global atmospheric chemistry using ristretto library (version 1.0). *Geoscientific Model Development*, 12(4):1525–1539, 2019.

[99] Miles N Wernick, Yongyi Yang, Jovan G Brankov, Grigori Yourganov, and Stephen C Strother. Machine learning in medical imaging. *IEEE signal processing magazine*, 27(4):25–38, 2010.

[100] Ralph WCGR Wijshoff, Antoine MTM van Asten, Wouter H Peeters, Rick Bezemer, Gerrit Jan Noordergraaf, Massimo Mischi, and Ronald M Aarts. Photoplethysmography-based algorithm for detection of cardiogenic output during cardiopulmonary resuscitation. *IEEE Transactions on Biomedical Engineering*, 62(3):909–921, 2014.

[101] Ting Yu, Max Harry Weil, Wanchun Tang, Shijie Sun, Kada Klouche, Heitor Povoas, and Joe Bisera. Adverse outcomes of interrupted precordial compression during automated defibrillation. *Circulation*, 106(3):368–372, 2002.

Appendix A

CODE AVAILABILITY

Code for Chapter 3 (Pulse Prediction) can be found at:

`http://github.com/dsashid/PulsePrediction`

Code for Chapter 4 and 5 can be found at:

`http://github.com/dsashid/BOPDMD`