12-2011

Application of the Empirical Mode Decomposition On the Characterization and Forecasting of the Arrival Data of an Enterprise Cluster

Linh Bao Ngo
University of Arkansas, Fayetteville

Application of the Empirical Mode Decomposition On the Characterization and Forecasting Of the Arrival Data of an Enterprise Cluster

Application of the Empirical Mode Decomposition On the Characterization and Forecasting Of the Arrival Data of an Enterprise Cluster

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Engineering

By

Linh Bao Ngo
University of Arkansas
Bachelor of Science in Computer Engineering, 2003
University of Arkansas
Master of Science in Computer Engineering, 2005

December 2011
University of Arkansas

Abstract

Characterization and forecasting are two important processes in capacity planning. While they are closely related, their approaches have been different. In this research, a decomposition method called Empirical Mode Decomposition (EMD) has been applied as a preprocessing tool in order to bridge the input of both characterization and forecasting processes of the job arrivals of an enterprise cluster. Based on the facts that an enterprise cluster follows a standard preset working schedule and that EMD has the capability to extract hidden patterns within a data stream, we have developed a set of procedures that can preprocess the data for characterization as well as for forecasting. This comprehensive empirical study demonstrates that the addition of the preprocessing step is an improvement over the standard approaches in both characterization and forecasting. In addition, it is also shown that EMD is better than the popular wavelet-based decomposition in term of extracting different patterns from within a data stream.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

_____

Dr. Amy Apon

Dissertation Committee:

_____

Dr. Jia Di

_____

Dr. David Douglas

_____

Dr. Craig W. Thompson

Dissertation Duplication Release

I hereby authorize the University of Arkansas Libraries to duplicate this dissertation when needed for research and/or scholarship.

Agreed

_____

Linh Bao Ngo

Refused

_____

Linh Bao Ngo

Acknowledgements

First and foremost, I would like to offer my sincerest appreciation toward Dr. Amy Apon, my advisor. Without her support, advices, and patience throughout my academic career, the completion of this dissertation will not be possible.

I would like to thank my committee members for their valuable comments during the course of my Ph.D. career. I would also like to thank the faculty and staff of the Computer Science Computer Engineering Department at the University of Arkansas for their academic and non-academic guidance and support.

Acxiom Corporation was the initial sponsor for the project that leads to the materials in this dissertation. I would like to thank Dr. Larry Dowdy, Dr. Doug Hoffman, Dr. Baochuan Lu, Hung Bui, and Denny Brewer for their helpful suggestions. Particularly, Dr. Doug Hoffman was the one that points my attention to the Empirical Mode Decomposition technique. I thank Wesley Emeneker, Seth Warner, and Dr. Hiep Luong for giving me pointers in technical issues that I am not familiar with.

The success of my academic career is solidly based upon the love and support of my family. My mother taught me how to understand and love the simplicity and elegance of logic, while my father showed me how to be careful as well as be opened to different options. They believe in a strong education and have both worked very hard to support me and my siblings. I thank my wife, Huong Pham, for her love and support and for enduring my erratic behaviors during the writing of this dissertation. We are thankful for our children, William and Sophia, whose existences bring a lot of happiness into our life and make our work meaningful.

This material is based upon work supported by the National Science Foundation under Grant No. 0946726.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The focus of this dissertation is to answer the question whether Empirical Mode Decomposition (EMD) can bridge the advantageous characteristics of both workload characterization and workload forecasting in order to increase the efficiency of the capacity planning and performance evaluation as well as prediction processes.

Previous work has demonstrated the existence of patterns in the job arrivals of a computing system as well as the effects these patterns placed upon the performance of the system [1]. Since most of these jobs are generated by the people who work and study according to schedules, these patterns are created and affected by periodic events that influence these schedules. For example, a daily work schedule can create a workload that begins to increase in the morning and decrease at the end of the workday. Official holidays will contribute to a dip in workload during those days. If the total job arrival stream is considered a base signal, then the separation of different patterns containing different job arrivals can be considered to be the decomposition of the base signal. Instead of working on the entire job arrival stream, signal decomposition techniques can be used to identify the different patterns, and these resulting patterns can be used in workload characterization and prediction.

However, not many patterns are as clear cut and recognizable as the daily schedule or the holidays. Within each company, there exist different schedules and work cultures that contribute to the formation of job arrival patterns. Many of these patterns are also not constant and may only last for a short period of time. For example, the company might have a rush order that could increase the workload throughout one week. The appearances of these instances further increase the complexity of the patterns' frequencies. As a result, traditional signal decomposition

techniques that generate signals based on mathematical transformation might not represent these abnormal patterns well. In 1998, Huang et al [2] developed a time series analysis method called Empirical Mode Decomposition (EMD). The EMD technique can decompose any data set into a finite number of functions called Intrinsic Mode Functions (IMF) that exhibit symmetric with respect to their zero crossings. EMD has been successfully applied to the fields of health science and geology. In the field of information technology, the application of EMD has been limited to the area of image processing in which EMD is used for identification of edges inside a picture.

1.2 Motivation

Workload characterization and workload forecasting are two closely related fields. Both look for unique characteristics of a workload in order to generate a workload model. Figure 1.1 demonstrates the overall strategies and applications of the two processes. However, while workload characterization helps with capacity planning and performance measurements to answer hypothetical questions about the system' capability for different future workload scenarios, workload forecasting attempts to let the administrators to have a glimpse into the immediate future for the purpose of scheduling and resource allocating. As a result, workload characterization and workload forecasting differ from each other in the following areas:

- The length of time in which the models span.
- The time period in which the models are created.
- The level of accuracy of the models
- The modification capability of the models

For characterization models, the time span is usually as same as the time span of the original data. This allows the optimal capturing of the characteristics of the original data. As a result, the length of the original data is usually predetermined in order to generate the desired characterization models. In the case of forecasting models, the length of the original data also affects the length of the models. However, unlike the length similarity of characterization models, the time span of a

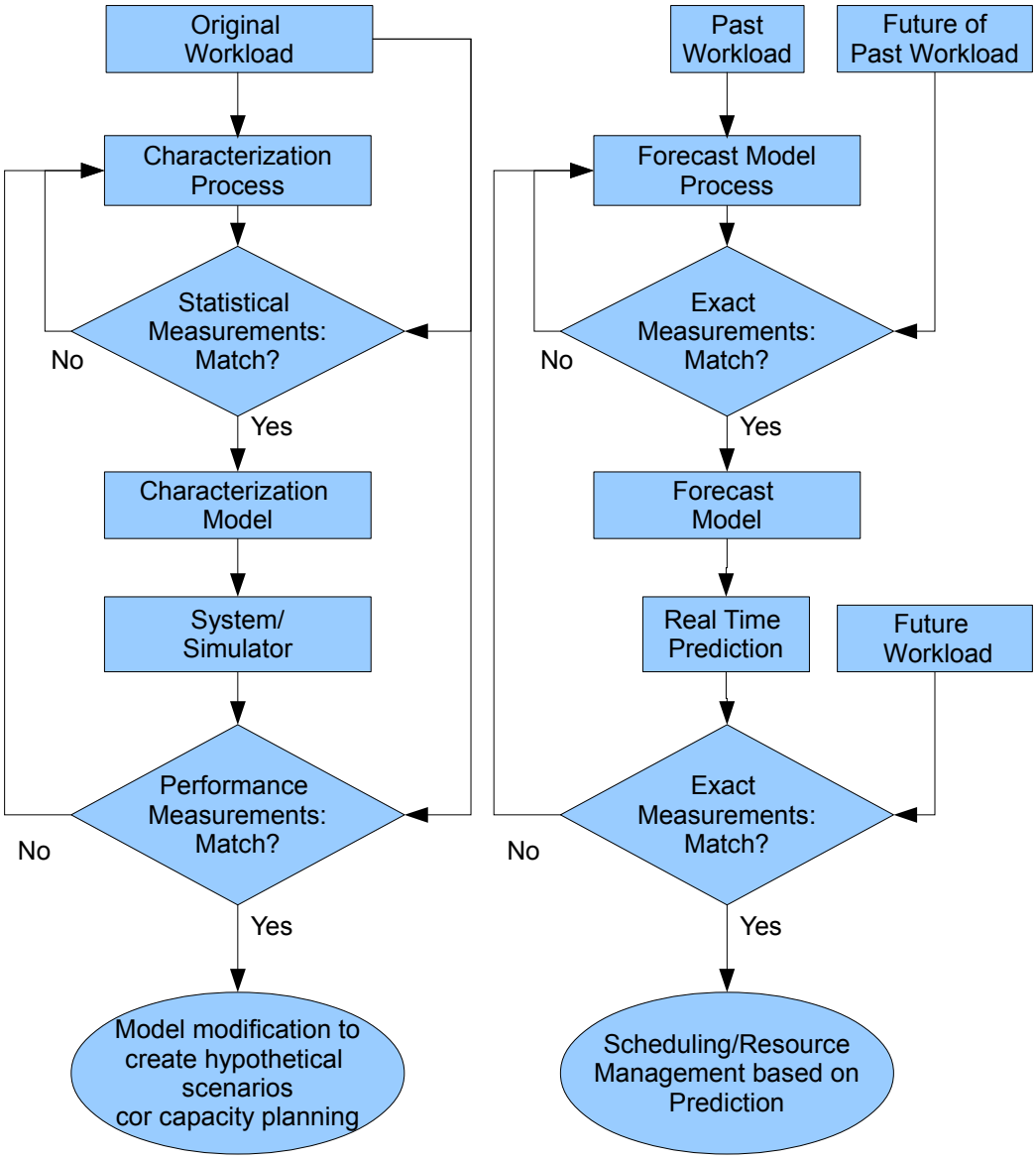

Figure 1.1: *Comparing overall strategies and applications of Characterization and Forecasting*

forecasting model is usually much shorter length than that of the original data. The dependency of the forecasting models' length on the original data's length is based on the patterns that the original data contains.

For characterization models, the time period in which the models are created is irrelevant. Rather, the size and the level of accuracy is important. This, in turn, relates to the modification capability of characterization models to help answer "What If" questions for capacity planning purposes. On the other hand, the forecasting models are usually generated for an immediate future based on recent workload activities or trends. There is no "What If" question. Rather, there is the expectation that the future true data will be similar to the synthetic data of the forecasting model. The purpose of the model is to help with the immediate scheduling and reallocation of computing resources.

The level of accuracy for characterization models usually does not carry the literal meaning of "accuracy." Rather, characterization models are evaluated upon their level at which they accurately capture the characteristics of the original system. The evaluation is done by comparing the statistical measurements of the models and the original data, as well as by comparing the performance measurements recorded when running the models and the original data through the same computing system (or the same simulator). For the purpose of capacity planning and performance evaluation, the accuracy of the performance measurements of a characterization model are more important than the level of detail. On the other hand, since forecasting models are trying to predict the future, they need to be as similar to the future workload as possible. This level of exact matching also implies that the performance measurements of a forecasting model should also match closely with those of the future workload.

Since one of the primary goals of characterization models is to generate a representative synthetic workload to replace the original workload in capacity planning and performance evaluation, a characterization model should be easy to modify to accommodate the "What If" questions related to the different effects that different hypothetical synthetic workloads would have on the system. The forecasting models are expected to generate synthetic workloads that not only match statisti-

cally but also carry the same similarities with the individual job entries of the future data. With this level of detail, it is difficult, if not possible, for forecasting models to be modified. In addition, the predictive goal of forecasting models also does not require them to be modifiable.

The literature surveys in chapter 2 shows that the approaches for workload characterization for the purpose of capacity planning, performance evaluation, and workload forecasting are different. While the dominant tools for workload characterization are probability distributions, workload forecasting relies more on time series analysis technique. Previous work has shown that EMD, a time series decomposition technique, can be used to characterize workload data from an enterprise cluster [3]. However, while offering a better level of accuracy, this technique requires a nontrivial amount of manual calibration in order to achieve optimal results. Consequently, it is often still more practical to utilize a traditional probabilistic-distribution based technique in characterizing workloads. To justify the use of EMD in analyzing computing workload data, additional work has been done to investigate the capability of EMD in the forecasting process of the workload [4]. The results of this work support the use of EMD as a preprocessing technique for data forecasting.

The above research form the basis for the motivation of this thesis. Workload characterization and workload forecasting each have their own advantages and disadvantages. A common approach that can bridge these characteristics will significantly increase the efficiency of capacity planning and performance evaluation as well as the prediction processes. That is, a workload characterization model that can reflect the actual future workload will not only help with capacity planning but also allow system designers to contemplate possible future scheduling and resource allocation. On the other hand, a forecasting model that has an extended range and modification capability will not only help with the immediate scheduling and resource allocation decisions but also allow the administrators to modify the model to test the immediate impact of these decisions. The successful application of EMD to workload characterization and the feasibility of using EMD in workload forecasting have made EMD an important component for such an approach.

1.3 Dissertation Roadmap

There are two parts to the approach of this problem. First, the EMD technique is tested for its capability for characterization. Performance measurements of synthetic workloads generated by EMD are compared to those of the original workloads and those of workload generated by traditional techniques. A similar process is also done for EMD's capability in forecasting. Second, an architectural design is created in which the EMD technique is the core component, from which outputs for workload characterization and forecasting are derived simultaneously. The remainder of the dissertation is organized as followed:

- Chapter 2 is a literature survey of the previous work on related topics. It also gives an overall description of the software and software packages as well as the data sets that were used in the dissertation.
- Chapter 3 studies application of EMD to the characterization process
- Chapter 4 describes the feasibility study in applying EMD as a preprocessing tool for forecasting purposes.
- Chapter 5 uses a set of well known time series forecasting techniques to compare the results between using EMD-based decomposed data and using the original data in the forecasting process. In this chapter, the usage of the popular wavelet-based decomposition is compared to those of the EMD-based decomposition.
- Chapter 7 highlights the contributions of this dissertation and discusses future work.

Chapter 2

Background

2.1 Introduction

Workload characterization and workload forecasting are two closely related fields in the sense that they both utilize unique characteristics of a workload to generate a synthetic model. However, while workload characterization helps with capacity planning and performance evaluation to answer hypothetical questions about the system's capabilities for different future workload scenarios, workload forecasting attempts to let administrators have a glimpse into the immediate future for the purpose of scheduling and resource allocation.

For computing system type workloads with jobs arriving from external sources, the process to characterize and forecast the job arrivals is particularly complicated. Due to the differences in scheduling practices, resource requirements, and work place behaviors, jobs arriving from different sources form different arrival patterns that are mixed together into the overall arrival stream of the workload. Consequently, different workloads require different characterizing, forecasting, and calibration strategies.

In this chapter, techniques used for characterization and forecasting of several workload types are surveyed. Different clustering and probabilistic distribution methods for the characterization of system workloads, networking and storage workloads, and high performance computing workloads are described. For workload forecasting, while references to traditional stochastic methods are mentioned, emphasis is placed upon the varieties of more recent forecasting techniques such as autoregressive analysis and neural networks. In addition, different signal decomposition techniques are reviewed, as they play an important role in preprocessing the data in order to improve the quality of the forecasting process. The remaining sections of this chapter are structured as followed:

- Section 2.2 discusses different characterization techniques on three different types of systems: single centralized system workloads, workload of networks and storage systems, and high performance computing workloads.
- Section 2.3 talks about workload forecasting and its two most common techniques: time series analysis and neural networks. This section also discusses the practice of using signal decomposition techniques to preprocess time series data before applying it to a forecasting mechanism such as time series analysis or neural networks.
- Section 2.4 describes the software and software packages that are used in this research work.
- Section 2.5 describes the Acxiom workload data set.

2.2 Workload Characterization

One goal of performance evaluation research is to identify the appropriate set of characteristics of synthetic test workloads for the purpose of benchmarking and assessing the performance of a computing system. In the early stages of computer hardware and software development, synthetic workloads were more finely tuned toward the individual job requests. That is, the synthetic workloads were created from a collection of synthetic programming jobs. These synthetic programming jobs were generated based on not only the equipment requirements but also on their algorithms and code implementations [1, 5, 6].

2.2.1 Overall Characterization Stategy

To address the issue of representativeness of the test workloads for performance studies, Agrawala, Mohr, and Bryant propose a set of models for characterizing the system workload [7]. A computing workload is a recorded set of user requests over a period of time to a specific system. The authors divide different characteristics of a user request into four primary variables:

- A vector X contains different resource requests (memory, hard disk, bandwidth) to the system.
- A time instant T signifies the time that the request arrives at the system.

- A location L shows the location from where the request originated.
- A flag F that identifies the request as timesharing, batch, or real-time processing.

Based on these characteristics, the authors propose three simple models: Type A models contain resource requests and flags (X, F), type B models contain time, resource requests, and flags (X, T, F), and type C models contain all of the characteristics (X, T, L, F). For type A models, a mixed distribution approach is used: the probability of a user request is calculated as the sum of the product of the probability of a class of requests and the probability of this user request appearing in this class for all the classes of the population. While the authors do not discuss any probability distribution function for the user requests, they mention K-means clustering as a viable method of identifying available classes from a workload and placing the user requests into these classes. For models of type B and C, the authors suggest the usage of time series analysis in order to identifying possible periodic behaviors. However, the authors make no further discussion about this issue.

Ferrari, in "Computer Systems Performance Evaluation," identifies eight "principal characteristics" of a workload model [8]. These characteristics are Representativeness (Accuracy), Flexibility, Simplicity of Construction, Compactness, Usage Costs, System Independence, Reproducibility, and Compatibility. Out of these eight, Ferrari considers representativeness and system independence as the two important problems in characterization. For representativeness, Ferrari recognizes the effects of the interactions between workload parameters on the system performance. As a result, a characterization that ignores these interactions (i.e.: arrival order) may fail to adequately represent the workload. System independence is another important characteristic of the workload model for cases where different systems are evaluated. The principles of artificial workload design are reviewed by Ferrari in [9]. In this paper, the author outlines five common steps usually taken to construct a synthetic workload:

1. Identifying of basic components
2. Choosing parameters to be characterized
3. Measuring chosen parameters, placing measured values into tuples
4. Applying statistical techniques (sampling and clustering methods) to the tuples

5. Replacing tuples by the workload components whose parameters equal or characterize the tuples.

Ferrari states that the applications of the above procedure, although differing from each other in a variety of ways, raise three problems [9]:

- There is no scientific way to identify/quantify the impact of resource components on the system performance in order to choose accurately
- The workload model cannot be guaranteed to be correct
- With few exceptions, the design techniques commonly used in step 4 usually ignore the temporal information of the workload (arrival time, component sequences, and mix compositions).

While offering queuing network models and a performance-oriented solution for the first two problems, Ferrari suggests that further studies are needed for the approximation of workloads whose accuracy depends on temporal information.

2.2.2 Basic Component Identification

The nature of basic component identification is to identify similar observations from which a baseline model can be derived. This process can also be called a clustering process. [10] provides an in-depth survey of different varieties of clustering techniques. Among these techniques, the partitional clustering, whose representative is K-means analysis, is concluded to be most effective in dealing with large data sets. The general formulation of this technique can be described as follows [11]: Let $z_1, z_2...$ be a random collection of points in E_N. Let $x_1, x_2...$ be a given k-tuple in E_N. A minimum distance partition of E_N is defined by:

$$S_1(x) = T_1(x)$$

$$S_2(x) = T_2(x){S_1}'(x)$$

...

$$S_k(x) = T_k(x){S_1}'(x){S_2}'(x)...{S_{k-1}}'(x)$$

with

$$T_i(x) = \xi : \xi \in E_N, |\xi - x_i| \leq |\xi - x_j|, j = 1, 2, .., k$$

In other words, the set $S_i(x)$ contains the points in E_N nearest to x_i, and the points at the same distance to different x values are arbitrarily assigned to the set of lower index. Alternative approaches to implement the K-means technique have been described in the literatures. One approach to implement the K-means technique can be summarized as followed:

- Step 1: Randomly divide the set of data points into k partitions.
- Step 2: Finding the centroid for each partition. The most straightforward technique is to find the average value of all the coordinates on each dimension (each attribute). Depending on the dataset, weights can be applied to this step.
- Step 3: For each data point, measure the distance between the data point to the centroid of each partition. If the distance between the data point and its own partition is not minimum, the data point is moved to the partition where this distance is minimum.
- Step 4: Repeat from step 2 until there is no data point is moved. There are some cases where a data point can keep switching between two different partitions. In this case, a stopping condition can be applied such that the procedure is stopped when the amount of data points that change partition is reduced to a certain number.

2.2.3 Statistical Modeling

Calzarossa and Serazzi also complete a survey on workload characterization in [12]. Generally speaking, the primary steps to generate a workload model still involve the identification of basic components and their parameters and measurements and statistical analysis of chosen parameters similar to [9]. However, compared to [9], there are more statistical analysis steps, including preliminary analysis, distribution analysis, sampling, static analysis, and dynamic analysis. The

criteria for the evaluation of representativeness are also established earlier in the process. The survey also discusses the three types of systems on which workload characterization are usually done [12]. These systems are centralized systems (batch and interactive systems and database systems), network-based systems, and multiprocessor systems (parallel systems and supercomputers). Among these systems, the survey concludes that only the problems for the centralized systems are well understood. The architectures of the other two system types were relatively new and more investigations into characterization of these systems' workloads were needed.

Centralized Systems

In characterizing eight months worth of data from the University of Maryland machines, Mohr observes the hourly, daily, weekly, and monthly variations of the jobs generated by students and researchers [13]. The author identifies the existence of several oscillatory components within the workload and acknowledges the necessity to maintain the data throughout the complete cycle of an oscillation in order to accurately capture the characteristics of this oscillation. In addition, Mohr also cautions about using the characterization of one measurement session of a monotonically increasing workload for near future analysis.

Calzarossa and Serazzi propose a numerical fitting technique for arrival patterns [14]. Through this research, the authors also approach the issue of identifying representative arrival patterns of the analyzed workload. Using 14 one-day traces, the authors derive a polynomial arrival rate function. The optimal degree of the polynomial function in this case is determined to be 8. After acquiring the overall arrival rate function, further efforts are made to identify possible different arrival patterns hidden within this main function. By using a k-means algorithm, the authors observe three clusters whose data have the same polynomial function degree as the original arrival rate function but with different coefficients. These three clusters can be described with respect to the users' behaviors: cluster 1 is the normal user behavior, cluster 2 matches with the afternoon-off days of the administrative offices, and cluster 3 describes the workloads following holidays or down time periods.

Network and Storage Workload Characterization

With the increases in power and complexity of computing technologies, the importance of dynamic analysis (arrival patterns) in performance evaluation and capacity planning techniques are emphasized not only on full computing systems but also on subcomponents of a system, especially storage I/O access and network connection in particular. The classic paper by Leland et al identifies a significant characteristic of network packet traffic: self-similarity ("burstiness") [15]. By analyzing LAN network traffic collected over three years, the authors discover that different time scales of the traffic of a network carry similar structural burstiness behavior. This degree of self-similarity is measured by the Hurst parameter and typically depends on the level of utilization of the network. In addition, aggregating streams of different traffic does not reducing the degree of burstiness through Poisson-like superposition but increases this burstiness instead [15]. Based on [15], Paxson and Floyd further question the conventional usage of Poisson processes in modeling network traffic [16]. The authors recommend a thorough self-similarity study before deciding on a distribution model for the analysis of network traffic.

For storage workloads, Ruemmler and Wilkes demonstrate the burstiness of I/O accesses in 1993 [17]. In their research, Ruemmler and Wilkes analyze the traces of three different HP-UX systems over a four-month period. Notably among the results are the facts that the mean request queue lengths range from 1.7 to 8.9 entries and while the 95th percentile queue lengths are around 89 entries, the maximum queue length reaches the size of over 1000 entries. Although the authors do not directly address the issue of characterizing I/O workloads in this paper, they provide an important foundation for later modeling and simulation techniques on I/O workloads.

In his discussion about generating representative synthetic storage workloads, Ganger shows that simple assumptions and traditional Poisson processes create a large margin of error for the synthetic trace [18]. In fact, Ganger states that the disk workloads are "neither independent nor exponentially distributed." Another observation is that a system level modeling approach [19] can be more appropriate rather than a synthetic trace generator due to the narrow scope of standalone I/O subsystem models.

Gomez and Santonja demonstrate in [20] that disk-level I/O requests are self-similar in nature.

Using adjusted range (R/S) statistical analysis to derive the Hurst parameters in two sets of I/O traces, the authors confirm that the read and write arrival patterns are self-similar in nature. In order to model the disk arrival pattern, the authors look at how self-similar Ethernet traffic can be modeled based on the ON/OFF behavior of their traffic sources. Although there is no information about the sources of disk arrivals, the authors suspected that for a source, the reading and writing activities compose the ON period while the file preparations compose the OFF period. Based on this assumption, Gomez and Santonja create a model using heavy-tailed Pareto distribution with the parameter $alpha$ extracted from the Hurst parameter. The resulting model exhibits the self-similarity and burstiness of the original traces. However, the authors do not provide any further validation in term of statistical measurements and performance of the synthetic model.

In [21], Gomez and Santonja perform a more detailed study on the composition of the traces. They divide the processes (the sources) that generate disk accesses into two types: permanent and vanishing. The permanent sources are responsible for approximately 88% of the total accesses and are active for most of the time during the period of observation. On the other hand, the vanishing sources appear randomly and generate much less disk access. Based on this information, Gomez and Santonja further improve their model by using the ON/OFF sources model to synthesize the permanent sources and the M/G/∞ Cox's queuing model with for the vanishing sources. Again, the authors only emphasize the demonstration of the self-similarity and burstiness in their resulting synthetic model.

Acknowledging the burstiness and self-similarity of disk I/O traffic, Wang proposes a modeling algorithm called the b-model to generate synthetic I/O data [22]. Related to the "80/20 law" in databases, the b-model uses a bias parameter b to identify the distribution of requests within a given time interval. For example, b = 0.8 means that 80% of requests will take place in one half of the interval, and the remaining 20% will take place in the other half. Recursively, this process is performed until all the requests had been placed. Using entropy [23] to fit the synthetic model with the original trace, Wang creates a traffic model that generates data with burstiness and self-similar measurements as the original traces. The synthetic data are also self-similar and display behaviors

close to those of the real data. For inter-arrival time distribution and queue length, the synthetic trace exhibits similar shapes and forms as the original trace but still contains a large margin of error in term of numerical accuracy. In a later research, Wang, Ailamaki, and Faloutsos focus on generating a model that can capture the spatio-temporal behavior of the data: the relationship between the arrival time of the disk request and the location on disk of the request [24]. The authors propose a preliminary model called the i-model. This model has an underlying assumption that the distributions of arrival time are independent from each other, thus these two distributions can be "multiplied" to form a spatio-temporal synthetic trace. In reality, the actual data traffic does have a strong spatio-temporal correlation, and this invalidates the i-model. Next, the authors develop an improved model called the PQRS model. Although the basic partitioning scheme is similar to the partitioning idea of the b-model, the PQRS model extends its reach to two dimensional space, which allows it not only express burstiness and self-similar but also capture the spatio-temporal behavior of real traffic data [22].The PQRS model also carries the same strengths and weaknesses as the b-model. That is, the PQRS synthetic trace maintains the self-similarity and burstiness as well as the form of inter-arrival time and queue length as the original trace. However, the margin of error is still relatively high at around 30%.

High Performance Computing (HPC) Characterization

Parallel workload modeling has always been a fascinating problem. In [25], Sevcik relies on the different level of details of parallelism in an application in order to characterize it. Calzarossa et al further divide a parallel application into three related layers (the application, the algorithm, and the routine). These layers are then characterized based on their functional, sequential, parallel, and quantitative descriptions [26]. Similar to Sevcik, Downey also focuses on the level of parallelism when developing a workload model for parallel computers [27]. Jann, Pattnaik, and Franke approach the problem of characterizing the workloads for a large massively parallel processor (MPP) supercomputing center from a higher level: the inter-arrival time and service time distributions for jobs [28]. The authors choose a phase type distribution called the Hyper Erlang distribution of Common Order that could fit exactly the first three moments of the observed workload. Even

though the model still sometimes produces a relatively high error for the fourth moment, the authors, based on the previous works on Schassberger [29, 30, 31], conclude that their model is adequate for its purpose. In later research, this model is successfully utilized to build a synthetic workload in order to analyze different parallel job scheduling schemes for the Sustained Stewardship TeraOPS (SST) "Hyper-Cluster" [32]. Downey and Feitelson observe the workload traces of several supercomputers and discuss the possible issues in characterizing these workloads [33]. While recognizing the usefulness of lower-level detailed modeling techniques, the authors comment that this approach increases complexity as well as requires a lot more information from the traces in order to successfully generate a workload model. Downey and Feitelson highlights possible problems in characterization such as moment-based statistics, goodness of fit, distributions with mathematical models, time scale, and sample size.

The characterization of parallel workloads is highly dependent on the information about the jobs. In the case where the jobs' algorithms are known, it is possible to characterize the whole workload by analyzing and combining the run-time characteristics of individual algorithms. An example of this type of workload would be an academic or research system, where there are fewer jobs whose applications are more parallelized and take longer to run and the algorithms are usually based on similar common foundation theorems [34], [35], [36], [37]. Without the detailed knowledge about the algorithms, metric measurements of the jobs such as arrival time, run time, and amount/type of computing resources requested are utilized in partitioning the workload into groups. The resulting characterization is usually a set of statistical distributions, each distribution representing a group [38, 39, 40]. After the grouping and clustering process, the only usable parameter left for statistical analysis is typically the arrival time. One way to characterize the arrival time is to view it as a time series through the set of calculated interarrival times or arrival time plot. An example of characterizing arrival time using time series analysis techniques is the work of [41], where the ARMA and ARIMA models are utilized to develop a class of traffic models. Further details on time series analysis are discuss in section 2.3 of chapter 2. Another approach is to fit the set of interarrival times and arrival time plot to a statistical distribution. In [40], by combining

the two-stage hyper-exponential distribution model of [42], Lu presents an empirical model with an optional daily arrival pattern.

2.2.4 Summary

The early works outlines the five main steps of the characterization process as the identification of basic components, the choices of parameters to be characterized, the measurements of chosen parameters and placements of measured values into tuples, the application of statistical techniques, and the replacement of tuples by the workload components whose parameters equal or characterize the tuples [9]. While workload data can contain information about many components and parameters, most, if not all, of the characterization works involve timing-related parameters such as arrival time, arrival pattern, wait time, disk access time, and so on. With this in mind, the focus of the workload characterization is usually placed on the fourth step, which can include a variety of statistical analysis steps. Since the timing components are highly dependent on the behaviors of the systems' users, different system will exhibit different timing characteristics. Consequently, the statistical analysis step of the characterization process will usually be quite different on a case by case basis.

2.3 Workload Forecasting

In this section, we will look at the two main approaches to workload forecasting: time series analysis and neural network approach. We will also discuss how signal decomposition can also be used in workload forecasting in order to improve the forecast results. Finally, the Empirical Mode Decomposition, an alternative to the traditional signal decomposition, is investigated.

2.3.1 Time Series Analysis

A survey by Lo divides the workload management process into three sections: workload characterization, workload forecasting, and workload control [43]. On workload forecasting, the author refers to three standard routines used by the data processing industry: short-range forecasting,

medium-range forecasting, and long-range forecasting. The longer the forecasting range is, the less accurate the forecasting data becomes. The author also states that the most common approach for data forecasting is the time series technique, in which the time series is modeled and the forecasts are extracted based on this model [44].

In [45], the author outlines eight different techniques to time series forecasting: Simple Exponential Smoothing, Holt-Winters, Decomposition, Box-Jenkins, Bayesian forecasting, Multiple Regression, Econometrics, and Multivariate Box-Jenkins. Modern time series forecasting techniques are more numerous. However, they actually are different varieties and combinations of the above basic techniques. In essence, the forecasting procedure can be described in the following steps and basic techniques:

- Decomposition: Most, if not all, of the forecasting procedures start with this step. A typical time series is consider to have at least two components: trend and randomness. A seasonal time series will also have a seasonal component. A cyclical component is defined as short-term patterns with unknown periodicity and can appear in both non-seasonal and seasonal time series.
- Application of Forecasting Techniques: Depending on the characteristics of the acquired components, different forecasting techniques can be applied. While [45] mentioned eight techniques, these can be viewed as different flavors of the following: Box-Jenkins (ARMA/ARIMA), exponential smoothing, Bayesian forecasting, and regression. The forecasting process might be the application of one of the above techniques or a composition of several techniques on different components.

Decomposition

Traditionally, time-series decomposition aims to decompose the time series into three components: trend, seasonal, and random error. While the existence of a cyclical component is acknowledged, it is often grouped together with the random error term or the trend [46, 45]. The relationship between these components can be either additive, multiplicative, or both. The models for additive

and multiplcative decompostion are as followed:

$$Additive : X_i = T_i + S_i + E_i, i = 1, ..., n,$$

$$Multiplicative : X_t = T_i * S_i * E_i, i = 1, ..., n,$$

In these models, X_i is the original time series, T_i is the non-periodic trend, S_i is the seasonal component with known period, and E_i is the error term which belongs to a normal distribution with mean 0. Based on the known period, the seasonal component can be extracted using some form of moving average and seasonal indices.

Box-Jenkins: ARMA and ARIMA

In their work [47], Box and Jenkins separate time series into two types: stationary and non-stationary. A stationary time-series is defined as one whose values oscillate in equilibrium accross a "constant mean level". A nonstationary series, therefore, is one without a natural mean level. Stationary times series can be modeled using autoregressive models, moving average models, or a mixture of both (autoregressive moving average - ARMA). A generalized form of ARMA, called ARIMA (autoregressive integrated moving average) is shown to be an effective techique in modeling both stationary and nonstationary time series.

ARMA

Assuming the time series $(z_t, z_{t-1}, z_{t-2}, ...)$ has a mean μ, and deviations of the time series' values from the mean are denoted $(\tilde{z}_t, \tilde{z}_{t-1}, \tilde{z}_{t-2}, ...)$.

Assuming a series of white noise $(a_t, a_{t-1}, a_{t-2}, ...)$whose values are randomly drawn from a fixed normal distribution with zero mean and variance ${\sigma^2}_a$. These values are called the "shock values" [47]

An autoregressive process of order p is a time series model where the current deviation at time t is

calculated as a linear aggregation of the previous *p* deviations before t and a shock value:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + ... + \phi_p \tilde{z}_{t-p} + a_t$$

A moving average process of order q is a time series model where the current deviation at time t is calculated from a linear combination of previous *q* shock values:

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-p}$$

The ARMA model represents the deviation from the mean of the time series as a mix of both the autoregressive model and the moving average model [47]:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + ... + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-q}$$

The unknown values $\mu; \phi_1, ..., \phi_p; \theta_1, ..., \theta_q$ can be estimated from the time series data.

ARIMA

According to the basic definition of nonstationary time series, it would be impossible to capture the nonstationarity of a time series using fixed models. However, Box and Jenkins assume that there exists some kind of homogeneous behavior inside industry or business data. Particularly, while the direct differences between successive values are nonstationary, the difference of these differences or the difference of even higher level of difference can exhibit stationarity and thus, can be modeled [47]. The "assumed" stationary *d*'th difference of the series is defined as followed:

$$w_t = \bigtriangledown^d z_t = \sum_{m=0}^{d} (-1)^m \binom{d}{m} z_{t-m}$$

This *d*'th difference, in turn, can be modeled with a standard ARMA model:

$$w_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + ... + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-q}$$

In short, Box and Jenkins describe ARIMA as a three-step process to model a nonstationary time series from white noises randomly drawn from a normal distribution of mean zero and variance σ^2:

- Step 1: The white noise is passed through a moving average filter.
- Step 2: The output of step 1 is passed through a stationary autoregressive filter.
- Step 3: The output of step 2 is passed through a nonstationary Summation/Integration (the I in ARIMA) and the final result provides the estimated values of the time series

Forecasting with ARIMA

With the ARIMA model, a future value *h* time steps away from the current time *t* of a nonseasonal nonstationary time series can be forecasted. Because the generation of the forecasted value z_{t+h} from z_t involves the white noise series a_t, the confidence level of this forecast depends on the

variance ${\sigma_a}^2$ of the white noises' distribution and the number of steps into the future *h* [47]:

$$V(h) = \{1 + \sum_{j=1}^{h-1} {\phi_j}^2\} {\sigma_a}^2$$

Exponential Smoothing

Exponential smoothing was first published by R. G. Brown [48].Since then, exponential smoothing has become a successful base model for many statistical techniques. Hyndman identifies four basic varieties of exponential smoothing models in [46]: simple exponential smoothing, Holt's linear method, damped trend method, and Holt-Winters' trend and seasonality method.

Simple exponential smoothing

Brown noticed that the sum of squares of the differences between the most recent N observations and the estimate of the coefficient in a constant model with random noise can be minimized using the moving average of these N observations. To avoid the problem of storing the past N observations, Brown defined the smoothing function with the smoothing constant α as [48]

$$S_t(x) = \alpha x_t + (1 - \alpha) S_{t-1}(x)$$

x_t represents the value of the series at time *t*, and $S_t(x)$ represents the smoothed value of x_t, which could also be understood as the predictive value at *t+1*. Brown further develops this model into a smoothing function for polynomial models.

Holt's Linear Method

For the case of linear time series, Brown's model only works for the basic $y = ax + b$ model. Holt extended Brown's method to allow the forecasting of linear data with trends: [46].

$$Level : l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$Growth : b_t = \beta(l_t - l_{t-1}) + (1 - \beta)(b_{t-1})$$

$$Forecast : \hat{y}_{t+h|t} = l_t + b_t h$$

In these equations,

- l represents the trend
- b represents the change or the slope of the trend
- $\hat{y}_{t+h|t}$ is the forecast value at distance *h* into the future from the current time *t*
- α and β are the smoothing coefficients.

Damped Trend Method

Gardner and McKenzie generalize the Holt's linear model with a parameter ϕ called the autoregressive-damping parameter [49]. With this parameter, the forecasting process is automated and the forecasting accuracy is improved.

$$Level : l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1})$$

$$Growth : b_t = \beta(l_t - l_{t-1}) + (1 - \beta)\phi(b_{t-1})$$

$$Forecast : \hat{y}_{t+h|t} = l_t + b_t h$$

If ϕ equals zero, the equation becomes the simple exponential smoothing, and if ϕ equals 1, it is the Holt's linear model. ϕ should stay between 0 and 1. In the case that ϕ is greater than 1, the trend increases exponentially.

Holt-Winters' Trend and Seasonality Method

This method, proposed by Holt and later empirically proven by Winters [50], forecasts seasonal time series using a linear nonstationary trend. There are two different models, additive and multiplicative. These models are summarized as followed [46]:

- Additive Model

$$Level : l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$Growth : b_t = \beta(l_t - l_{t-1}) + (1 - \beta)(b_{t-1})$$

$$Seasonal : s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)(s_{t-m})$$

$$Forecast : \hat{y}_{t+h|t} = l_t + b_t h + s_{t-m+h_m{}^+}$$

- Multiplicative Model

$$Level : l_t = \alpha(/fracy_t s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$Growth : b_t = \beta(l_t - l_{t-1}) + (1 - \beta)(b_{t-1})$$

$$Seasonal : s_t = \gamma(\frac{y_t}{l_{t-1} + b_{t-1}}) + (1 - \gamma)(s_{t-m})$$

$$Forecast : \hat{y}_{t+h|t} = l_t + b_t h + s_{t-m+h_m{}^+}$$

In these equations,

- l represents the trend
- b represents the change or the slope of the trend
- s is the seasonal components of the series with known period *m*
- $\hat{y}_{t+h|t}$ is the forecast value at distance *h* into the future from the current time *t*
- α, β, and γ are the smoothing coefficients. These coefficients are to be estimated from the data, but it is recommended that they be restricted to be between 0 and 1 [46].
- $h_m{}^+ = [(h - 1)modem] + 1$

State Space Approach for Exponential Smoothing In [51], the authors have gathered all the different varieties of exponential smoothing techniques described above into a single equation model called the state space model:

$$Y_t = h(x_{t-1}) + k(x_{t-1})\epsilon_t$$

$$x_t = f(x_{t-1}) + g(x_{t-1})\epsilon_t$$

Y_t is the observation at time *t*, and x_t is the set of trend, growth (slope), and seasonal component

with known period m: $(l_t, b_t, s_t, s_{t-1}, ...s_{t-(m-1)})$. The notation of this model is call (E,T,S), which stand for error, trend, and seasonality. The possible options for each parameter of this model are No (N), Additive (A), or Multiplicative (M). The Additive and Multiplicative can also be dampened, denoted by A_d and M_d. The total possible combinations of state space models is 30. The introduction of this framework has led to an overall structure that subsumes all the exponential smoothing models and allowed for the creation of an automatic forecasting strategy [51].

Bayesian Forecasting

Bayesian forecasting is a quantitative technique. It is called Bayesian forecasting because at the heart of this method is the Bayes Theorem:

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

The basis of Bayesian statistics can be interpreted as followed [52]:

- Suppose there are *k* unknown values $\theta = (\theta_1, \theta_2, ...\theta_k)$ with a **priori** belief about these values $p(\theta)$.
- Suppose that we have *n* data observations $X = (X_1, X_2, ..., X_n)$ that relate to θ with a probability distribution function $p(X|\theta)$
- The Bayes Theorem can then be written as $p(\theta|X)$ is proportional to the product of $p(\theta)$ and $p(X|\theta)$. This is due to the fact that the probability distribution function $p(X)$ is independent from θ and likely to be known. Thus $p(X)$ can be considered a "normalizing" constant.

A formal introduction of Bayesian forecasting is presented in [53]. In this work, the authors suggest that the parameters of a statistical model (linear, Holt-Winter, ARIMA,...) may not be stationary and that these parameters can be estimated by previously known probability distribution models. As a result, the statistical models are then modified to incorporate the probabilistic information in order to help make the forecast results more dynamic in nature [53].

Regression-based Forecasting

Regression-based forecasting relies on the relationship between the known observations of the value of interest with a set of other known values [54]. A mathematical relationship will then be established in order to estimate the known observations of the interested values. These observations will not be guaranteed to be 100% correct. Rather, they will stay within a confidence interval range, typically expected to be at least 95% confident. The mathematical process for regression-based techniques requires the users to first determine an appropriate functional model. Depending on the structure of the data, the function can be linear or nonlinear. The regression model is then placed in the format:

$$Y = f(X) + \epsilon$$

with *y* is the value (or the set of values) to be estimated, *X* represents the set of know observations, f is the regression function whose coefficients are to be estimated based on the known values of *y* and *X*, and ϵ is the error term that belongs to a normal distribution with mean 0 and variance σ^2. A regression process that only estimates a single value is called univariate regression. With two or more values being estimated, the process is called multivariate regression. For time series forecasting, Y and X are actually different snapshots in time of the same time series. If there exists cyclical or seasonal components, the application of regression-based techniques has to be done with care to prevent the endogeneity effect, a correlation between X and the error term, that can affect the accuracy of the regression coefficients.

2.3.2 Neural Networks

While Lo's survey covers many approaches to workload forecasting using traditional time series analysis techniques, the subsequent years have seen the use of artificial neural networks for enhancing the accuracy of time series forecasting. The remaining of this section presents the application of neural networks to workload forecasting. With the introduction of the concepts by McCulloch and Pitts [55] and the subsequent key research contributions from many others such as Kohonen

[56], McClelland [57], Rumelhart [58], Grossberg [59], and Fukushima [60], neural networks have become an important tool in the field of data analysis. In their research, Lowe and Webb point out the difficulties with the two traditional forecasting approaches: model-based and statistical-based [61]. For the first approach, the lack of a complete understanding of the events as well as insufficient data collections usually leads to inaccurate models. On the other hand, the second approach is problematic due to the usual irregular, chaotic, and complicated real world events. The authors perform a forecasting test using neural network on a variety of time series and received more accurate results from the networks than the standard linear models. However, Lowe and Webb state that this difference could be due to the *a prior* knowledge of the original data generator. The authors conclude that in the event of insufficient information about the data, the neural network approach might not be much better than a linear model approach.

Rao, Sethuraman, and Ramamurti compare the time series prediction capability of a recurrent neural network against conventional nonlinear prediction schemes. In this research, there is no *a priori* information about the time series data model [62]. The authors state that while the network's predictions are close in term of accuracy with those of a standard Kalman predictor, it is still advisable to use conventional prediction techniques whenever possible due to the computationally intensive characteristics of the network. The authors also suggest the feed-forward neural network as a possible addition in order to achieve a higher performance than a Kalman predictor.

Connor, Martin, and Atlas propose a learning algorithm for a recurrent neural network in [63]. This algorithm is based on the estimation of filter data with the outliers removed. In this process, the authors showed that recurrent networks are a special case of nonlinear autoregressive moving average models (NARMA) and feed forward networks are a special case of nonlinear autoregressive models (NAR). As a result, they conclude that time series with moving average components are suitable for the application of recurrent neural networks.

Drossu and Obradovic's research confirm the suitability of neural networks in time series prediction based on two categories [64]:

- Neural networks can approximate not only any arbitrary continuous function on a compact

domain but also unknown functions by learning from examples.

- There exists a direct relationship between the basic stochastic models for time series and the neural network models.

However, the authors' approach to the prediction of time series is not limited to the usage of neural networks alone. Instead, they hypothesize that initial analysis using traditional linear models can yield valuable information about the data to help reduce the learning time and increase the accuracy of the neural network models. From their experiment on the Mackey-Glass data, the authors also conclude that the neural networks performed better than most of the stochastic model approximation due to the nonlinearity of the time series.

The work of Toda and Usui explores the possibility of combining neural networks with traditional nonlinear autoregressive models in order to estimate the parameters of the higher order spectra of time series [65]. The authors prove that their NNAR model has a stationary joint probability density function of Markov Chain. As a result, with the successful training of a NNAR model on a nonlinear time series, the authors derive the stationary joint probability density function of this time series.

Matsumoto, Hamagishi, and Chonan investigate the prediction of nonlinear time series using neural nets and hierarchical Bayes approach [66]. The authors recognize the difficulty of using conventional methods on nonlinear time series, especially those without a known functional form. With the utilization of neural nets, the authors achieve a high success rate in predicting chaotic time series.

The wide varieties of neural network types (e.g. feed forward, recurrent, self-organizing), choices of the activation functions, numbers of layers, and number of perceptrons have given neural networks an exceptional flexibility in adapting to different time series prediction problems. For example, the predictive capability of neural networks can be improved by changing the activation functions of the networks [67, 68, 69, 70, 71]. The layout structure of the neural networks can also be modified to improve the performance [72, 73]. Multiple neural networks can also be incorporated in order to achieve better predictions [74].

2.3.3 Signal Decomposition

Not all time series carry seasonal components with known period. Even if the seasonal period is known, too many cyclic patterns can still affect the accuracy of the forecast. An alternative to traditional time-series decomposition is to apply techniques based on signal decomposition methods. With signal decomposition, the most ideal assumption is that the decomposition process can separate different periodic patterns inside the time series. While this is often not the case, it is reasonable to assume that the decomposition technique can combine or split periodic patterns into components that exhibit consistent periodicity of themselves. Consequently, the forecasting techniques applied on these different patterns can offer a more accurate prediction.

Soltani, Canu, and Boichu use the recorded sunspots data to test the effect of combining wavelet-decomposition and neural networks [75]. Reasoning that the sunspots data is a Markovian type series with infinite possibilities for a minimal empirical risk, the authors decompose the data into a set of smaller components with their frequency spectrum localized. Theoretically, a forecasting technique applied to these components will yield better results than when applied to the original data. The final prediction results of this work validate the efficiency of this combined technique over traditional forecasting techniques.

A more detailed comparison of the application of signal decomposition techniques to time series data preprocessing is done by Shin and Han in [76] on the changes over time of the ratio between the Korean won and the US dollar. In this work, two different decomposition techniques, Fourier and Wavelet, are compared against each other in the role of preprocessing data for a neural network. They are also compared against a tradition time series analysis technique: ARMA (Autoregressive Moving Average) as well as a prediction using neural network (NN) alone. The results of this work showed that:

- While ARMA and NN are comparable in performance, both are outperformed by the NN using the preprocessed data.
- Between the two decomposition techniques, Fourier Transform outperforms Wavelet Trans-

form. However, this result is not conclusive, since there exist many other wavelet transformation techniques that are not tested in this work.

Based on the above results, the authors conclude that preprocessing data will yield better results than using just the original data.

Another combination of signal decomposition and forecasting techniques is done by Yu, Goldenberg, and Bi in [77] on the classic sunspot data. However, instead of using a neural network, the authors decompose the data into stationary signals to feed into a linear statistical model. The model of choice is the ARMA model. For the decomposition technique, the authors rely on previous work to state that wavelet transform is better than Fourier transform. Besides using the decomposed signals as the input for the primary prediction mechanism, the authors also use the predicted values with the original data as a boundary to limit the depth of the decomposition process. While not offering a numerical evaluation, the authors presented compelling visual results to make the case for the effectiveness of the effectiveness of their methods: the use of data decomposition as well as the imposed boundary using original data.

Instead of using the most recent section of time series, Xuefeng and De rely on a set of different time series segments from the past in order to predict the future [78]. These segments are decomposed into multiple sets of wavelet components. Based on fractal theory, the components from different sets that are closest to each other are chosen to be predicted. The final results are summed up from these predicted components from different time series segments. While the different percentages of errors remain uncertain, this technique does have potential in predicting workload data that has a significant level of noise.

Work by Mao on the workload data of network arrivals also confirms the effectiveness of using the wavelet transform to preprocess data [79]. Similar to the above work, Mao applied the Haar wavelet transform to the workload data before feeding the data through a forecasting mechanism, the ARIMA model. Unlike the above work, Mao not only applies the wavelet decomposition on the original workload but also on different time scale of the workload. Given that the workload is the histogram of network arrivals; it is easy to create different histograms on different time scales.

The final prediction is based on the combination of the decomposed components of different time scales. A final comparison on the multiscale decomposition technique and a traditional neural network shows that the multiscale decomposition technique generates better results on all cases.

2.3.4 Empirical Mode Decomposition

Complex processes often exhibit sporadic behavior resulting from multiple causes, with each of these causes occurring at different time intervals. The frequencies of the causes may also vary with time. These types of signals are readily identified in an Empirical Mode Decomposition analysis, but can remain hidden in Fourier or wavelet analysis. Introduced by Huang [2], Empirical Mode Decomposition (EMD) is an adaptive technique that represents complex data as the sum of a small number of orthogonal empirical modes, called Intrinsic Mode Functions (IMF). Each mode has a symmetric envelope defined by local maximums and minimums such that its mean is zero everywhere but does not require linearity or time invariance in the underlying signal. In comparison, Fourier decomposition presents time series as a sum of harmonics with fixed frequencies and amplitudes. This usually works well for stationary processes, when the harmonic frequencies are present at all times. When representing non-periodic or non-stationary data sets that have discontinuities or sharp peaks Fourier decomposition may become computationally inefficient or offer poor signal representation. Under these conditions, wavelet decomposition is often used, though it also has limitations such as the limited length of the basis function and the difficulty in defining local changes.

EMD is similar to wavelet decomposition in its applicability to non-stationary data. Like Fourier analysis, wavelet decomposition employs a predetermined basis [80]. However, the EMD basis changes in time to adapt to actual data variability. By using information from the signal, instead of prescribing basis functions with fixed frequency or imposing a particular set of basic functions, EMD results in a relatively small set of IMFs.

An IMF is a function that satisfies two conditions. The first condition requires that the number of extrema and the number of zero crossings in the complete data set must either equal or differ

by at most one. The second condition specifies that at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero. The process to find an IMF is called the sifting process and has the following steps [2]:

- Step 1: Given a data set $X(t)$, identify all the local maxima and minima
- Step 2: Interpolate all the local maxima into an upper envelope $U(t)$ and all the local minima together into a lower envelope $L(t)$ using a cubic-spline technique.
- Step 3: Find the mean of the upper and lower envelopes.

$$m(t) = U(t) - L(t)$$

The first prospective component $H(t)$ is the difference between the original data and the mean. If $H(t)$ does not satisfy the conditions of an IMF, repeat step 1 with $H(t)$ until the resulting component is an IMF. In reality, it is unlikely for the $H(t)$ to satisfy the second condition of the IMF's definition. In this case, when the sifting process falls into an infinite loop, it is necessary to manually stop the process. Huang proposes a stopping condition by limiting the standard deviation between two consecutive residue [2]:

$$SD = \sum_{j=0}^{T} [\frac{|h_{1(k-1)}(t) - h_{1k}(t)|^2}{h^2{}_{1(k-1)}(t)}]$$

The typical range for SD is between 0.2 and 0.3.

- Step 4: Subtract the IMF component of step 1 from the data set to find the residue. Repeat step 1 on the residue to find the next IMF component. When the residue becomes a monotonic function or it is no longer feasible or efficient to decompose the data further, the process is ended. In practice, immediately when the reside contains one optimal point, the sifting process is stopped due to the inability to generate the proper upper and lower envelopes for the next iteration. In addition, in some cases, after a certain number of IMF's extraction, the residues become repetitive in form (containing only two or three optimal points) and are no

longer meaningful. At this point, visual observation is necessary to determine when to end the sifting process.

An early study of the EMD process was done by Sharpley and Vatchev. In this study, the authors prove that a solution of a self-adjoint second order ordinary differential equation will satisfy the first condition of IMF. If this solution also satisfies the second condition, it is then considered an IMF component of the original signal [81]. In [82], Kizhner et. al. publish further theoretical developments about the Hilber-Huang Transform. While still recognizing the lack of explanations for the behavior of EMD algorithm, the authors attempted to fill in some of these gaps. By reproducing the procedure of EMD's sifting technique with more details and performing a comparison with other known decomposition techniques, the authors formulate and prove three hypotheses about the characteristics of EMD.

- Hypothesis 1: The order of IMFs being sifted out of the main time series is based on each IMF's average frequency, begins by the highest and ends with the lowest (before the residue).
- Hypothesis 2: The intermediate locally symmetric zero-crossing pair of extrema points with interleaved regions of diminishing amplitudes are preserved by the EMD sifting process, thus follow the first condition of IMF's definition.
- Hypothesis 3: The convergence of the EMD sifting process using piecewise cubic spline is of order $O(\frac{1}{2^{k-1}})$, with k is the maximum allowable number of EMD sifts for each IMF. This hypothesis explains that the convergence of the EMD sifting process is actually fast. The process of proving this hypothesis also offers more insights about the piecewise cubic spline's significance, the preservance of the symmetric pairs of adjacent extremas, as described in Hypothesis 2.

These hypotheses form the foundation for several data partition techniques that help speed-up the computation time of the EMD sifting process [82].

While not contributing toward the main theoretical foundation of EMD, other research focuses on improving the EMD sifting procedure, mainly dealing with how to interpolate the upper and

lower envelope and how to mitigate the end effect. Different alternative analytic approaches for the mean envelope formulation of the sifting process have been developed. Rather than trying to find the upper and lower envelopes, Delechelle et. al. calculate the mean directly from the extremas using a technique based on the partial difference equation work by Sharpley [83, 81]. The usage of the B-spline instead of the classic cubic-spline was proposed in order to help identify a more explicit formula for the envelopes [84]. A different approach to the construction of the mean envelope using rational splines is presented by [85]. In this method, the interpolation process is done between segments of consecutive extrema instead of a group of three extrema like the cubic-spline. The fitting functions for the different segments are of the same form but with different coefficients. The requirement for these functions is that the adjacent functions need to have matching values as well as matching first and second derivatives at the connecting points. Kopsinis and McLaughlin propose the usage of genetic algorithms in order to identify the optimal interpolation points and demonstrate improvements over the cubic-spline in the simulation examples [86].

To handle the end-point effect of the EMD's sifting process, a number of methods have been proposed that deal with how the interpolation points at the ends of the time series are chosen. Coughlin and Tung propose to extend the original time series at both ends with additional data based on the structure of the time series [87]. Rilling et. al. duplicate the extrema closest to the end points for the interpolation process [88]. The SZero method by Peel et. al. only performs the interpolation process for the optimal points between the end points, and the two end splines are projected toward the end-points [89]. Zhidong and Yang compare the end-points with the average of all the maxima(minima) and choose the greater(smaller) value to participate in the interpolation effect [90]. The effectiveness of all of these methods are compared empirically without any theoretical proof.

Even with the lack of theoretical foundations, the empirical evidence for the usefulness of EMD is abundant, with extensive applications across a variety of disciplines. In health sciences, the EMD process is utilized to perform analysis on patients' biological information. This is due to the relationship between physical meaning of the biological data and the decomposed IMF components

[91, 92, 93, 94]. With recent advances in computing power, the application of EMD in this area has been carried out in real time [95]. In the field of image processing, EMD proves its capability in different areas of image detection and extraction [96, 97, 98] as well as fusion [99]. EMD is also used to characterize the arrival time data of an enterprise computing cluster [3]. The sinusoidal form of IMF components can be used for prediction purposes. Hamad combines EMD and an artificial neural network in order to predict the travel time on an express highway [100]. In a more recent work, Li and Wang forecast the short-term wind speed using a combination of EMD and ARMA (Autoregressive Moving Average Model) [101]. Similar approaches to forecasting, where EMD is combined with another technique for prediction purpose, are described in [102, 103].

2.4 Software and and Software Packages

In this research, the initial implementation of EMD was done with Java. The choice was not about preference, but rather to fit the characterization module for the Intergrated Capacity Planning Environment [40]. However, when more calculation was needed in the workload characterization process, Matlab became a more convenient choice for the dissertation. As more and more statistical calculation is needed, R has become a better choice of programming language. R is an open source project, developed since 1997 and geared to be a programming language for statistical and graphics development [104]. At a first glance, R seems to be similar to Matlab, with much less internal functions and without an easy-to-use GUI. However, being an open source project, R benefits from a large variety of contributed packages. A more detailed comparison between R and Matlab can be found at [105, 106]. However, one disadvantage of R is the lack of data cleaning capacity. Due to the fact that R was designed for statistical calculations, the preferred format for input files are either comma or tab delimited with clear separations between columns of data. As a result, the data files that also include metadata and heading information need to be prepared first with a Java program before being fed into the R program.

Throughout the research, several R packages that have been proven to be the correct implementation of the required techniques are used. The first R package is called *EMD*. It contains a

procedure to perform the EMD sifting process on a set of time series data. The implemented sifting process uses the classical cubic spline and stopping rule. To handle the end-effect conditions, the procedure offers four choices, one of which is similar to [88]. Another two choices are similar to [87], and they make the assumptions that the original series is either symmetric or periodic for the extension purpose. The final choice for the end-effect condition extends the data near the end points twice, one is a direct mirror across the end-points, another is a mirror and negation process across the end-points. Each extension will produce an upper and a lower envelope. The average of all four envelopes is the desired sifting result. Aside from the implementation of the EMD sifting process, the EMD R package also offer a forecasting technique using vector auto-regression that allows the prediction of sum of the IMF components minus the trend. The second R package is called *forecast*. This package has been in continuous development since early 2008, and allows the automatic forecasting (automatic determination of forecast parameters) using the extensive varieties of the state-space exponential smoothing models as well as the classic ARIMA models. Another R package for forecasting purpose is *vars*, which implements the vector-autoregressive-based forecast. For the study of wavelet, we use the package *wmtsa*, which is the companion software package for [107]

The output of the workload characterization process is tested on the ICPE simulator [40]. This is a discrete time/event simulation model based on the JavaSim simulation package [108]. There are two simulation modes: trace driven and distribution driven. In the trace driven mode, the simulation runs on a list of jobs from a trace file. This file can either be a real workload trace or a synthetic workload trace. In the distribution driven mode, the simulation uses an exponential or hyper-exponential distribution to generate synthetic jobs to drive the simulation process.

2.5 Acxiom Data

The specific parallel batch system that is modeled in this research is an enterprise grid system that belongs to Acxiom Corporation. It consists of a batch scheduler, a number of server systems providing common services, and a pool of nodes that are dynamically allocated to individual pro-

cessing jobs. In this enterprise grid, jobs are submitted to the scheduler, where they wait in a queue until selected by the scheduler for execution. Once all required resources have been allocated, a job executes on its assigned processing nodes mutually exclusively from other jobs until completion. When a job completes its processing, its assigned nodes are returned to the available resource pool for use by subsequent jobs. Workload and performance data (e.g., node utilizations, job response times, system throughput) are recorded in log files, which are collected monthly. The size of the acquired log files spans from January of 2006 to March of 2007, consisting of thirteen trace files for each month. The trace files collect the statistics of each job arrival to the system, including arrival time, wait time, run time, number of CPU requested, amount of RAM requested. All the log files have been cleaned and converted to the standard workload format [40]. This standard workload format is an extension of the standard workload format by [109] with the addition of a "Records Processed" field. Figures 2.1 and 2.2 illustrate the arrival counts of the entire Acxiom data set for different timescales.

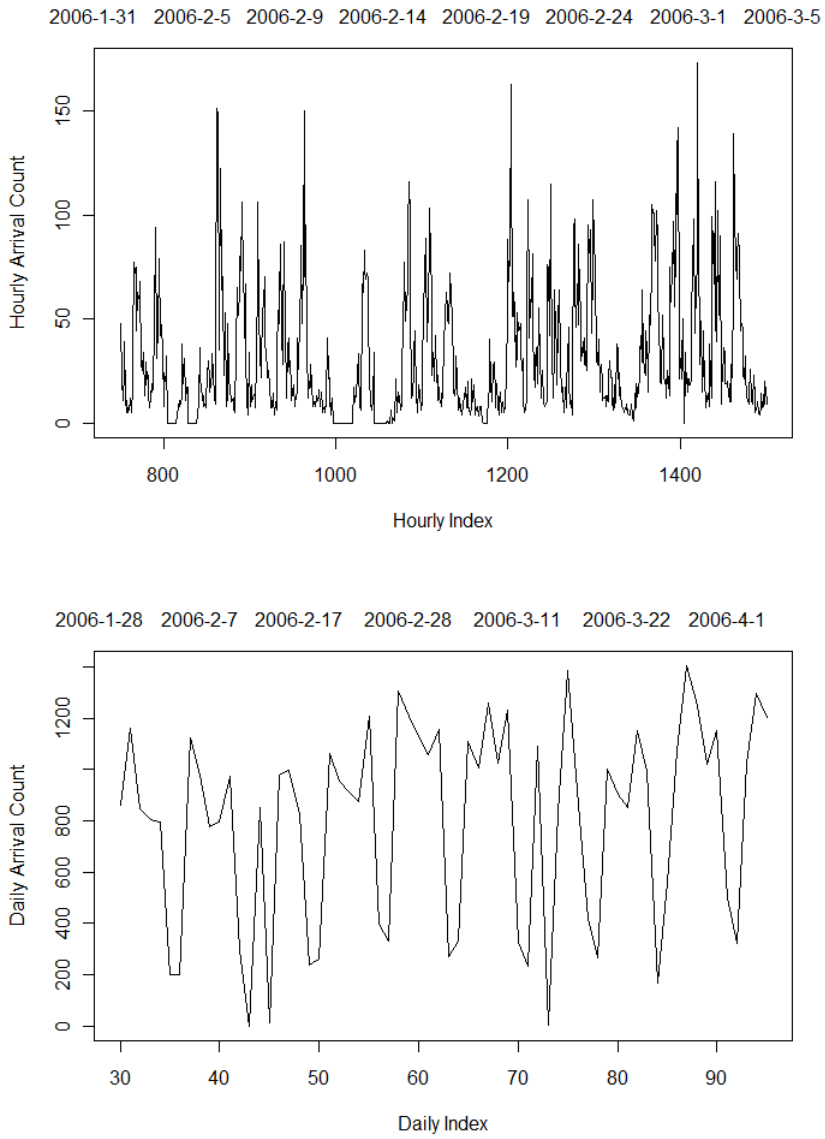

Figure 2.1: *Acxiom Arrival Counts at Hourly and Daily Scale*

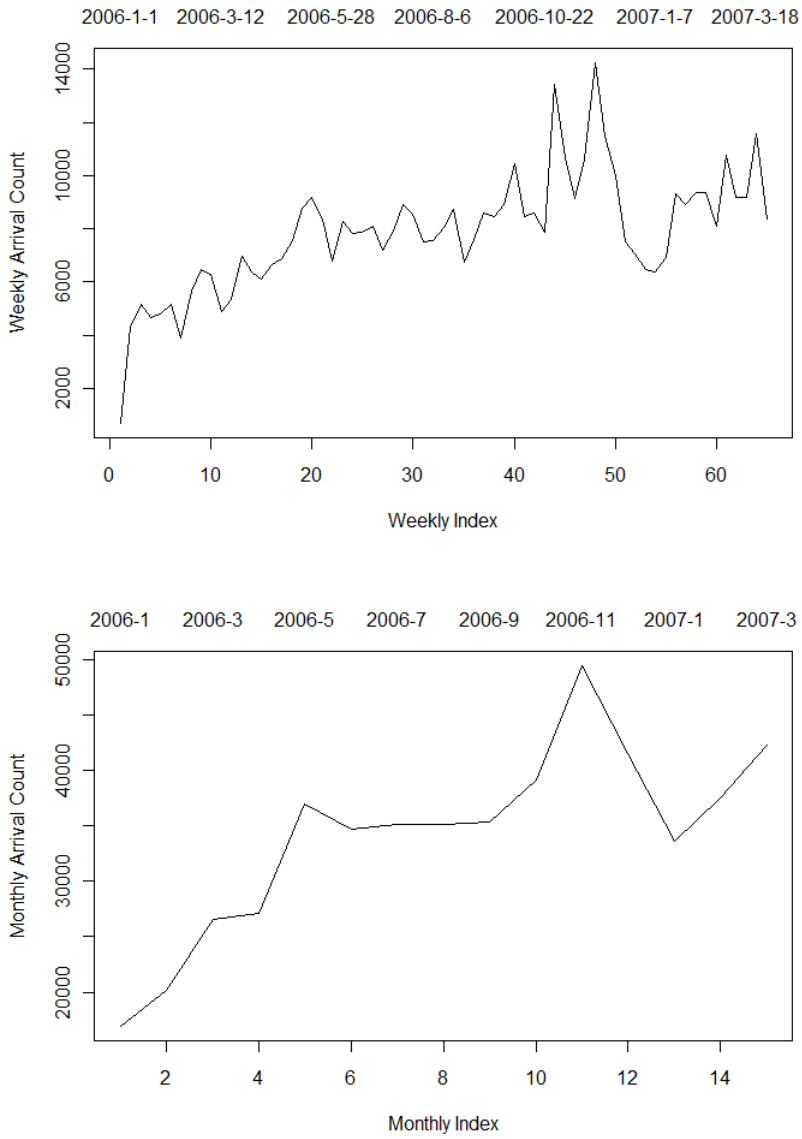

Figure 2.2: *Acxiom Arrival Counts at Weekly and Monthly Scale*

Chapter 3

Workload Characterization

In this chapter, we investigate the use of Empirical Mode Decomposition as a workload characterization tool. The results from this chapter are published as [3].

3.1 Introduction

Performance analysis of a cluster of computer systems usually starts with the collection of log files containing node utilizations, job response times, system throughput, and workload information. However, the volume of information recorded for large clusters over any significant time span eventually becomes so large that analysis directly from the log files is cumbersome. In order to reduce the amount of data analyzed, a workload model can be created based on the statistical characteristics of the log file. Based on such a model, synthetic workload data can be generated that closely resembles the original log file data. A workload model can also be modified to generate synthetic log files for different hypothetical workloads in order to answer interesting "what if" questions for various capacity planning studies [110]. Although a workload model can generate synthetic data that faithfully reproduces the overall statistical measurements of the original log file data, the generation of individual entries in the log file often does not accurately capture the different patterns of randomness of the real data. The arrival time behavior of jobs is particularly hard to recreate. This is due to the fact that the synthetic data are generated by a single statistical distribution while the real data can exhibit many different, complex, and interdependent patterns.

From the literature survey in Chapter 2, it is typical in the characterization process to filter the non-stationary arrival stream into a stationary stream in order to fit a distribution. In order to find an alternative approach that could maintain the non-stationary characteristic, thus the realistic nature of the synthetic workload, as well as to improve the accuracy of the synthetic data set, this chapter investigates the usage of Empirical Mode Decomposition in the characterization process.

Section 3.2 describes the initial application of traditional approach to the characterization of the sample workload. Section 3.3 shows how the EMD method is applied to modeling job arrival times. Section 3.4 analyzes the advantages and disadvantages between EMD and the hierarchical hyper-exponential technique. Section 3.5 summarizes and concludes the chapter.

3.2 Workload Characterization using Traditional Approach

In this section, following the steps to construct a synthetic workload by [9] and the additional statistical steps by [12], a characterization procedure based on the traditional approach is proposed. The procedure has the following steps:

- Step 1: Analysis of the full Acxiom data to identify the common data range, and analysis of the data parameters stored in the workload file to identify the range and parameters to be characterized
- Step 2: Statistical analysis of the chosen data
- Step 3: Application of a statistical distribution to generate the synthetic workload.

3.2.1 Description of the Workload Data

Acxiom's data is recorded on a monthly basis using a separate data file for each month, and it is reasonable to use a month as the range of data to be characterized. As previously mentioned in Chapter 2, the Acxiom data set contains thirteen months of data from a production cluster. The raw log file has been converted to the Standard Workload Format, which includes nineteen parameters. While the majority of the parameters are not used, the following parameters contain data:

- Submit Time: The instant in time when the job is submitted to the queue. The submit time is rounded to multiple of sixty seconds. The submit time is reset to 0 at the beginning of each month.
- Wait Time: The waiting time in the queue for the job.
- Run Time: The execution time in the system.

- Number of Processors: Number of processors that the job requests.
- Average CPU Time used: this is measured as the execution time in the system.
- Group ID: The resource type, which is the sub-clusters that the jobs are assigned.
- Execution Number: Job ID as indicated by Acxiom.
- Records Processed: The number of data records accessed by the job.

Since the purpose of workload characterization is to help with system capacity planning, the parameters that depend on system configuration such as *Wait Time*, *Run Time*, and *Averaged CPU Time Used* are not used in the characterization process. Putting the identification parameters *Group ID* and *Execution Number* aside, that leaves only the *Submit Time*, *Number of Processors*, and *Records Processed* to be characterized.

3.2.2 Workload Characterization using Hierarchical Characterization

The workload characterization case study is on the full Acxiom data set. In the data set, it is observed that for the *Number of Processors* field, there are 14 unique values: 1, 2, 3, 4, 5, 6, 8, 10, 11, 12, 15, 16, 20, and 24. A similar analysis on the number of records requested per job shows the *Records Processed* field to have 5125 unique values ranging between 0 and 966,982,867. For convenience, the histogram analysis of the *Records Processed* field has the record count divided into eight buckets as illustrated in Figure 3.1. This analysis shows that the majority of jobs require less than 100 data records. A similar histogram analysis for the *Number of Processors* yields a somewhat similar situation where the majority of jobs use between one and four CPU nodes, as illustrated in Figure 3.2. Figure 3.3 describes the relationship between the *Number of Processors* and *Records Processed* of jobs within a month. Clearly, with significantly less unique values to choose from, the *Number of Processors* field is an ideal target to be characterized. In fact, the characterization of this field is simply categorizing the jobs according to the number of processors requested. This is the first level of characterization in the hierarchical characterization approach.

With the jobs categorized into different groups, each of these groups is then fitted to an appropriate hyper-exponential distribution. The details of this process are described in [110]. The

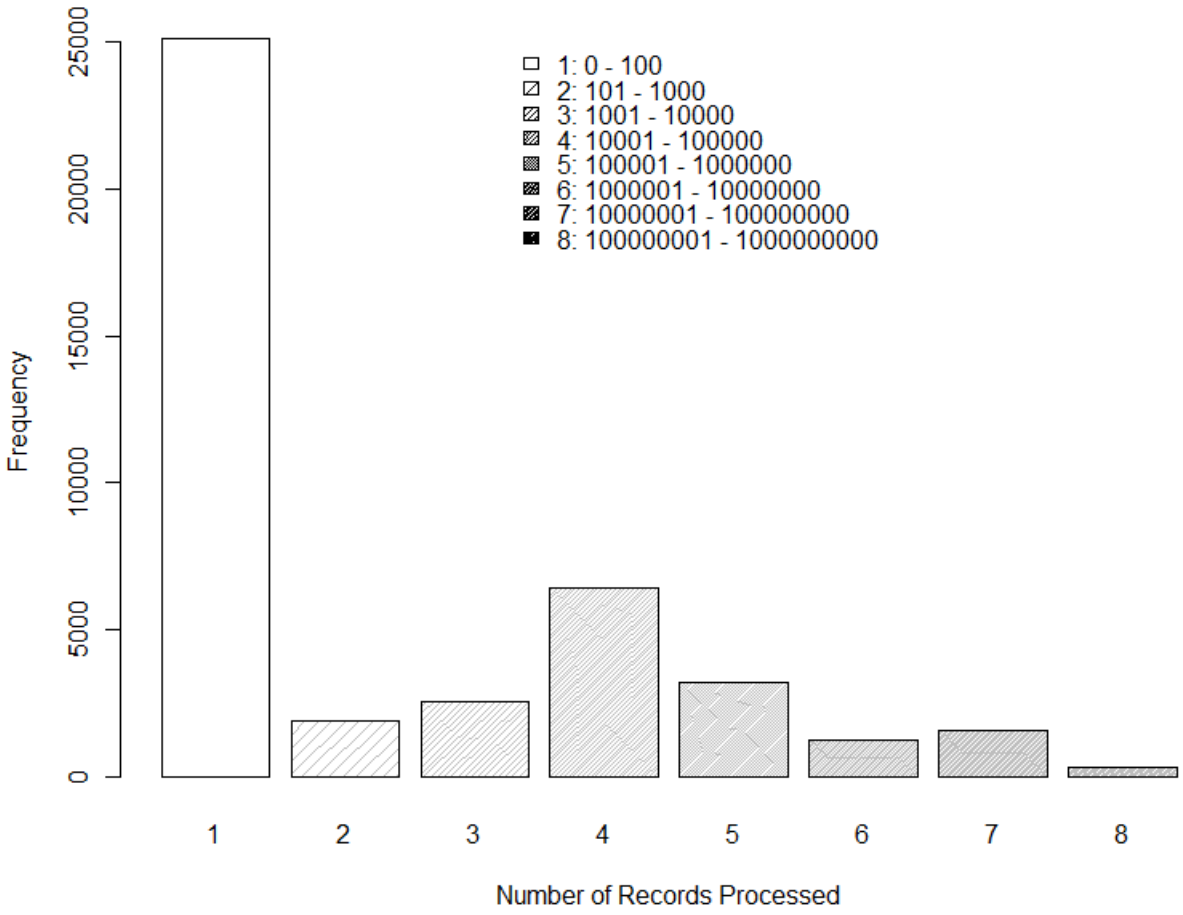

Figure 3.1: *A histogram of the number of records requested per job during Mar 2007 of Acxiom data*

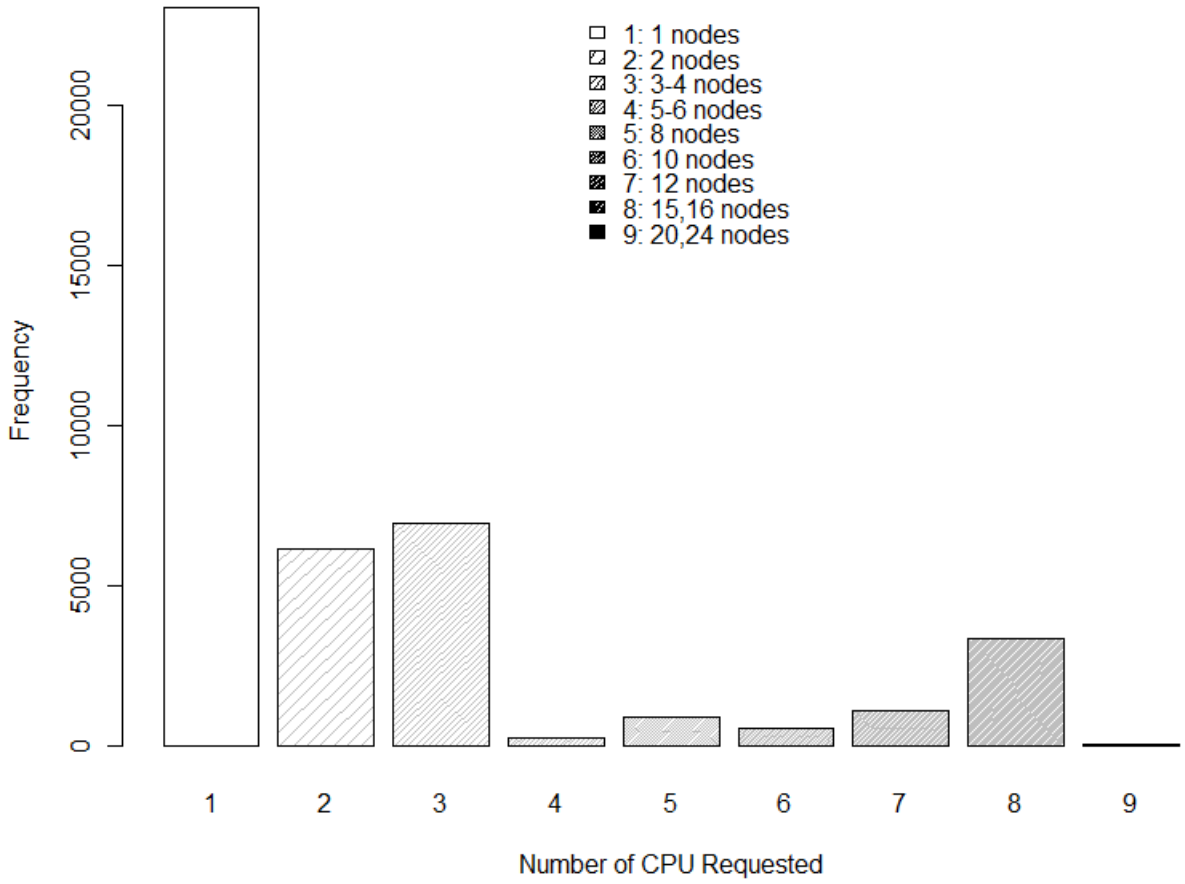

Figure 3.2: *A histogram of the number of records requested per job during Mar 2007 of Acxiom data*

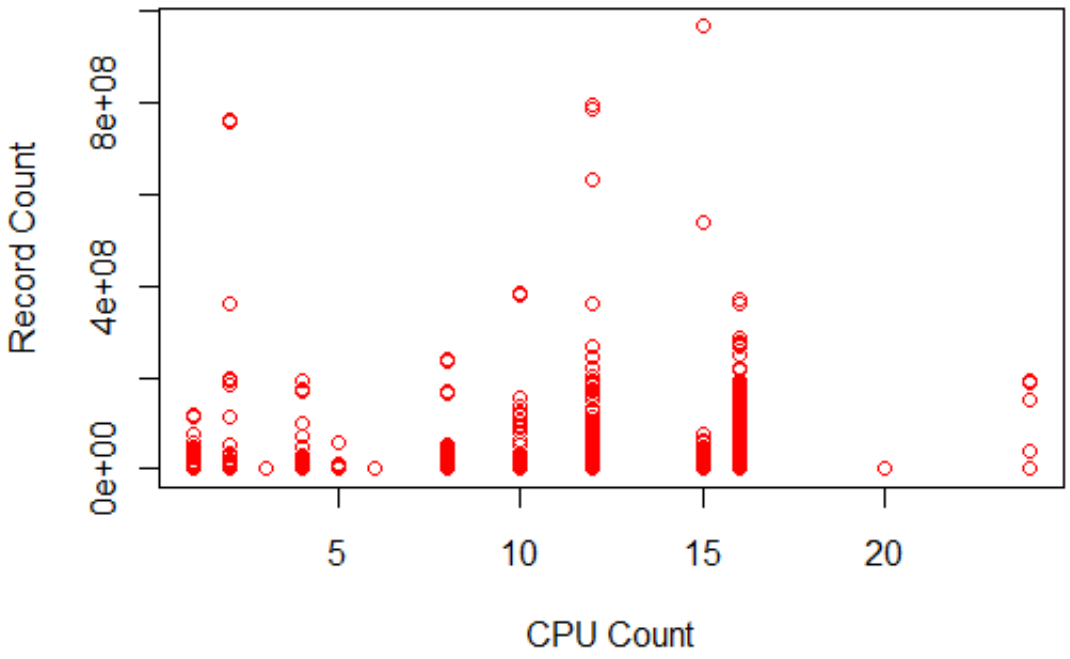

Figure 3.3: *Graph of CPU count (x-axis) versus Record count (y-axis) for each job during Mar 2007 of Acxiom data*

resulting distribution parameters are used as the characterization of the workload.

3.3 Workload Characterization using Empirical Mode Decomposition

In this section, we describe an alternative approach to the problem of workload characterization based on the application of the Empirical Mode Decomposition (EMD). Unlike the traditional approach, our method analyzes the time plot of the job arrival stream directly using the sifting process of EMD and separates resulting Intrinsic Mode Functions (IMF) from the stream. These IMFs are then mapped to a set of sinusoidal functions to form the characterization of the original stream. The experimental log file is the file of March 2007. This month was chosen because not only its overall arrival structure is similar to all the other months but this month also has an interesting time period where no job arrives to the system at all, which represents an abnormality that could potentially affect the characterization results.

3.3.1 Arrival Time Plot

The arrival time of each individual job is recorded in the log file at the exact time of arrival. In order to convert the arrival stream into a time series format to be usable by the EMD technique, a set of arrival buckets is created, where the arriving jobs are placed into buckets of one-hour length to create an arrival time plot.

Figure 3.4 shows the hourly arrival time plot of March 2006. The original log file contains 42,353 individual jobs that are distributed into 743 hourly buckets (31 days). The y axis is the total number of arrivals during a single time-bucket period. The x-axis is the index of the hours, start at the 1st hour at the beginning of the month and stop at the 743th hour at the end of the month. Figure 3.4 shows a distinct period of lack of arrivals between the 241st and 274th hours. There is no such period in any other month.

Next, the EMD process is applied to the arrival time plot of the March-2007 data. The resulting IMF components are shown in of Figure 3.5. The IMF components are labeled from component 1 to component 8. In addition, there is a residue line. For this data set, 10 IMF components and the

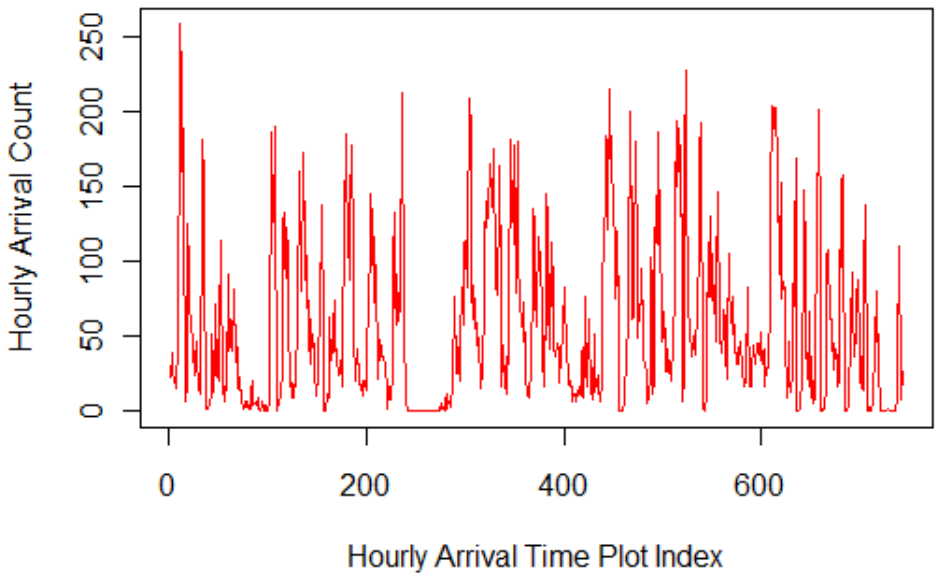

Figure 3.4: *A time plot for the arrival count per hour of Acxiom data during the month of March 2007*

trend line are generated. The components' graphs are in the form of oscillating signals. The first derived IMF component has the highest frequency, and subsequent components have decreasing frequencies, consistent with [82]. Without a theoretical proof to demonstrate the exact relationship between the IMFs and the original, we rely on visual observations. In Figure 3.6 each individual IMF is shown along with the original data. From Figure 3.6, it shows that the oscillations within the components follow with the different characteristics of the original histogram. In particular, the IMF components with highest frequencies (1, 2) reflect the hourly oscillations and the IMF components with lower frequencies (3,4,5,6) reflect the daily/half-day oscillations. IMF 7 captures the differences between the weekday/weekend schedule, while the IMF component with the lowest frequencies are relatively stable in term of amplitudes and follow a two-week period. It should be noted that the lack of job arrivals during the period between indexes 250 and 280 due to a system down time can be observed most clearly in IMFs 1 and 2. Given that this period lasted 33 hours, it should have also been seen more clearly on the medium-frequency IMFs (3, 4, 5, 6). However,

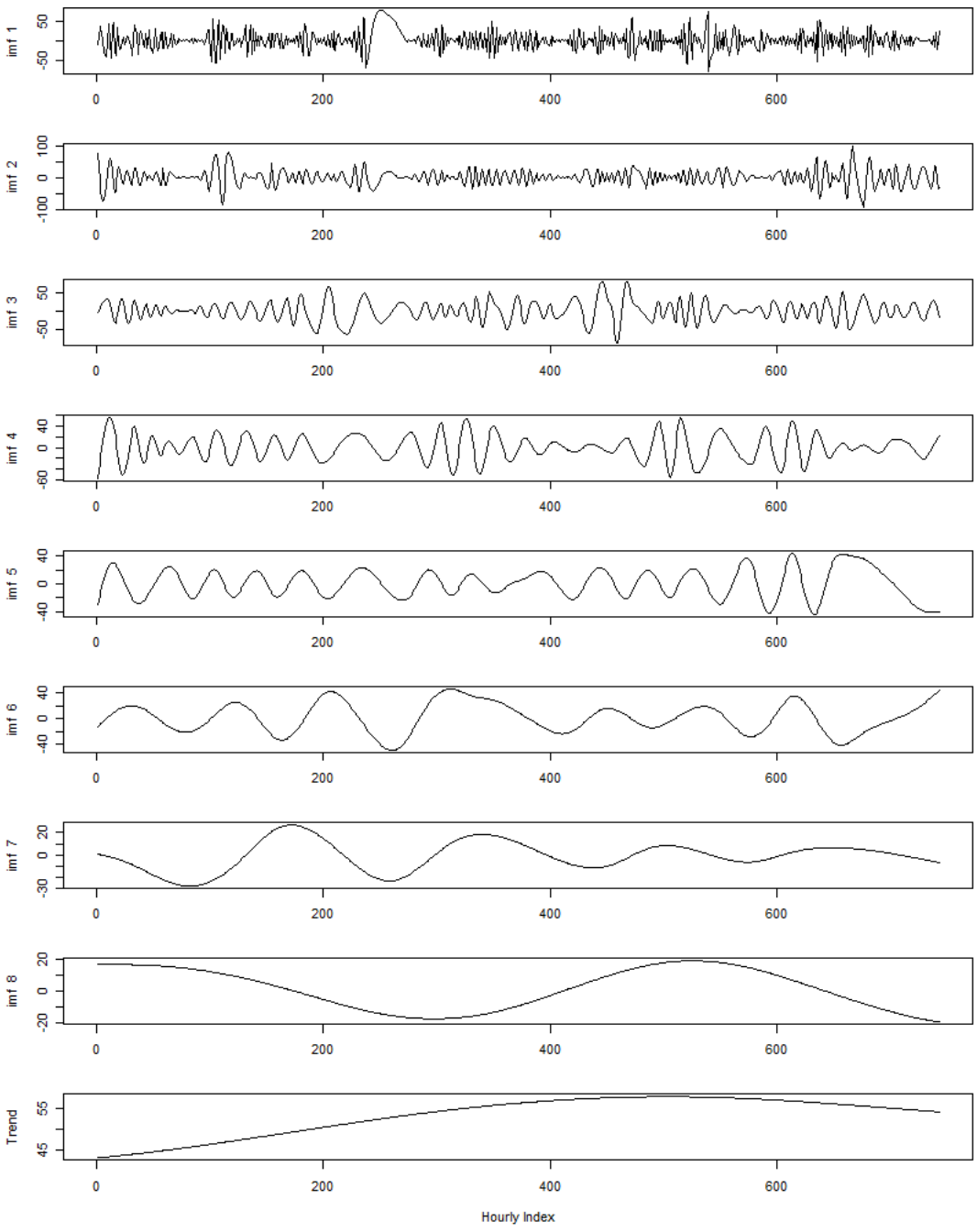

Figure 3.5: *Resulting IMF components of the application of the EMD sifting process on the March 2007 data*

since this event happened during a weekend, the effects of its are reflected more on the hourly-related IMF components rather than the daily-related components.

By definition, the IMFs are subtracted from the original data whenever they are identified. Therefore, the subsequent cumulative sum of these IMFs plus the residue will reproduce the original data. Obviously, as more IMFs are added, the partial sum will become closer to the original time plot. However, as each IMF does capture a different pattern from the original time plot, only a small percentage of IMF components are needed before a resemblance of the original time plot is recognized. In Figure 3.7, the cumulative sum of the IMFs are graphed, with the top graph contains only the residue and the lowest-frequency IMF, and more and more IMFs are added to the subsequent graphs. Typically, the more IMF components that are combined, the more closely the result resembles the original signal, much like adding subsequent terms of a Fourier series together to converge on the original wave form.

Table 3.1 compares the means and standard deviations between the original time plot data and the different combinations of IMF cumulation. From the table, it is shown that the means of the original time plot and the IMF combinations are very similar to each other, no matter how many IMFs there are in the combinations. On the other hand, the standard deviations are quite different for the combinations with a low number of IMFs. However, begin at the combination of five IMFs or more plus the residue, the difference between the combination's standard deviation and the original's standard deviation is 24%. This difference is lowered to 9.13%, 0.52%, and 0.00% as the number of IMFs is increased to six, seven, and eight IMFS (eight is the maximum IMFs). An exhaustive investigation shows that the other months of Acxiom's data also carry the same characteristics between the means and standard deviations of the original time plot data and the combinations of IMF cumulation.

3.3.2 IMF Component Curve Fitting

In the previous section, it is shown that the IMF components produced by the EMD process can be used to reconstruct the original time plot. In addition, only a part of the set of IMFs is needed

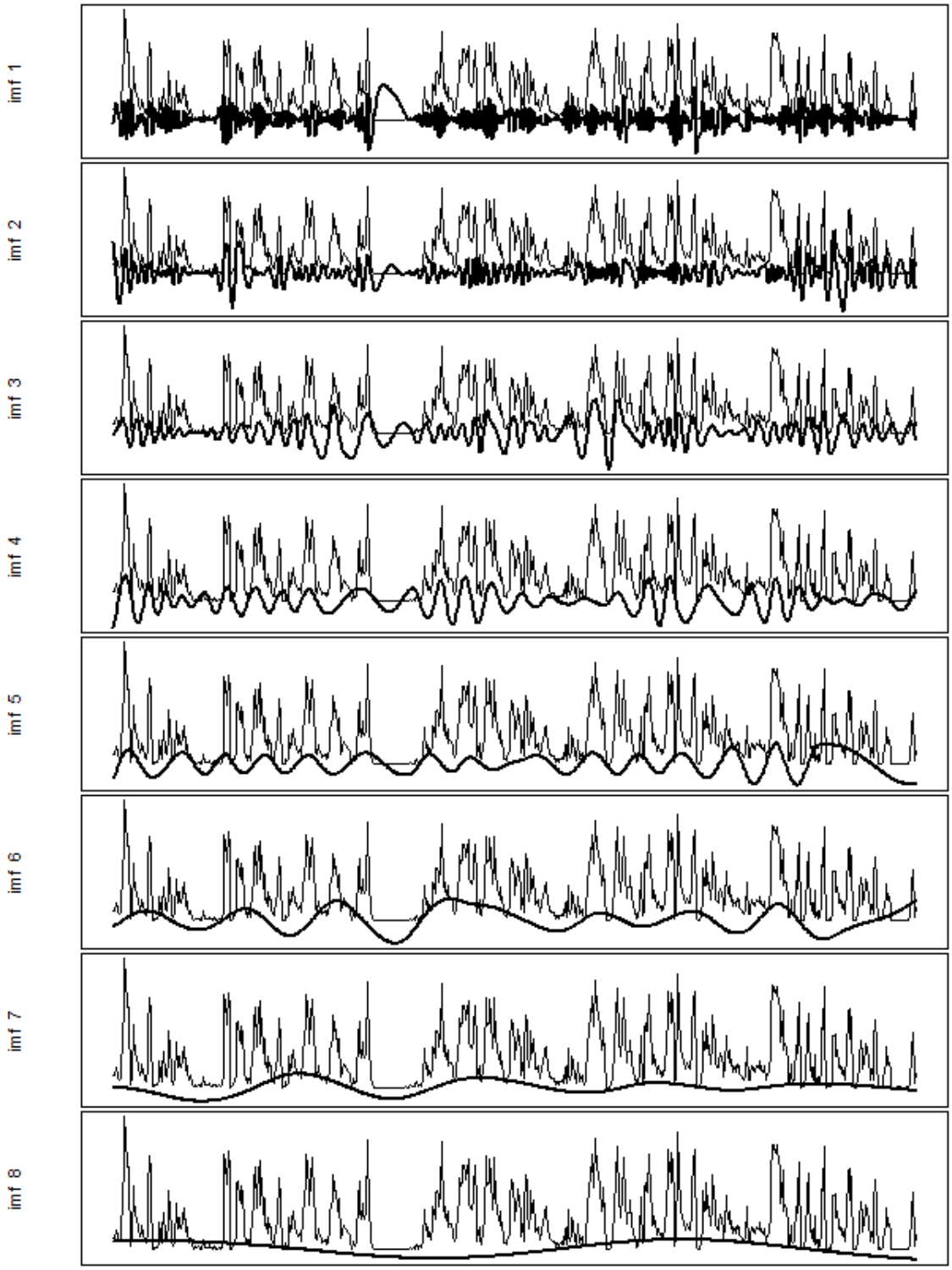

Figure 3.6: *Mapping IMFs on top of the original data*

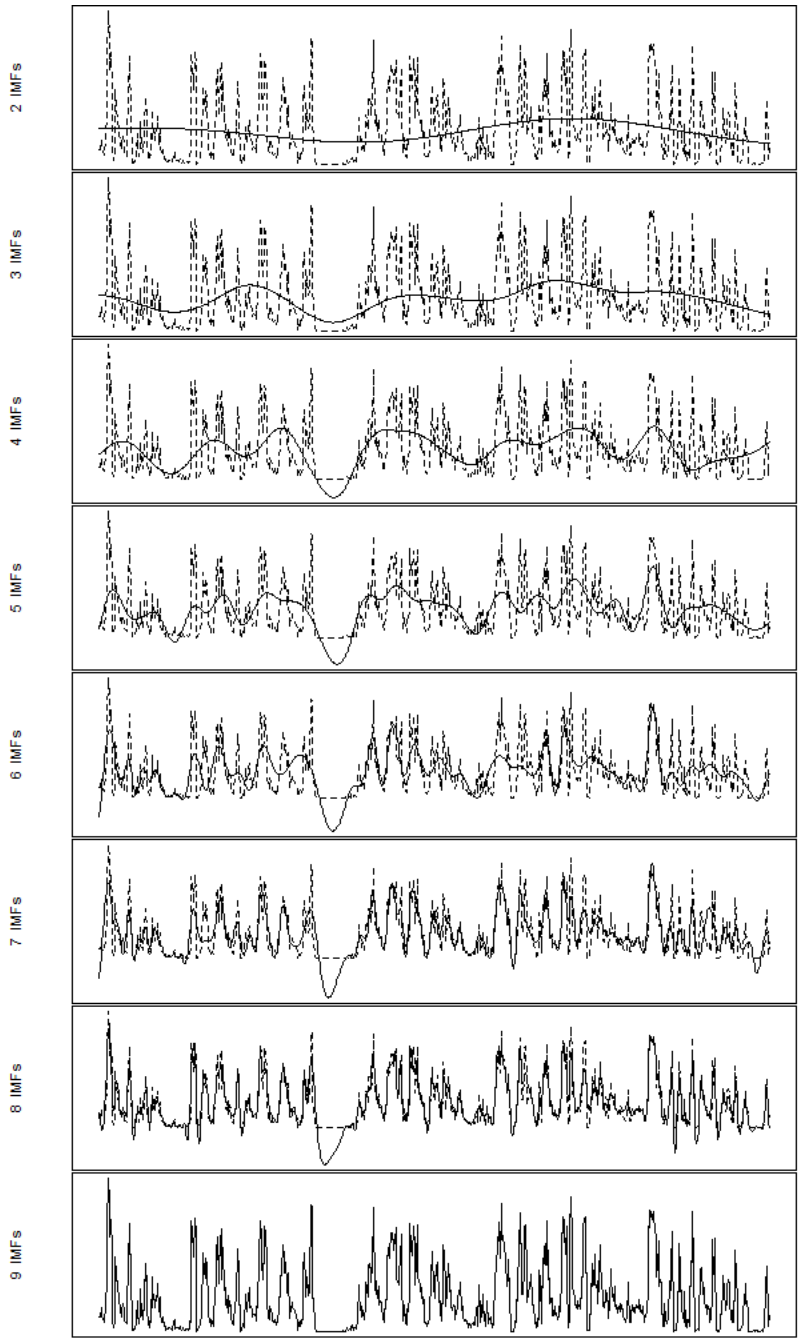

Figure 3.7: *Partial Sums of the IMF components, starting with the sum of the residue and the lowest-frequency component, with subsequent higher-frequency components added*

Data Set	Average	Standard Deviation
Original	57.00269	54.19422
1 IMF + Res	54.65462	12.69929
2 IMFs + Res	53.55117	17.17170
3 IMFs + Res	55.24514	29.82440
4 IMFs + Res	55.94378	34.60228
5 IMFs + Res	55.84559	40.99535
6 IMFs + Res	56.00857	49.24587
7 IMFs + Res	55.63788	54.47624
8 IMFs + Res	57.00269	54.19422

Table 3.1: *Statistical Measurement Comparison between IMF Combinations and The Original Time Plot*

in order to form a synthetic data stream that closely resembles the original data. However, since there is not a functional form for the arrival time plot, and thus, there are no easily recognizable functional forms for the IMF components. Consequently, the IMF components are stored as direct data points. Thus, the amount of storage needed is equivalent to the storage cost of the original time plot multiplied by the total number of IMFs plus one (for the residue). As a result, the direct usage of the IMF components as a modeling tool is not advantageous. A standard approach to this problem is the usage of Fourier Sine Series (FSS) to map each IMF component to a function of sine of the following form:

$$imf_i(x) = \{\sum_{j=1}^{N_i-1} b_j sin(\pi N_i x)\}$$

With N_i is the number of coefficients used in the fitting of IMF i to an FSS function. Figure 3.8 shows an example of the FSS application on the IMF components. This approach is convenient in term of procedure. However, there are two problems with this approach:

- For high-frequency IMFs, this procedure is computationally expensive.
- The FSS's performance suffers while trying to capture the ends of the IMFs.

The first problem can be resolved by ignoring the high-frequency IMFs. As indicated in Table

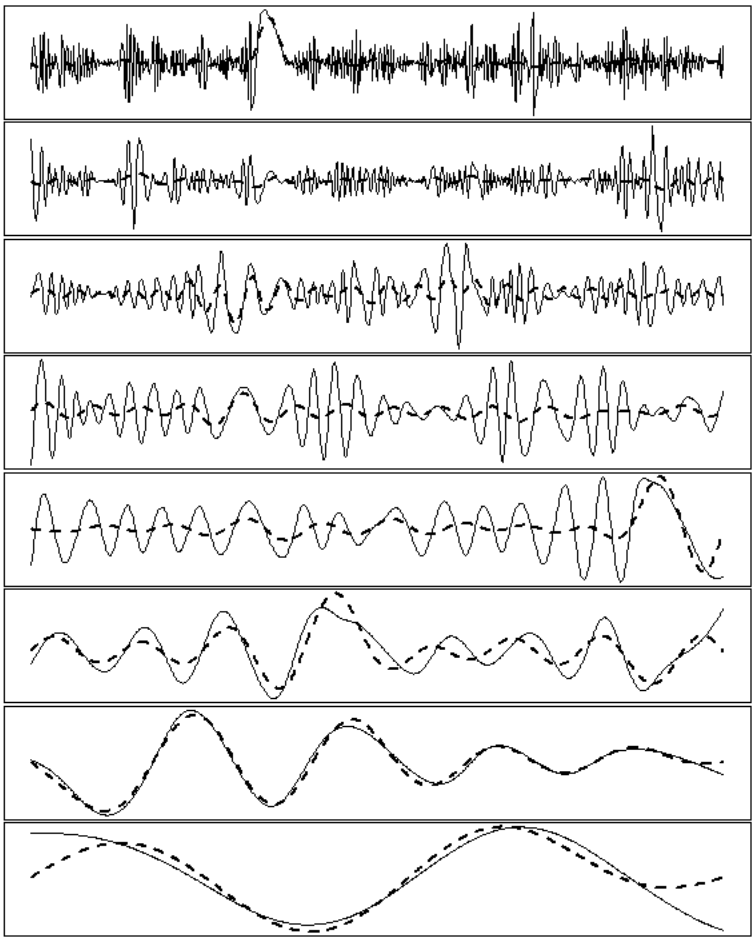

Figure 3.8: *Fourier Sine Series fitted upon the IMF components*

IMF	Fourier Sine Series Fit Time (seconds)	Piecewise Fit Time (seconds)
IMF 1	more than 7200	0.05
IMF 2	more than 7200	0.01
IMF 3	5696.25	0.01
IMF 4	121.04	0.01
IMF 5	150.5	0.02
IMF 6	17.66	0.02
IMF 7	5.33	0.01
IMF 8	5.44	0.01

Table 3.2: *Statistical Measurement Comparison between IMF Combinations and The Original Time Plot*

3.1, a combination of the residue and the five lowest IMF components has already come within 2% error for the mean and 13.6% error for the standard deviation. For the second problem, the issue of curve fitting for IMF components is approached from a piecewise perspective. Visual observation and experimental results show that a single "peak" going from one zero crossing to another can be fitted with a sine function. This sine function has the form:

$$y = Asin(\pi \frac{x - s}{s - e})$$

where A is the optimal amplitude of the peak, s is the index of the starting zero crossing, and e is the index of the ending zero crossing. Figure 3.9 demonstrates the results of this fitting process. Given that piecewise fitting is a relatively simple process, it is a better choice than the Fourier Sine Series fitting due to its linear runtime and its capability to alleviate the end-effect problem. The significant improvement in fitting time is demonstrated in Table 3.2, where the run times for each IMF from both approaches are compared. Since the nature of the piecewise fitting guarantees a highly accurate fit every time, it is only fair to limit the FSS fitting to a visually pleasing fit. It should be noted that the more accurate the FSS fittings are required to be, the longer it takes to run the FSS fitting algorithm.

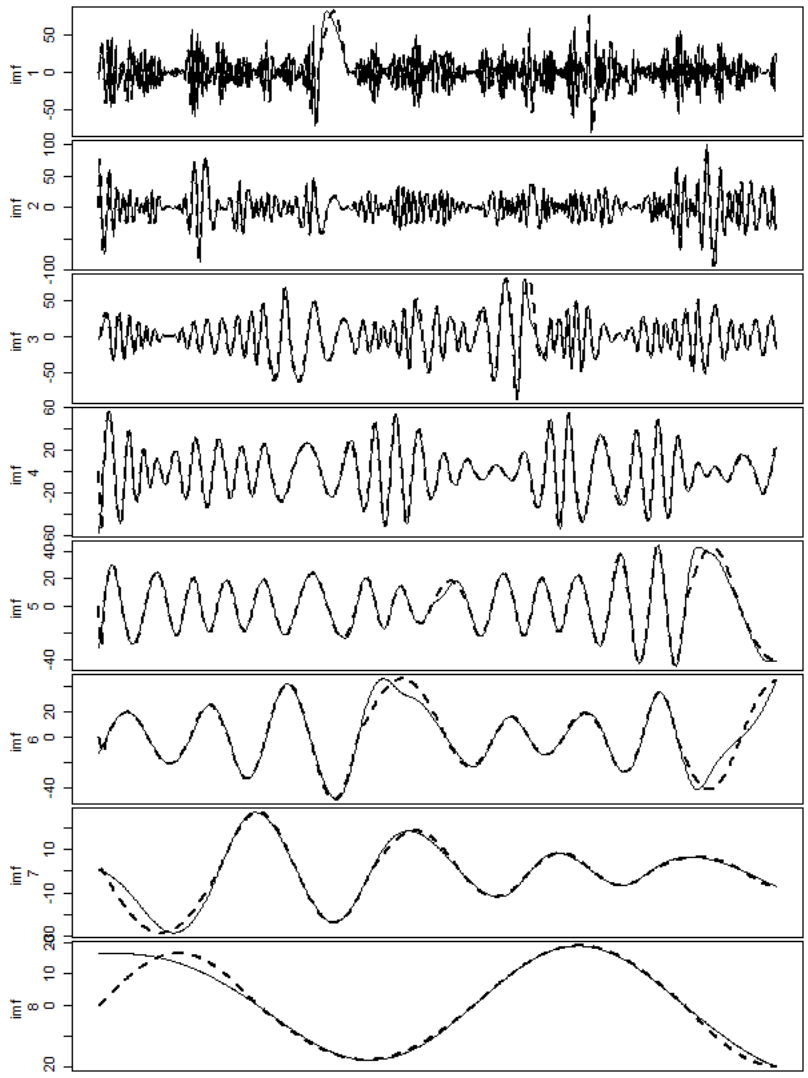

Figure 3.9: *Piecewise Fitting upon the IMF components*

3.3.3 Residue Fitting

The combination of using IMF components alone is not sufficient to accurately model the arrival data. The residue line is also needed. However, the residue line is usually a monotonic function or, as in the case of Figure 3.5, a relatively simple polynomial function. Thus, the polynomial fitting technique can be applied to the residue. Depending on the purpose of the characterization, the users can choose to not fit the residue but only use an average of the reside. This is the trade-off between simplicity and accuracy.

3.4 Analysis

In this section, we discuss the synthetic job arrivals generated from the EMD-based characterization and compare these with the results from the traditional statistical methods.

3.4.1 Overall Analysis

The first test of the overall analysis is to compare the job arrivals generated by different combinations of the IMF components and the residue with the original arrival data itself. The synthetic arrivals are created by a simple normal distribution random generator for each hour bucket. From the results in Table 3.1, we decide to start with the combination of at least five IMFs plus the residue, and we also choose not to include the IMF with highest frequency. Figure 3.10 shows the similarities between cumulative distribution functions of the original data, and the chosen combination of IMFs and residue. In the Figure, notations 5-, 6-, and 7-IMF represent the number of IMFs included in the combinations. Since the addition of the first IMF with highest frequency will return the original arrival stream, that combination is not examined. The highly matched cumulative distribution functions imply that the individual synthetic arrivals generated from the chosen combination will closely resemble the exact arrivals of the original data. This result also emphasizes the previous observation that high-frequency components can be omitted from the IMF combination.

3.4.2 Comparison to Hyper-Exponential Distribution (HED)

Previous work shows that the hyper-exponential distribution technique offers a good statistical match between observed and synthetic job arrivals on the first two statistical moments (mean and standard deviation) [40]. When the job arrivals are generated, the HED technique is also able to generally mimic the actual arrival times. Figure 3.11 compares the cumulative distribution functions of the original data, the IMF combinations, and the HED-based data. Similar to Figure 3.10, notations 5-, 6-, and 7-IMF represent the number of IMFs included in the combinations.

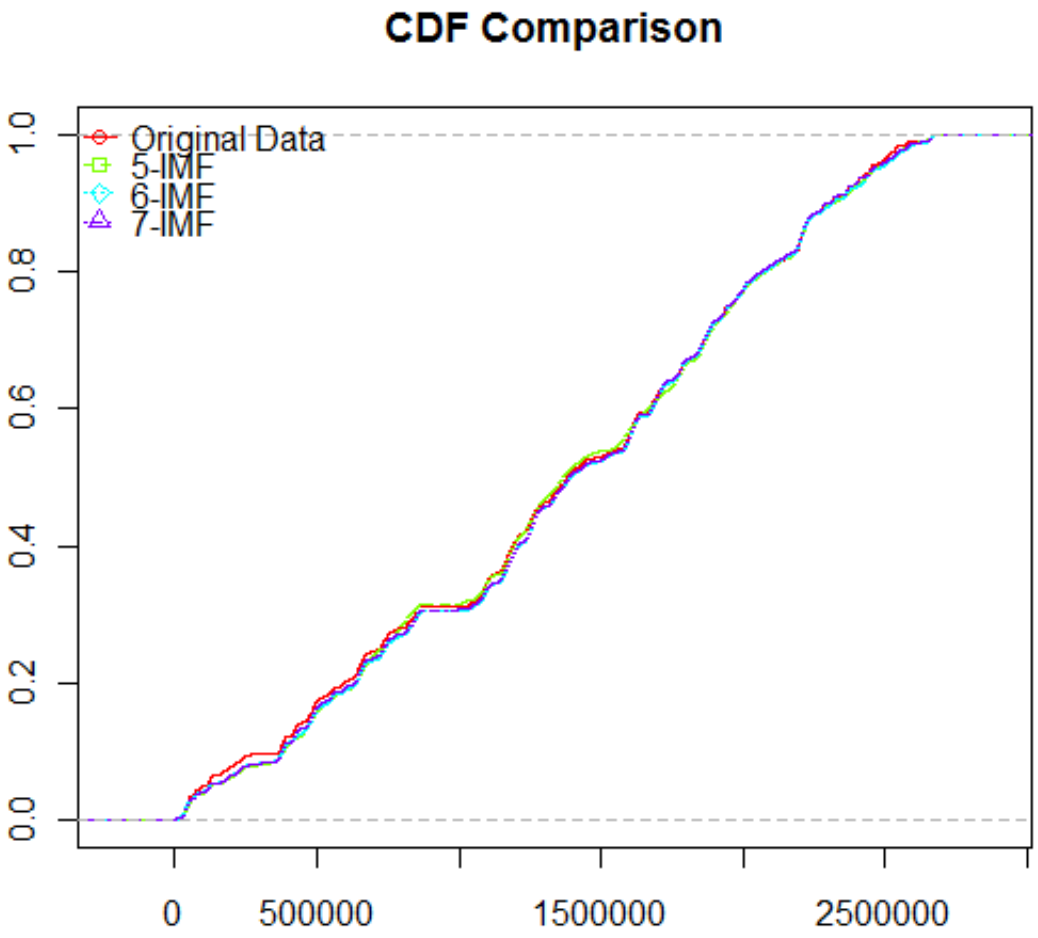

Figure 3.10: *Comparisons among Cumulative Distribution Functions of IMF-Combinations based Synthetics and the Original Arrival Stream. The y-axis is the cumulative distribution and the x-axis represents the job arrival index*

Clearly, Figure 3.11 shows that the traditional HED-based synthetic data does not perform as well as the EMD-based synthetic data for capturing the detail bursts of the original arrival stream.

3.5 Conclusion

In this chapter, the use of Empirical Mode Decomposition (EMD) as well as a traditional approach that relies on hierarchical characterization and hyper-exponential distribution (HED) as methods of arrival time characterization of workload arrival stream are demonstrated. The application of the sifting procedure of the EMD to the original arrival streams leads to the production of different IMF components, which carry patterns similar to those hidden within the original arrival stream. In order to characterize these components, the piecewise fitting technique is chosen over a Fourier Sine Series fitting, due to the high accuracy as well as a significant improvement in fitting time. The resulting synthetic job arrivals generated from a set of these fitted functions and the fitted residue follow a distribution that is highly similar to the original arrival stream's. In comparison to the synthetic job arrivals generated by the HED approach, the EMD-based arrivals capture the characteristics of the original arrival stream better. It should be noted that the EMD-based approach method requires a nontrivial amount of manual calibration, such as choosing the combination of IMFs with the best fit, in order to achieve optimal results. In addition, it is difficult and redundant to modify the resulting piecewise-fitted sinusoidal functions for the purpose of capacity planning, in comparison to the characterized ratios produced by the HED approach.

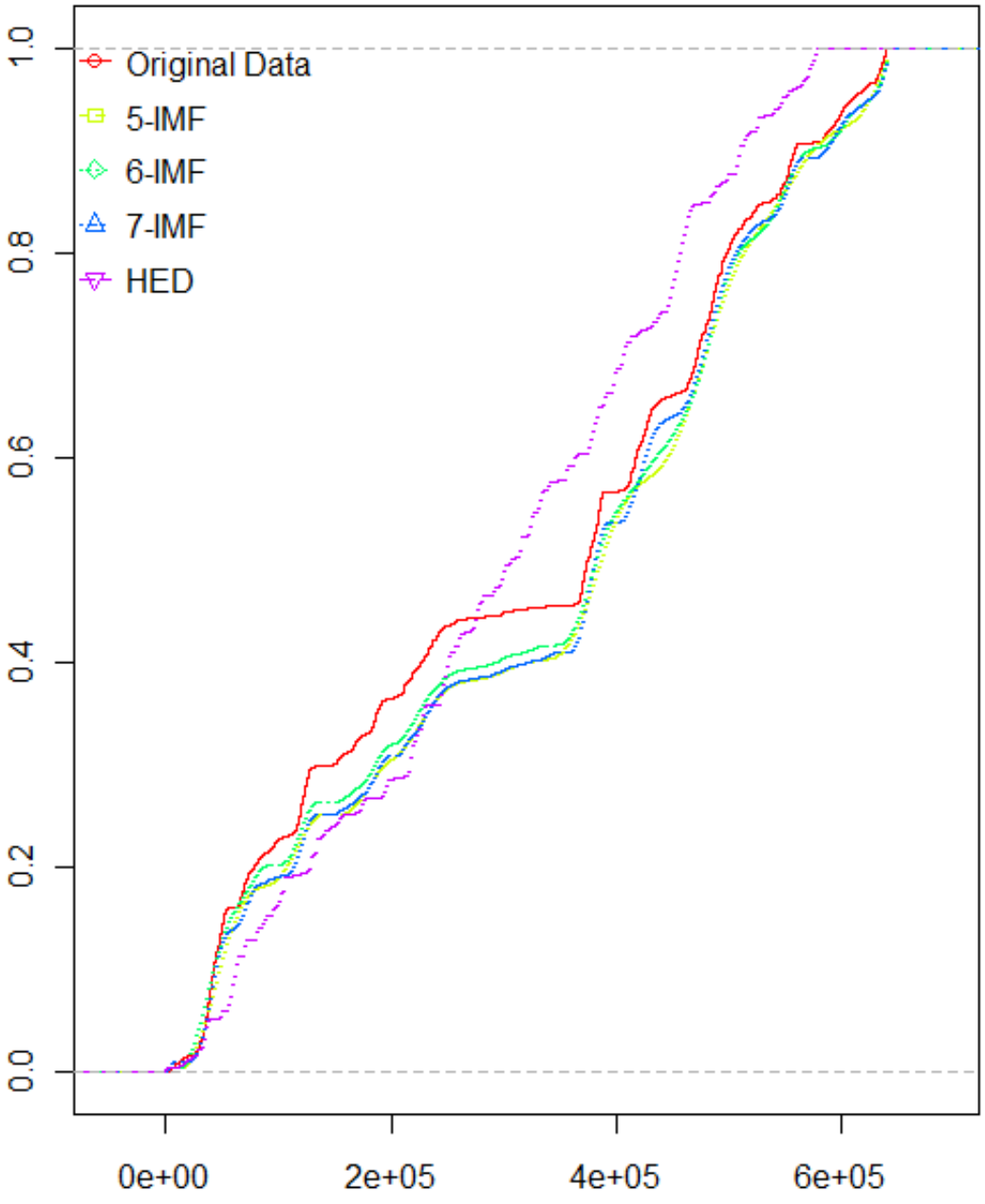

Figure 3.11: *Comparisons among Cummulative Distribution Functions of IMF-based Synthetics, HED-based Synthetic, and the Original Arrival Stream*

Chapter 4

Workload Forecasting: Baseline Study

In chapter 3, we discussed the application of EMD's sifting technique in the characterization of the arrival stream based on data from the Acxiom enterprise grid. While the EMD-based characterization offers better accuracy, the technique requires significant manual calibration in order to achieve optimal results. Consequently, it is more practical to utilize a traditional statistical distribution based technique in characterizing workloads. However, the strength of EMD-based approach lies with the ability to separate different frequency levels steams from an arrival time plot. Consequently, this opens up another possibility of application for EMD, which is to incorporate the IMF (Intrinsic Mode Functions) products of EMD into the workload forecasting process in addition to the characterization process. This chapter investigates this application of EMD. Results from this chapter are published as [4]. In particular, in section 4.1, the decomposition strengths of EMD are investigated empirically with examples. Next, section 4.2, a baseline study is performed to investigate the feasibility of this work. Finally, section 4.3 summarizes the findings of the baseline study.

4.1 Example of EMD's Decomposition Capability

In [2], EMD has been shown to successfully separate the sum of a sine component and a cosine component. Additionally, Chapter 3 also shows that the resulting IMF components (except for the trend line) of an EMD-decomposed time plot can be fitted piecewise sinusoidally. The foundation of this research is based on the hypothesis that EMD has the capability to extract and separate the different signals with different periodicities hiding within the arrival stream. As this hypothesis cannot be proven mathematically, it is necessary to empirically investigate this capability for sample functions. We look at three functions consisting of three sinusoidal components and a linear trend, but with frequencies of the sinusoidal components. The structure of this function represents

a simple data stream with three components, each having different periodicities, and one linear trend. The number of x values is 300. This functional structure is chosen because the workload analysis shows that the Acxiom data is composed of sinusoidal components and a linear trend. The three sets of coefficients/number of components is for the purpose of demonstrating the following questions:

- What is the capability of EMD in isolating components whose frequencies are close to each other?
- What is the capability of EMD in isolating a large number of components?

Figure 4.1 shows the graphs of these functions. The three functions are called $f_1(x)$, $f_2(x)$, and $f_3(x)$. The equations of these functions are:

$$f_1(x) = sin(x) + sin(2x) + sin(3x) + 0.5x$$

$$f_2(x) = sin(x) + sin(7x) + sin(20x) + 0.5x$$

$$f_3(x) = sin(x) + sin(4x) + sin(8x) + sin(12x) + sin(16x) + sin(24x) + sin(168x) + 0.5x$$

Figure 4.2, 4.3, and 4.4 show the sinusoidal components and the decomposed IMF for f_1, f_2, and f_3, respectively. EMD's sifting process successfully decomposes the functions into their respective set of sinusoidal components and trends. The decomposed IMFs and residues follow the original components and trends with a high degree of accuracy. Notice that the IMF is less accurate at the end of the curve due to the end-effects of the empirical process.

Further examinations indicate that when the number of sinusoid components are increased, it is necessary to also increase the number of observations and the interval containing the observations in order for the EMD's sifting process to successfully isolate the components. However, this also increases the influence of the end effects of the EMD's sifting process as well as the run time of the process. Therefore, manual calibration is needed at this step in order to ensure that the EMD can isolate the underlying components efficiently and accurately.

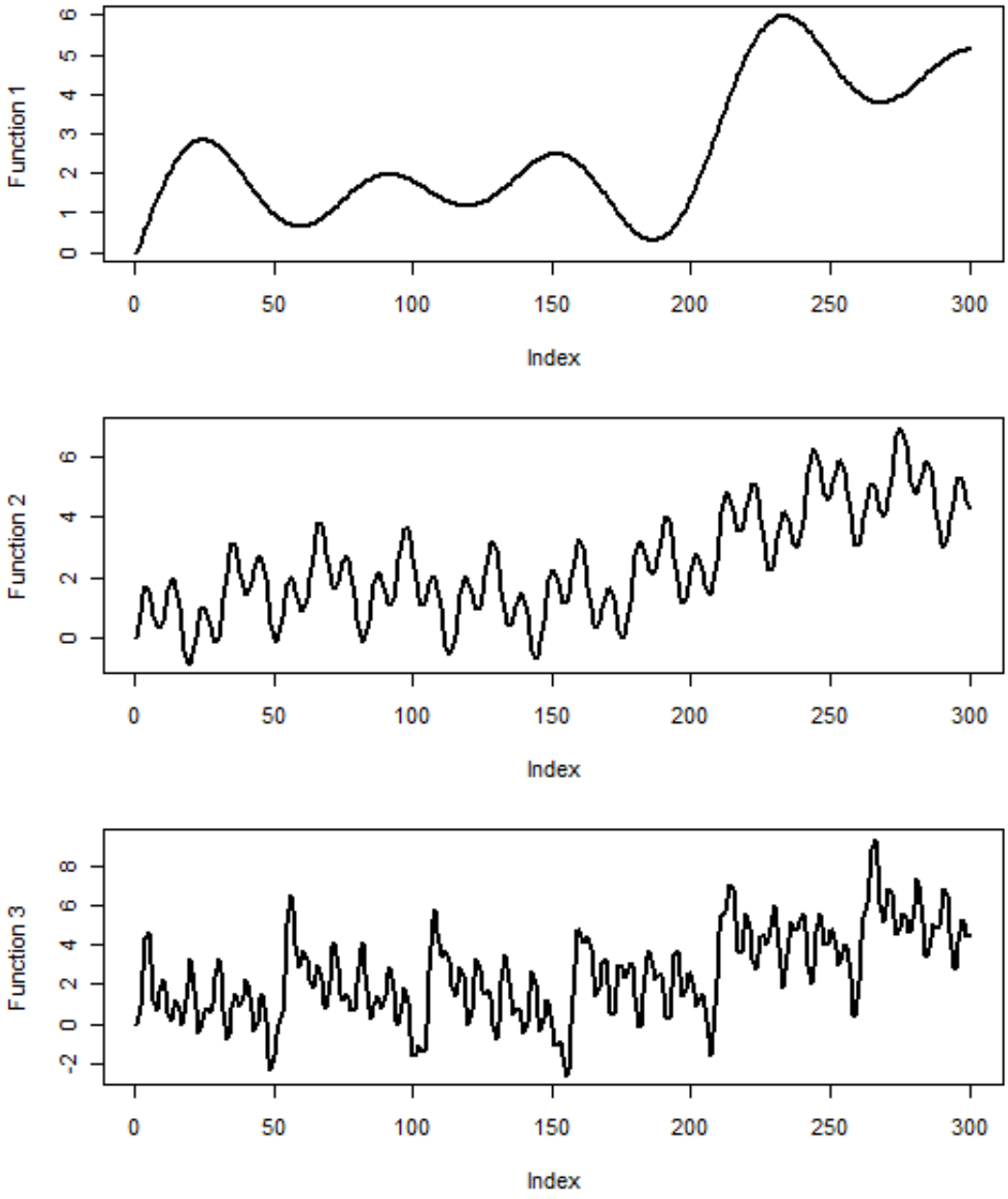

Figure 4.1: *Three Sinusoidal Functions. From top to bottom, respectively, are the graphs of* $f_1(x)$, $f_2(x)$, *and* $f_3(x)$.

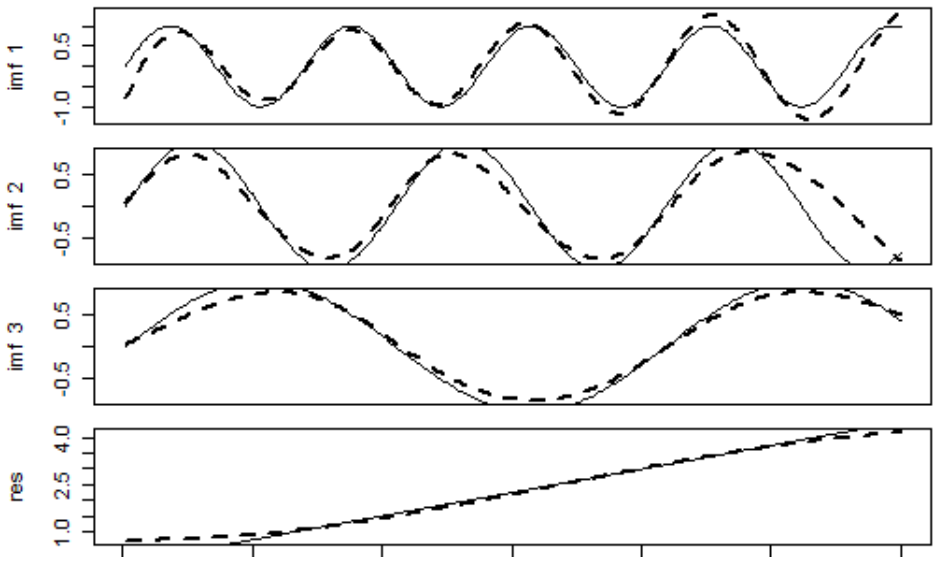

Figure 4.2: *Comparison between the IMFs and sinusoidal components of function f_1. The continous lines represent the sinusoidal components sin(3x), sin(2x), sin(x), and the linear component 0.5x. The dashed lines represent the IMF components with frequencies from highest to lowest and the trend line.*

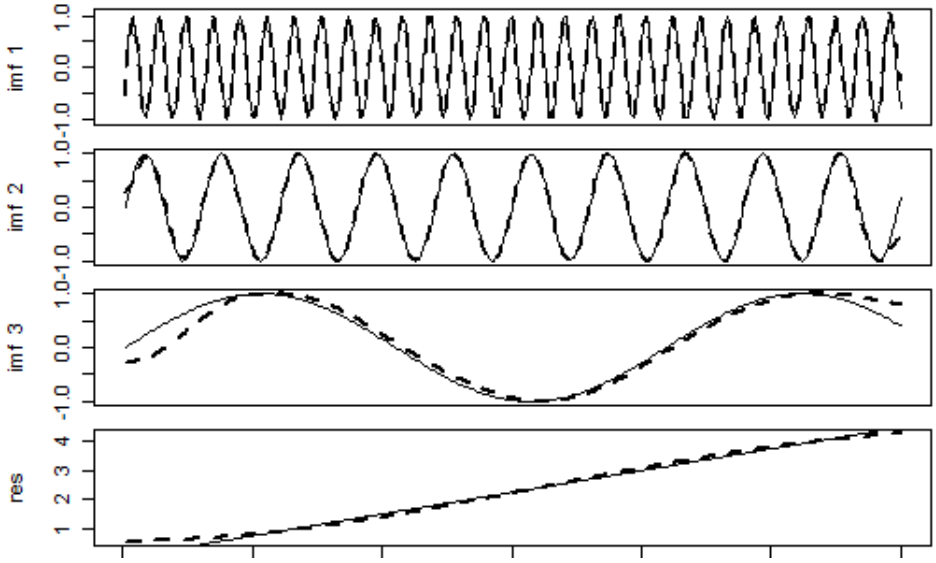

Figure 4.3: *Comparison between the IMFs and sinusoidal components of function f_2. The continous lines represent the sinusoidal components sin(20x), sin(7x), sin(x), and the linear component 0.5x. The dashed lines represent the IMF components with frequencies from highest to lowest and the trend line.*

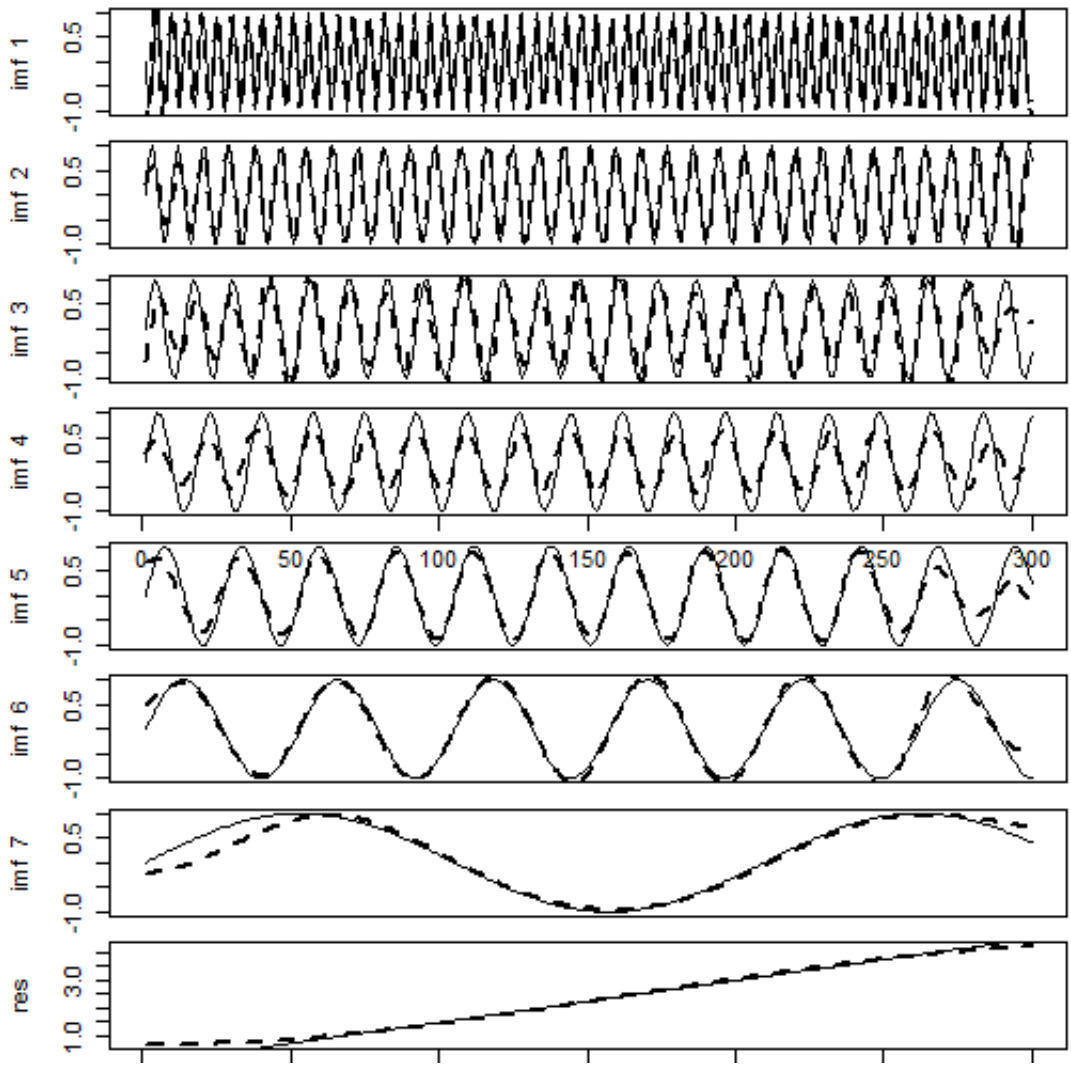

Figure 4.4: *Comparison between the IMFs and sinusoidal components of function f_3. The continuous lines represent the sinusoidal components sin(168x), sin(24x), sin(16x), sin(12x), sin(8x), sin(4x), sin(x), and the linear component 0.5x. The dashed lines represent the IMF components with frequencies from highest to lowest and the trend line.*

4.2 Baseline study

In this section, we attempt to determine whether it is feasible to apply EMD as a data preprocessing measure for forecasting purpose. First, the Acxiom workload data are studied to determine whether possible patterns exist as well as whether the workload data is stationary or not. Next, the IMF components resulting from decomposing the Acxiom workload data are examined. Finally, a simple prediction algorithm is applied to compare the performance between forecast using only the original workload data and forecast using EMD-based preprocessed data.

4.2.1 Workload Data

For the baseline study, the months containing abnormal data are removed. Abnormal data here means the unexpected absence of job arrivals in several consecutive hours. Figure 4.5 distinguishes between a month that has abnormal data and a month that does not. The exclusion of months with abnormal absence of data will help to reduce the effects of workload flurries [111] and allow a better degree of accuracy measurement in analyzing forecasting results. From the remaining months, the months from April to July 2006 are chosen as a continuous set of arrival data to be analyzed in detail by the EMD technique.

Based on the experimental data set (April-June 2006), the following observations can be made from the workday and weekend arrival behaviors:

- The daily arrival patterns of working days (Monday to Friday) are similar in shape but exhibit different magnitudes in peaks. For example, Figure 4.6 demonstrates the similarities in the arrival patterns from Monday to Friday of the first week of April 2006: The number of arrivals starts to increase around 08:00AM, dip around noon, and gradually decrease after 06:00PM.
- On the first Saturday of April, there is a peak around 08:00PM. This could be considered an outlier, since the remaining Saturdays in April in Figure 4.7 indicate a normal weekend behavior with a low and steady number of job arrivals throughout the days. However, further

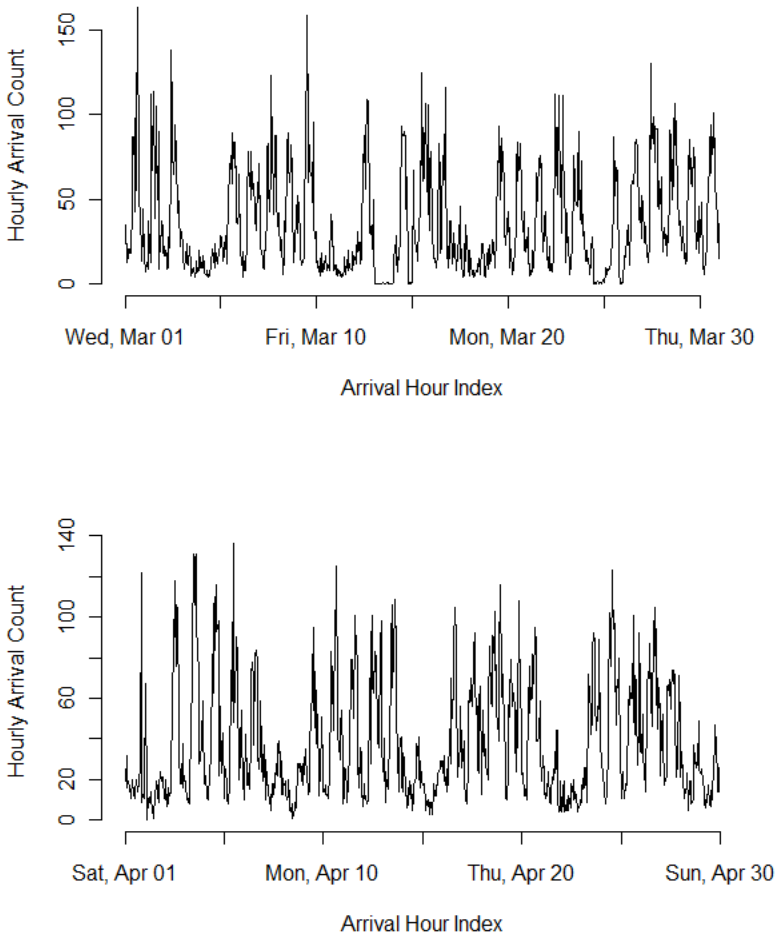

Figure 4.5: *The top frame demonstrates data from Mar 06 with an absence of job arrivals around Tuesday, Mar 14. The bottom frame shows the month of April 06 with no absence of job arrivals.*

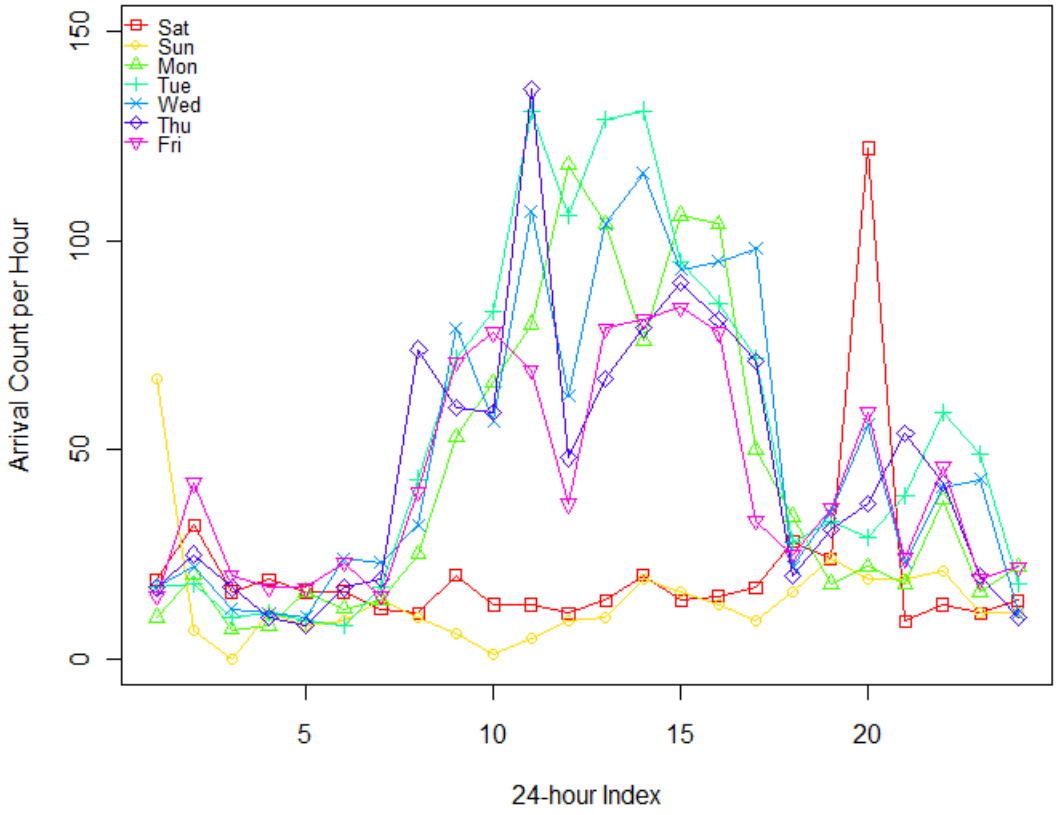

Figure 4.6: *Comparing arrival trend for seven days of the week.*

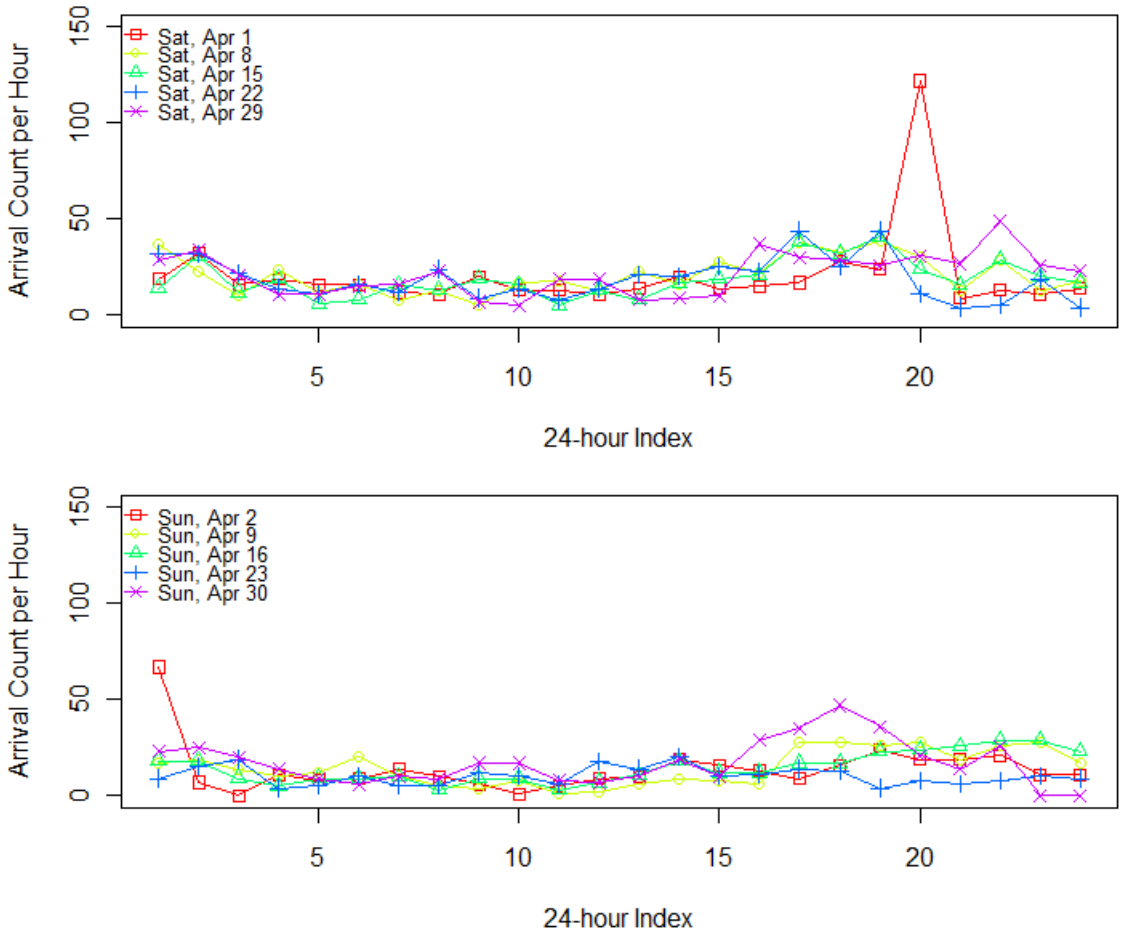

Figure 4.7: *Arrival trends for weekends of April, 2006*

examinations of weekends including both Saturdays and Sundays for the months April, May, and June, as illustrated in Figure 4.7, 4.8, 4.9 indicate that the behavior of Saturdays are neither completely weekday or weekend. Only the Sundays behave similarly on all sample data. This implies that the arrival trends during Saturday may still be under the influence of the work capacity during the week.

While the above observations are only made during the months of April, May, and June 2006, it is reasonable to assume that the remaining months of the data set also exhibit similar arrival patterns. However, it should be noted that the first month of the data set, January 2006, only had 16,984 jobs while the last month of the set, March 2007, had 42,353 jobs. This rate of increase is

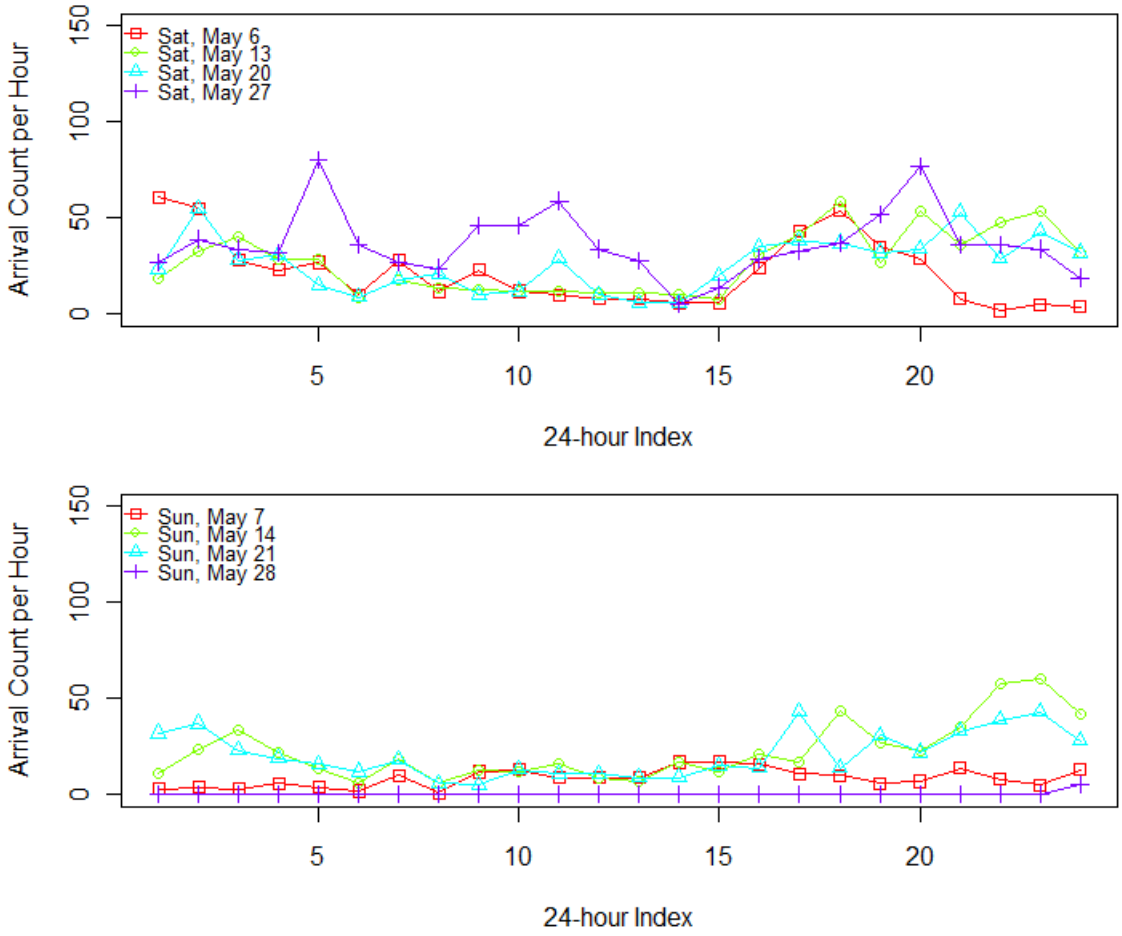

Figure 4.8: *Arrival trends for weekends of May 2006*

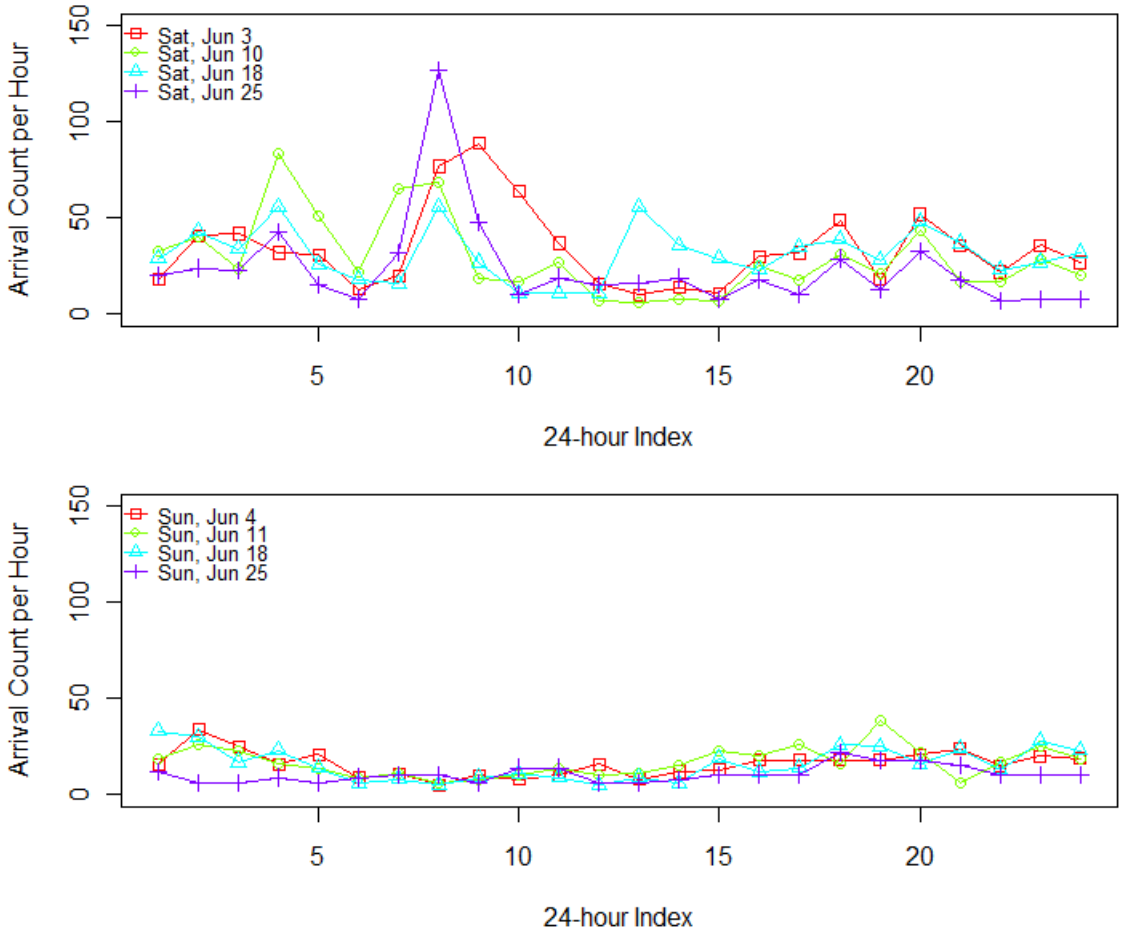

Figure 4.9: *Arrival trends for weekends of June 2006*

not monotonic throughout the data set. This means while the monthly data exhibit similarities in weekday and weekend cycles, they are not stationary.

4.2.2 Baseline Study of the Application of EMD for Forecasting Purposes

For this baseline study, we first need to show that it is logical to use the IMFs for forecasting purpose. Next by applying a simple weight-based algorithm on IMFs and trend of an arrival time plot, we test for the capability of IMFs versus the usage of the original time plot in forecasting. Based on the observations of the data in the previous section, the simple weight-based algorithm will attempt to predict the workload in the succeeding day (a 24-hour look-ahead) from:

- The arrival information of the previous day
- The differences between the similar two days of the previous week.

Validating the Usage of EMD

The validation of a characterization process compares the similarity of the real data against the synthetic data. Validation can be performed immediately upon generation of the synthetic data. In contrast, validation of a forecasting process is done by comparing the predicted data against the future data. This can only be done after the future has happened. While this might not seem to be difficult with traditional time series and probabilistic distributions, their timing functions can be extended into the future, it is different for EMD. Since the sifting process of EMD relies on empirical data, in addition to validating predicted data against actual data, it is necessary to show that the IMFs generated by the actual data resemble the IMFs generated by the predicted data. In a sense, this attempts to answer the question whether a synthetically generated IMF would work the same way as a timing function extended into the future. This provides a guarantee for validating and calibrating the forecasting system. We empirically show the above characteristics of the IMFs by looking at three different decomposition on the April 2006 data. The first decomposition, called Full, is done on the full April data. The second decomposition, called First Half, is done on the segment from hour 1 to hour 360. The third decomposition, called Second Half, is done on the

segment hour 360 until the last hour of the month. After the sifting process, the resulting IMFs from the sets First Half and Second Halfs are combined and mapped against the IMFs from the Full set. The followings observations are made:

- set Full generates 8 IMFs while sets First and Second Half, with only half the data of the Full set, can generate 7 IMFs. This indicates that within a month, the majority of arrival oscillations happen within a weekly cycle.
- Figures 4.10 and 4.11 show that when the first four IMFs with highest frequencies from each set are compared against each other, they match up with a high degree of accuracy even at the overlapping periods.
- Due to the differences in the number of IMFs within each group, the residues of the sets First Half and Second Half are compared against the sum of the residue and the last IMF of the set Full. Figure 4.12 shows that although the inner sections match up, the end effects do have an effect on the matching of low-frequency IMFs of the sets. On the other hand, the different residues also match up together nicely.

With these empirical results, it is demonstrated that IMFs from adjacent data segments capture a sense of continuity of the original data flow. Consequently, it is reasonable to use IMFs as the basis for forecasting purposes.

A Naive Weighted Forecasting Algorithm

In this section an approach common to machine learning is taken to develop an algorithm that use EMD's IMF components to predict arrivals. The algorithm consists of two phases, a training phase and a prediction phase. In the training phase, the weights are calculated from the training source and training target. In the prediction phase, these weights are applied to the prediction source in order to calculate the prediction target.

- Let n represent the number of hour indices included in the process. The value of n is identical for the source training day, the target training day, the source prediction day, the target prediction day, and for all the IMFs and generated trends from the above dates.

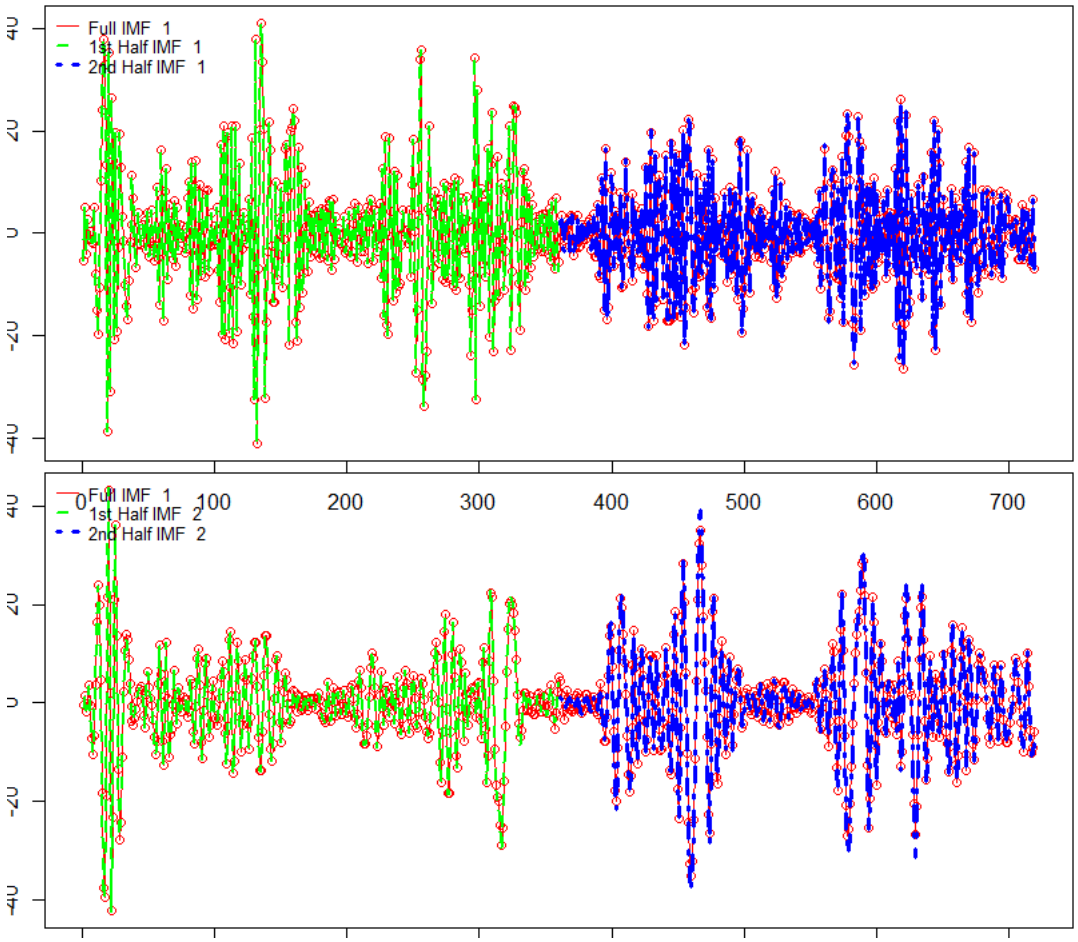

Figure 4.10: *Graphs of the first two IMFs of the Full, First Half, and Second Half sets*

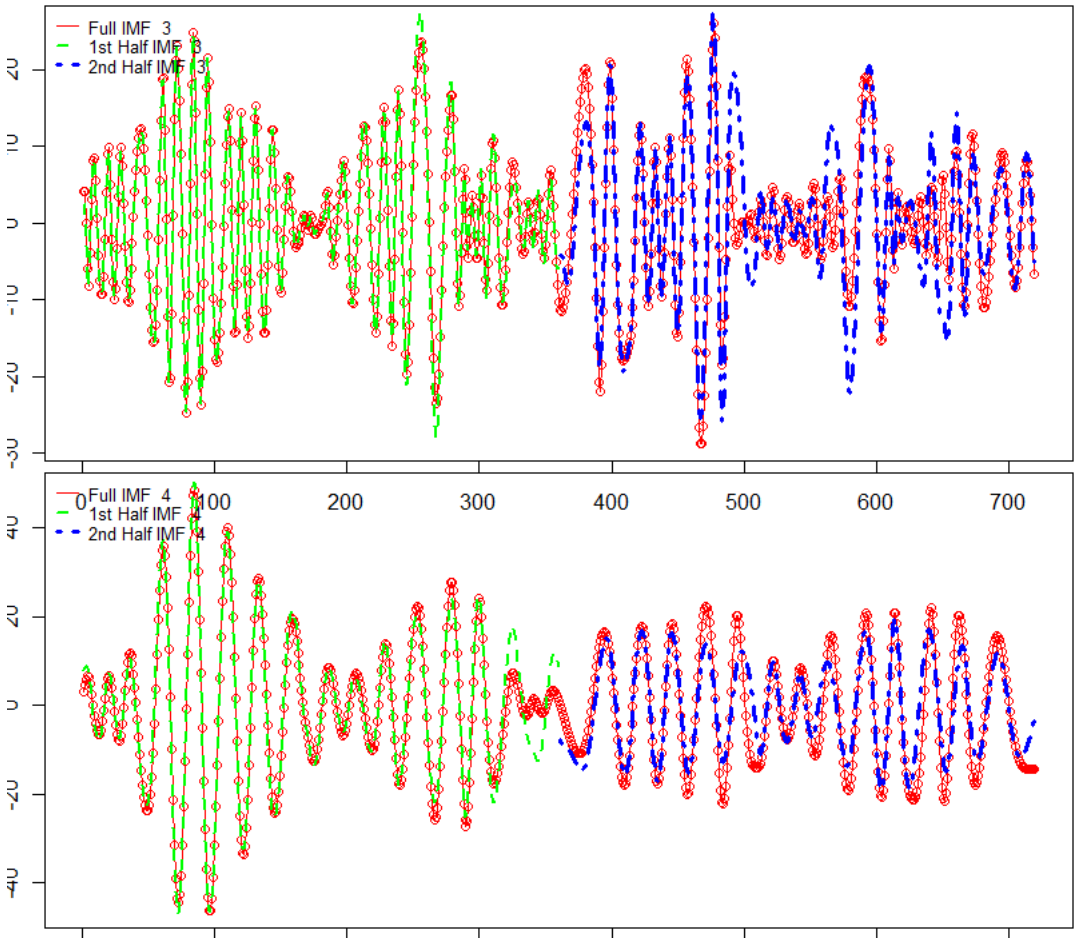

Figure 4.11: *Graphs of the next two IMFs of the Full, First Half, and Second Half sets*

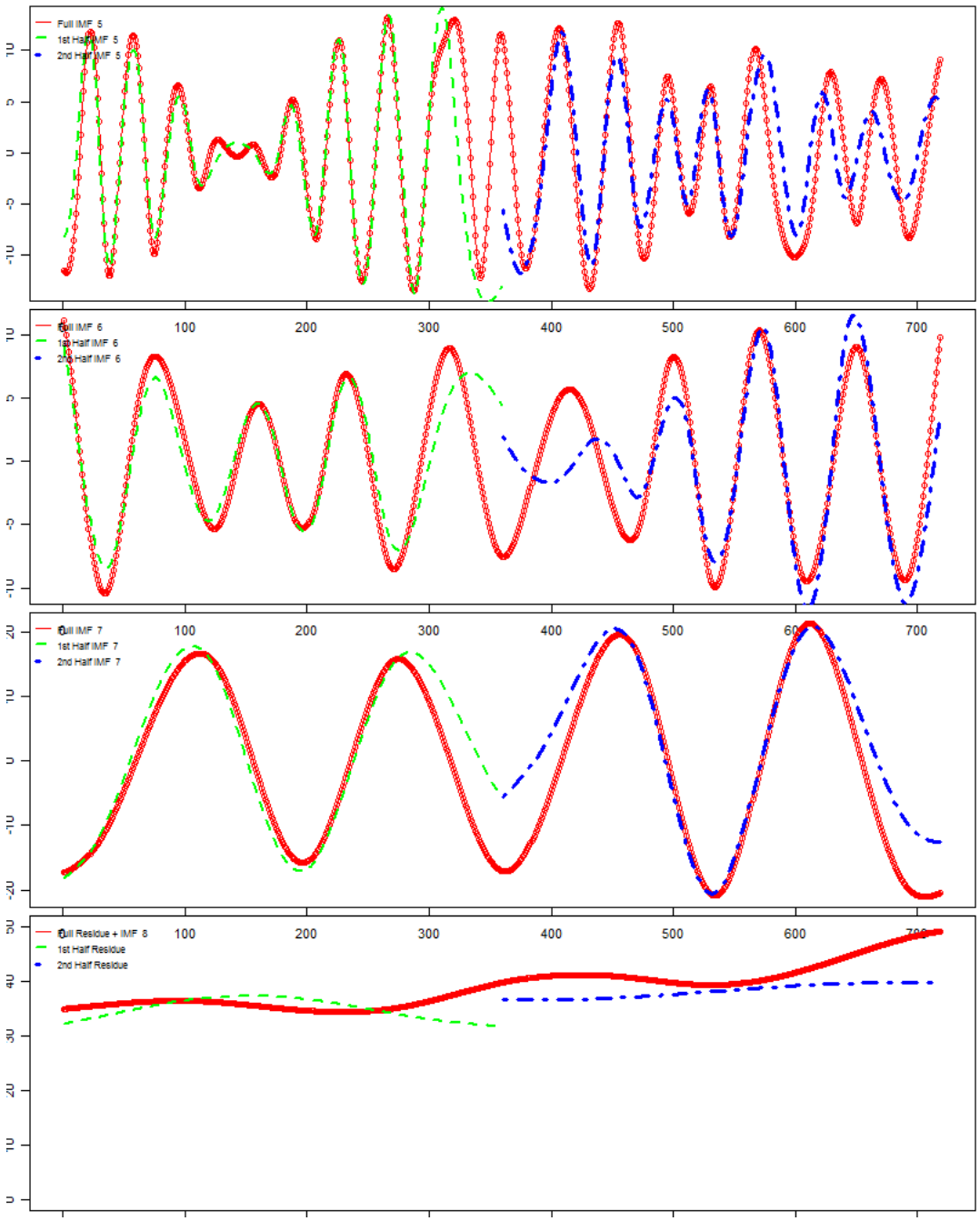

Figure 4.12: *Graphs of the last three IMFs and the trend of the Full, First Half, and Second Half sets*

- Let m represent the number of IMFs generated by the EMD. It is not guaranteed that the number of IMFs generated by the training source and prediction source will be identical. For this feasibility study, when the numbers are different, the set having the larger number will be reduced by adding the low frequency components to the final trend until the two numbers of IMFs are identical.
- Let $imfe_{ij}$ represent the value of IMF j for the hour index i of the source training, and let w_{ij} represent the weight of hour index i and IMF j with $i \in [0, n-1]$ and $j \in [0, m-1]$. The value of w_{ij} is between 0 and 1, and the modification step α of w_{ij} is 0.05. Initially, the values of all the weights are set to 1.
- Let se_i, st_i, pe_i, pt_i represent the source training, target training, source prediction, and target prediction of the hour index i, respectively.
- Let $imfp_{ij}$ represent the value of IMF j for the hour index i of the source prediction

The training algorithm is defined as followed:

```
for i = 0 to n-1
    t_i = 0
    for j = 0 to m-1
        t_i = t_i + w_ij * imfe_ij
    if (t_i > et_i)
        do
            for j = 0 to m-1
                if (imfe_ij >= 0)
                  w_ij = w_ij - α
                t'_i = 0
                for k = 0 to m - 1
                    t'_i = t'_i + w_ij * imfe_ij
                if (t'_i > et_i) continue
                else
                      w_ij = w_ij + α
                      break
    if (t_i < et_i)
        do
            for j = 0 to m-1
                if (imfe_ij < 0)
                  w_ij = w_ij + α
```

$t'_i = 0$
for k = 0 to m - 1
    $t'_i = t'_i + w_{ij} * imfe_{ij}$
if ($t'_i > et_i$) continue
else
    $w_{ij} = w_{ij} - \alpha$
    break

With the weights acquired from the training algorithm, the prediction target using the IMFs of the prediction source can be calculated as:

$$pt_i = \sum_{j=0}^{m} w_{ij} imfp_{ij}$$

To evaluate the accuracy of the algorithm, the Mean Average Percentage Error (MAPE) for the training and prediction processes is used. $MAPE_e$ represents the error margins between the source and target trainings, and $MAPE_p$ represents the error margins between the source and target predictions.

$$MAPE_e = \frac{\sum_{i=0}^{n-1} \frac{|es_i - et_i|}{et_i}}{n}$$

$$MAPE_p = \frac{\sum_{i=0}^{n-1} \frac{|ps_i - pt_i|}{pt_i}}{n}$$

4.2.3 Prediction Results

The algorithm was applied to the April 2006 data, with the training source and target being the Tuesday April 18 and Wednesday April 19. The prediction source and target are the Tuesday April 25 and Wednesday April 26. The EMD process was applied on the arrival data for 24 hours of those days. In the following experimental results, the same training and prediction sources and targets are maintained throughout. However, the ranges of data on which the EMD process is applied are different. Experiment 1 uses one week of data, from April 17 to April 25. Experiment 2 uses two week of data, from April 10 to April 25. Experiment 3 uses three week of data, from April 3 to

Exp.	Decomposition Range	IMF Count	Training MAPE	Prediction MAPE
1	One-week	6	5.35%	76.64%
2	Two-week	7	4.52%	69.55%
3	Three-week	8	5.16%	71.45%
4	Two-month	23	3.86%	70.71%
	No decomposition		0.00%	95.75%

Table 4.1: *Comparing MAPEs of the forecasts based on different data ranges and the original data set)*

April 25. Experiment 4 increases the range of the data for EMD all the way from March 1 to April 25. The number of IMF decomposed for each of these experiment, respectively, are 6, 7, 8, and 23. A comparison point is also included: the prediction result based on the above mentioned sources and targets. That is, the weight is simply calculated by the ratio of the training source and target.

Table 4.1 describes the recorded training and prediction MAPEs, and Figure 4.13 illustrates the graphs of the predicted data versus the actual data. From the table and the figure, the following observations are made:

- In all cases, the estimation MAPEs are highly accurate. This means that it is possible to apply the same algorithm across all the different decomposition sets and still able to calculate the appropriate sets of weights so that the estimation values are consistent.
- Overall, all the prediction MAPEs from the decomposition sets are better than the prediction MAPE of the data that has no been decomposed.
- Among the decomposition sets themselves, when the range of the data being decomposed is increased from one week to two weeks, the accuracy of the prediction MAPE is increased by 5%. However, when this range is increased further to three weeks and two months, respectively, there is not a significant improvement in prediction accuracy.
- While the MAPEs of Table 4.1 show undesirable errors (in the range of 70%), the graphs of Figure 4.13 indicates a visually pleasing forecast. That is, these forecasts can capture the burstiness of the daily arrival trend correctly.
- It should be noted that while the data range for decomposition spans at least one week, the

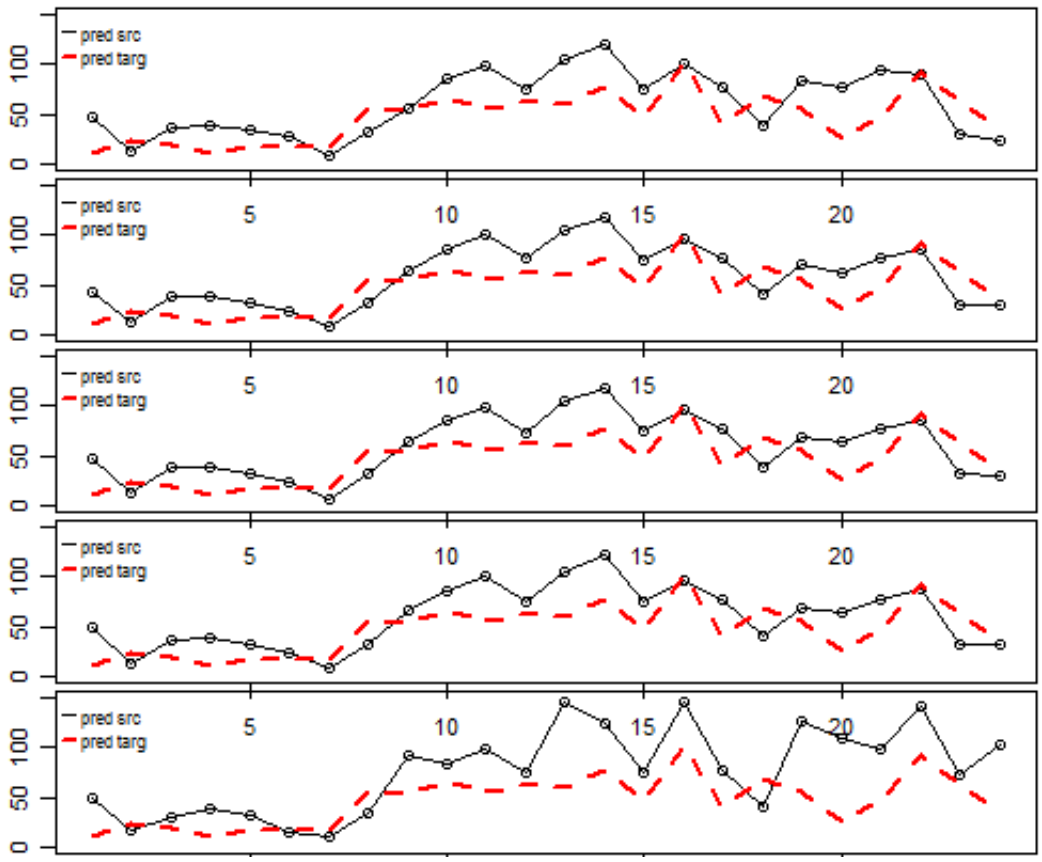

Figure 4.13: *Graphs of the forecasting results versus the actual data of the five different experiments (from top down): one-week decomposition, two-week decomposition, three week-decomposition, two month-decomposition, and no decomposition.*

actual data used in the prediction process contains only 24 hours of data, and it is being used to predict 24 hour ahead. This is long range prediction.

4.3 Conclusion

With these observations, it is reasonable to conclude that the utilization of EMD as a data processing platform before applying a prediction technique is indeed promising. While EMD shows evidence of the capability to decompose sinusoidal components from the data stream, manual calibrations are needed for this decomposition step. By separating the original time plot into the intrinsic mode functions, EMD creates an improved set of data for the prediction purposes. The initial observation shows that increasing the range of data for decomposition does increase the accuracy of the prediction. However, further investigations are needed to determine whether this holds true for more complicated prediction tasks and what the trade-off points between data range, prediction performance, and run time performance are.

Chapter 5

Workload Forecasting: Comparing Forecasting Approaches

The initial capabilities of EMD in forecasting have been demonstrated in Chapter 4. In this chapter, we utilize several more well known forecasting techniques in combination with EMD in order to investigate whether the EMD's decomposition step is useful for forecasting. This is done first on the set of sinusoidal functions from Chapter 4 and then on the sample Acxiom data. We also compare the wavelet-based approach to EMD in term of preprocessing data for forecasting.

5.1 Research Approach and Assumptions

The technique of decomposing time series data into components for forecasting purpose is not new. [112] describes the Bureau of Census' early adaptation and utilization of time series decomposition procedure called X11. This procedure first calculates the seasonal factor, then the cycle-trend, and finally the irregular component of the raw time series. The decomposition process of this procedure relies on either a 12-month (manual) or 15-month (automatic) moving average. [113] validates the appropriateness of this procedure against a standard Box-Jenkins' ARIMA (Autoregressive integrated moving average) model in fitting a seasonal time series. While X11 is effective, the fact that it is only a procedure limits the assessment of the series' statistical properties [114]. To overcome this, [114] proposes an alternative procedure based on the ARIMA model. The subsequence result not only produces the desired components but also estimated formulas which can be modified for further analysis. Simmons hypothesizes that a time series can be decomposed into components using a sinusoidal model [115]. These components are defined as trend, irregular, cyclical, and seasonal. Based on the assumption that the length of the seasonal cycle is known, the author demonstrates the decomposition of a time series into different sine waves in order to estimate the cyclical (seasonal and non-seasonal) and the irregular components. The trend component is estimated using a linear fit. Stating that the recombination of separate forecasts of trend-cycle,

seasonality, and irregular components is difficult to perform or only works relatively well, [116] presents the Theta method, in which the time series is only decomposed into a short term component and a long term component for forecasting purpose. [117] identifies and simplifies the procedure of the Theta method to simple exponential smoothing with drift. [118] decomposes the time series based on causal forces: domain-knowledge-based expectations related to the historical trends in the data.

5.1.1 Workload Assumptions

The procedures used in previous research make several important assumptions that might not be true for all time series. The seasonal adjustment process of the X11-based procedures ([112], [113], [114], [115], and [116] assumes that the seasonal component will be maintained throughout the entire series. The causal force procedure [118] requires prior knowledge about the causes that influence the time series. Next, we will analyze the Acxiom data to see whether it satisfies these assumptions.

Figure 5.1 contains three graphs. The top graph is the time plot of the arrival counts of the Acxiom workload for the month of April 2006. This month was chosen as the target window. The month of April 2006 is visually similar to other measured months of data and contains no abnormality. The graph in the middle is the autocorrelation graph with the lags ranging from 1 hour to 719 hours. The last graph is the partial autocorrelation graph with the lags ranging from 1 hour to 719 hours. The method of calculating the partial autocorrelation is similar to those of autocorrelation, except that when calculating the partial autocorrelation at lag k between X_t and X_{t-k}, the autocorrelation from lags 1 through k-1 is not included.

The time plot in Figure 5.1 gives the initial impression that the Acxiom data is stationary. This is not entirely true since the results from calculating the augmented Dickey-Fuller test on different lags length indicate that there are different stationary/non-stationary segments on the Acxiom data. The time plot also displays the possibility of weekly seasonal components. The autocorrelation plot shows a combination of linearly decaying and damped sinusoidal components.

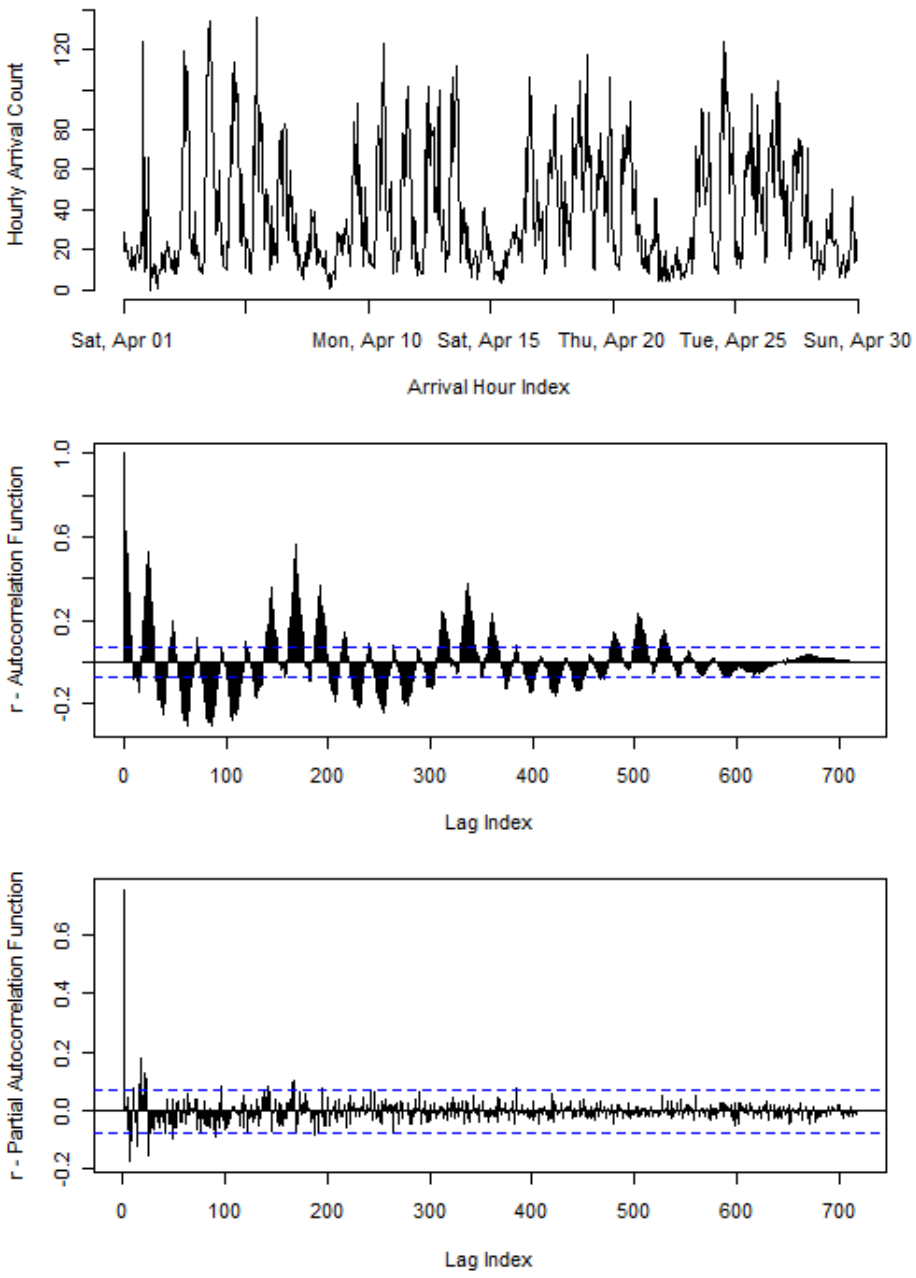

Figure 5.1: *Top Graph: Hourly Arrival Count of April 2006 beginning at 12:00 AM April 01, 2006 and ending at 11:59 PM April 30, 2006. Middle Graph: Autocorrelation Graph with the lags ranges from 1 hour to 719 hours. Bottom Graph: Partial Autocorrelation Graph with the lags ranges from 1 hour to 719 hours.*

The decaying rate of the remaining lags up to approximately 336 hours (two weeks) is small (with the exception of the first 12-hour lags). This shows that an appropriate model for this data could be a mixture of autoregressive and moving average models with order greater than one. In addition, the visuals of the autocorrelation plot shows high values at intervals, although it is difficult to isolate and identify these intervals. This suggests the possible existence of a seasonal autoregressive term and also explains why both decomposition methods *decompose()* and Loess-based *stl()* could not find any seasonal component on the Acxiom data. This is true for both the additive and multiplicative models. The partial autocorrelation function identifies a few possible lags for the autoregressive term (the values lying outside the two dotted lines (the 95% confidence level) in the Partial Autocorrelation graph of Figure 5.1.

5.1.2 Assumptions of Decomposed IMFs

The arrival histogram of April 2006 is decomposed into IMF components and trend, as illustrated in Figure 5.2. However, there are no known functional components of the time plot in order to measure whether the IMF components actually are the hidden patterns. In order to compensate for this problem, we look at the IMF components and confirm two assumptions:

1. Each IMF component exhibits a relatively regular periodic pattern as compared to the original data.
2. These periodic patterns make sense with respect to the known work-hour schedule of the environment.

To confirm the first assumption, we look at autocorrelation values of the original data as well as the IMF components with lagging distance going back as far as 336 hours, or two weeks. The autocorrelations of the IMFs components and the trend are shown in Figures 5.3 and 5.4. From the figures, the two highest-frequency IMFs are essentially alternating positive and negative with only a few significant but unrelated spikes. As a result, it is likely that these two IMFs contain mostly random information with a few patterns that are too minor to detect. However, similar to the autocorrelations of the original data, the autocorrelations of the remaining IMF components

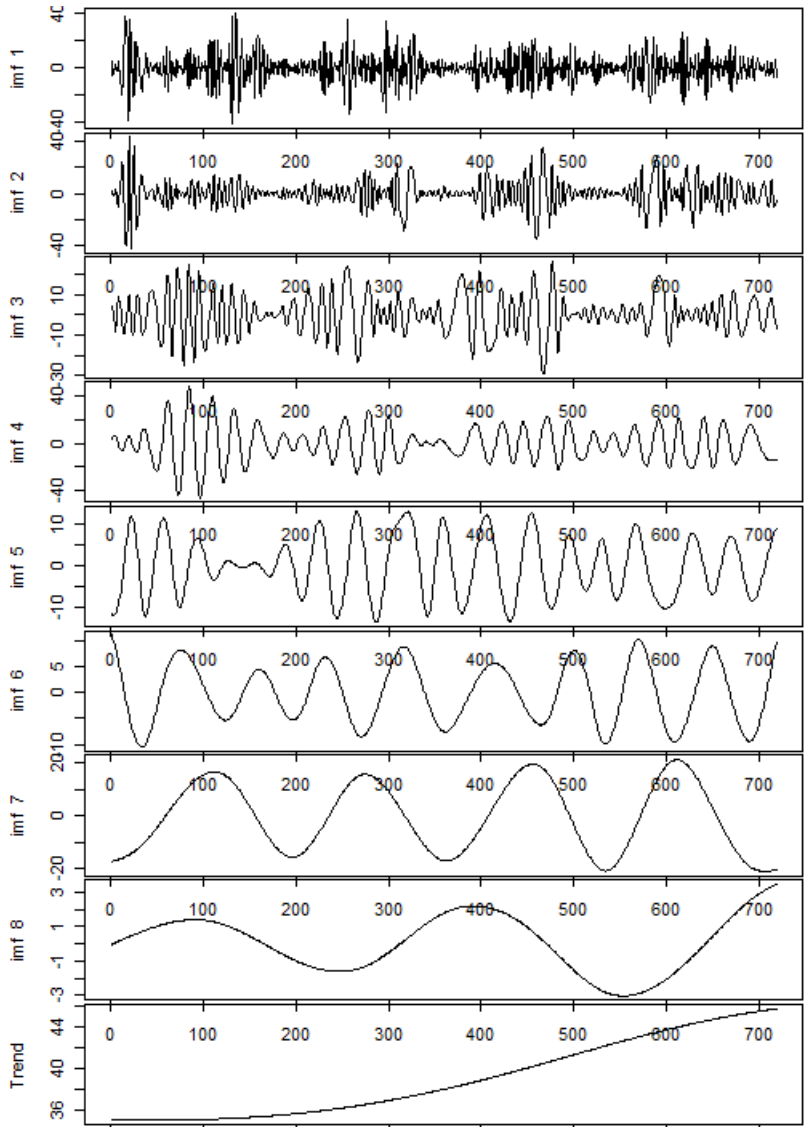

Figure 5.2: *Graphs of the decomposed IMFs and trend of April 2006. The top graph is the IMF with highest frequency. The next-to-last graph is the IMF with the lowest frequency, and the last graph is the trend.*

IMF	Mean	Standard Deviation
1	2.93	1.26
2	6.12	1.13
3	19.625	6.14
4	24.07	0.86
5	38.12	8.45
6	82.75	8.45
7	170	single point data
8	336	single point data

Table 5.1: *Average distance between group of high value at fixed interval of the autocorrelation functions of IMF components (in hours)*

3-8 also have groups of high values at fixed intervals. Compared to the original data, it is easier to identify the fixed interval for the autocorrelations of the IMF components. Figures 5.3 and 5.4 confirm the second hypothesis, as they enable the calculation of the average distance between the local optima for each autocorrelation of IMFs. Table 5.1 shows the means and standard deviations of the distances between the local optima of each IMF's autocorrelation function. Table 5.1 shows that each IMF carries an averaged periodic pattern. The patterns of the first four components are consistent with small standard deviations. The expected pattern of 24-hour is shown at IMF 4, with a 0.86 for standard deviations. From Table 5.1 and Figures 5.3 and 5.4, we can bserve that the following (approximate) periods exist within the data: 3 hours, 6 hours, 24 hours, 48 hours (weekends), 86 hours (weekdays), 168 hours (full week), and 336 hours (two week).

By common intuition, it would have suggested that the appropriate autoregressive terms of the IMFs would be 3, 6, 24, 48, 86, 168, and 336 respectively for a forecasting approach using ARIMA. However, the partial autocorrelation graphs of the IMF components in Figures 5.5 and 5.6 show that this is not true. In fact, most, if not all, the significant autoregressive terms are less than 9. This means that the observed correlations on 3, 6, 24, 48, 86, 168, and 336 are not considered periods of auto regression.

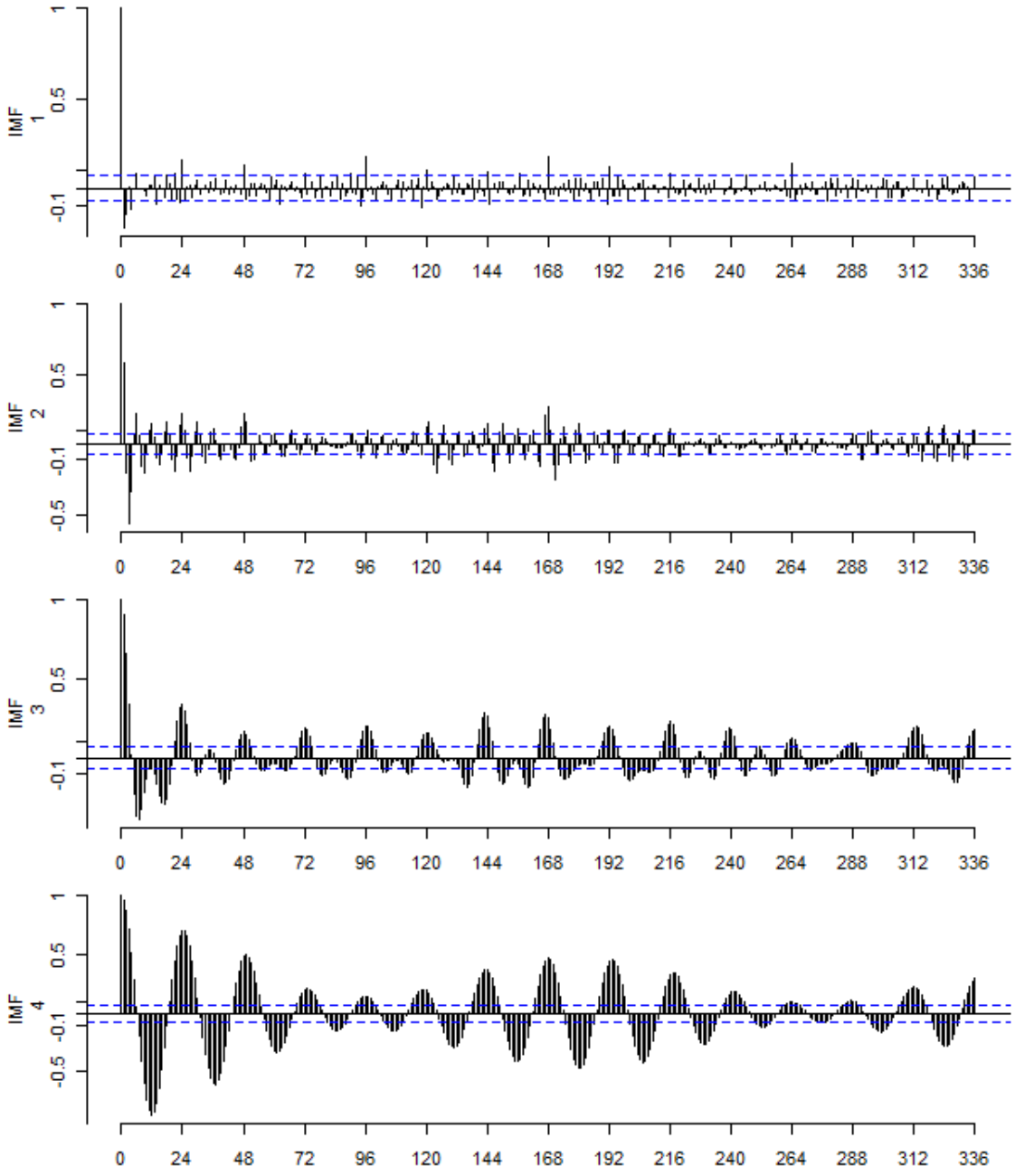

Figure 5.3: *Autocorrelation of the first four IMF components with a 336-hour lag*

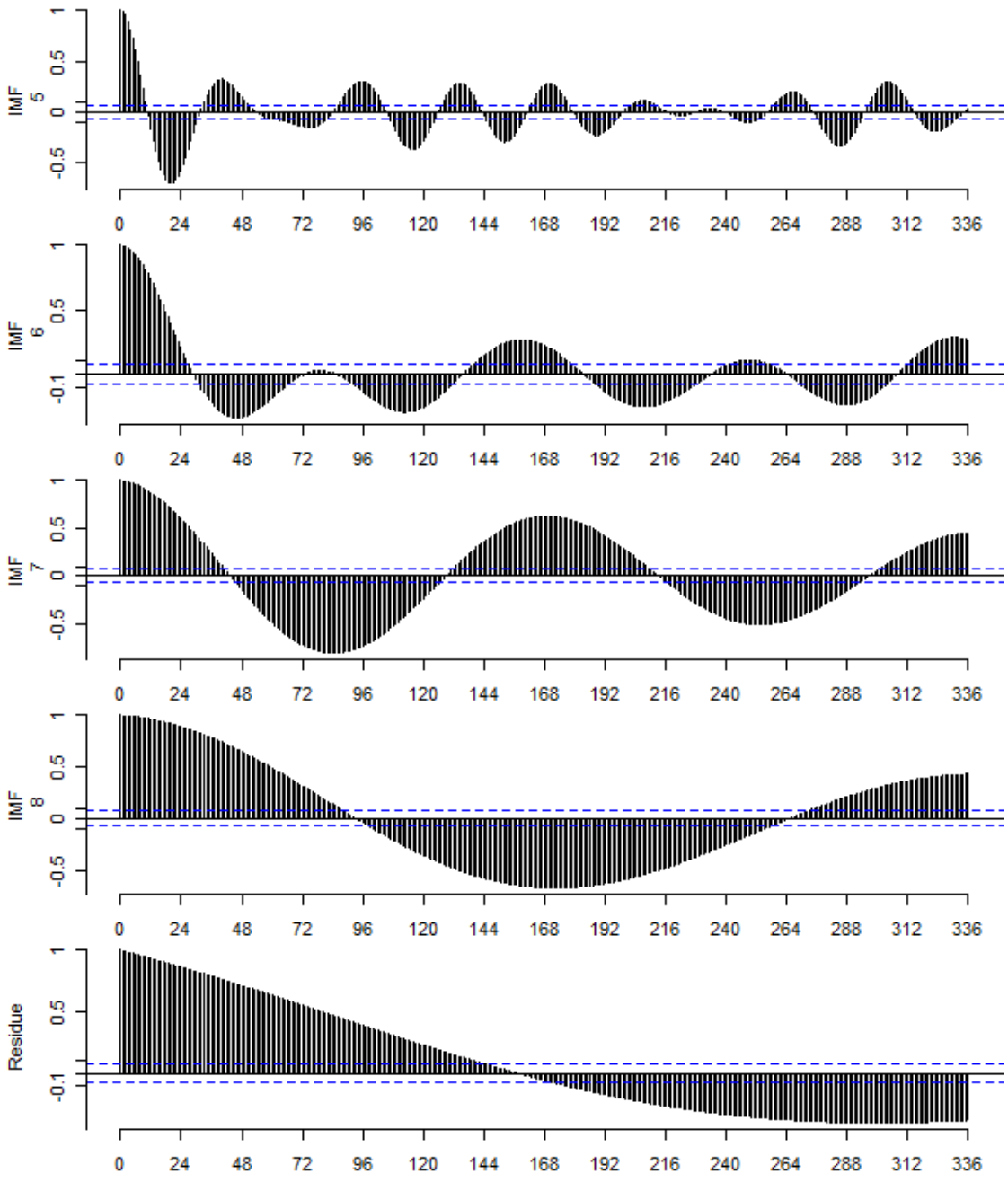

Figure 5.4: *Autocorrelation of remaining IMF components and the trend with a 336-hour lag*

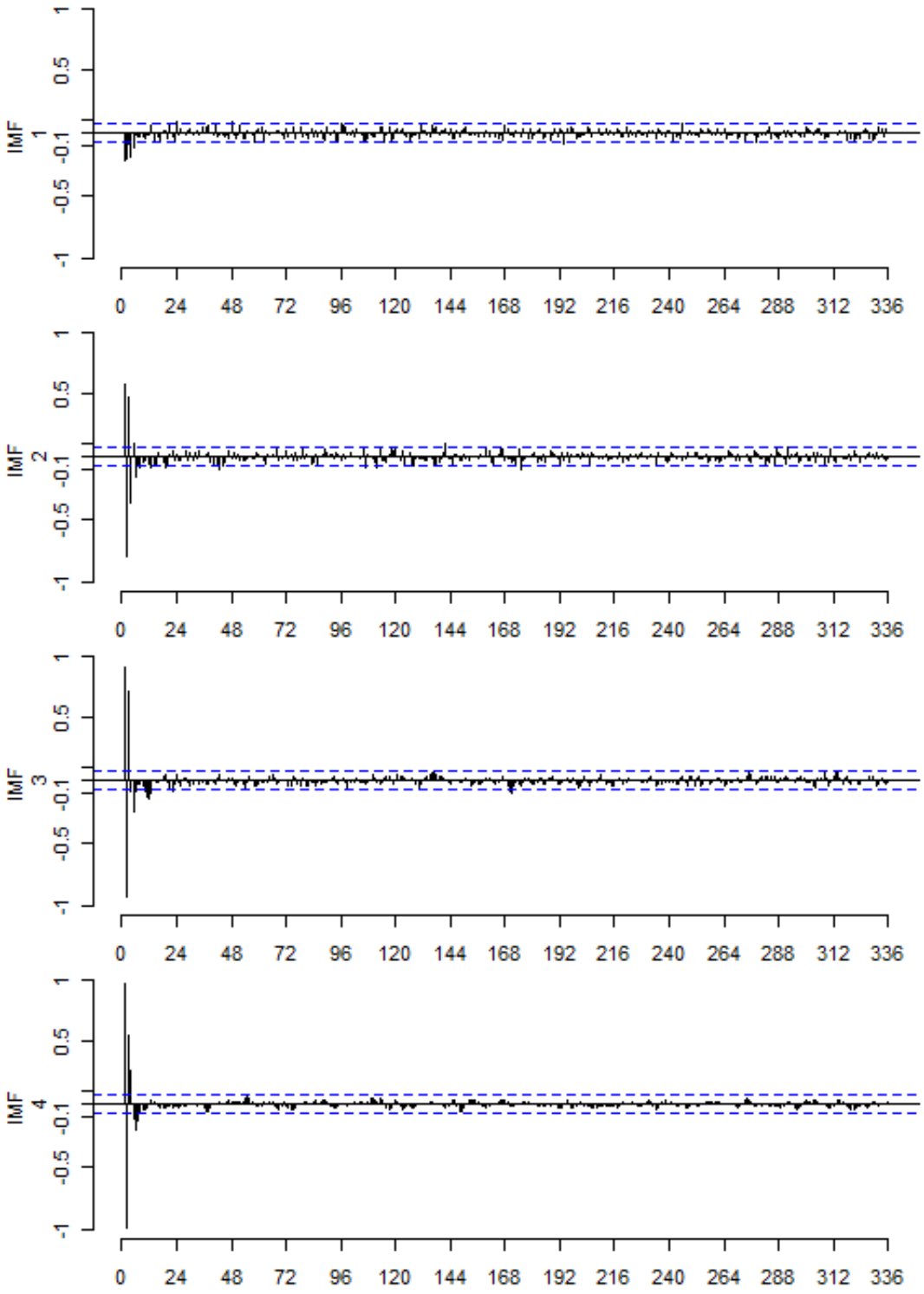

Figure 5.5: *Partial autocorrelation of the first four IMF components with a 336-hour lag*

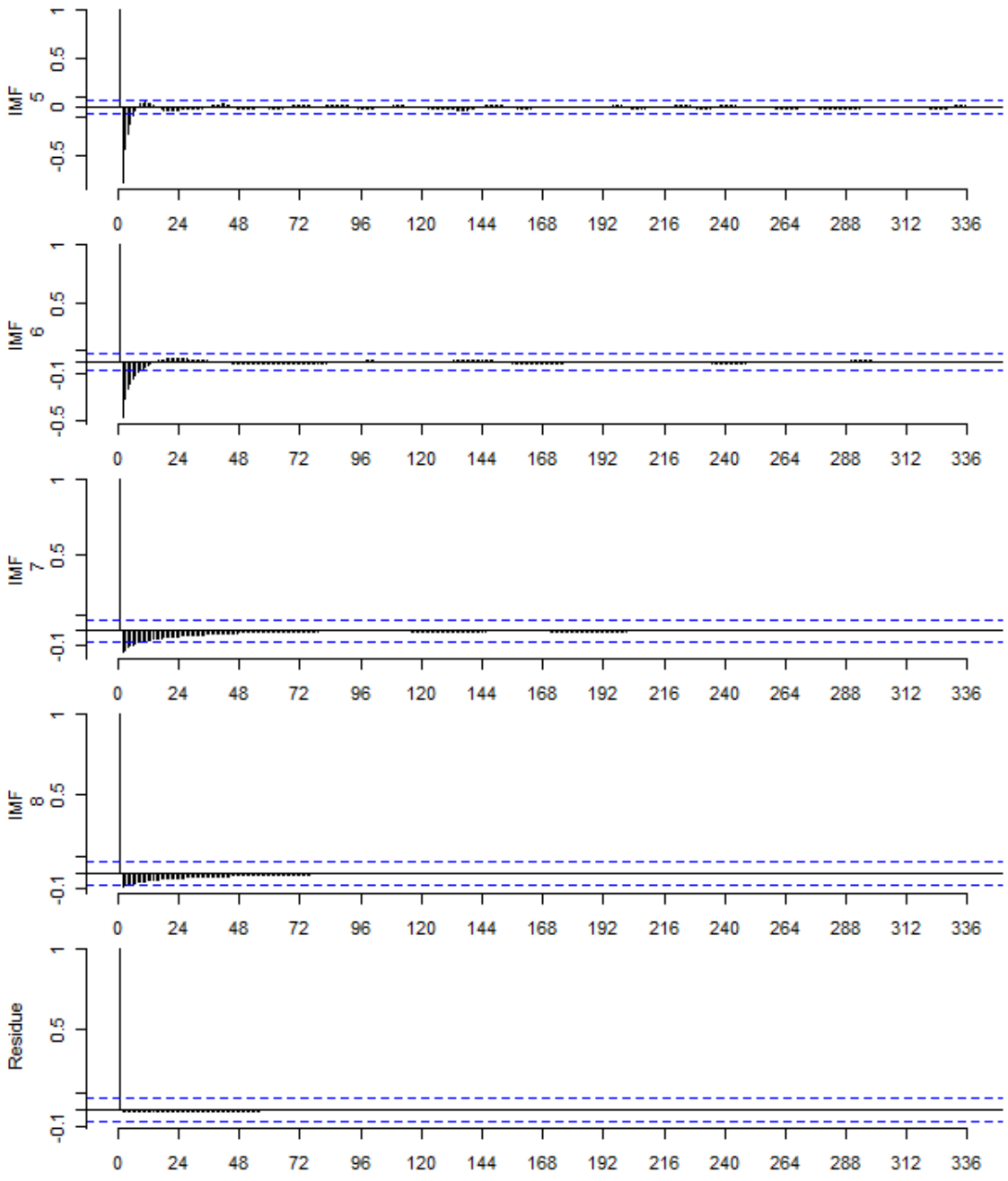

Figure 5.6: *Partial autocorrelation of the remaining IMF components and the trend with a 336-hour lag*

5.2 Forecasting Study and Analysis

It has been shown that EMD carries the potential of forecasting arrival data with a naive weighted forecasting procedure [4]. To further investigate the applicability of EMD in forecasting further, we use a set of well known time series forecasting techniques:

- ARIMA (Autoregressive Integrated Moving Average)
- ETS (Exponential Smoothing - the State Space version by [119]
- VAR (Vector Autoregressive)

Among these techniques, ARIMA will be applied for both the original data and the decomposed components. The VAR method can only be applied to the set of decomposed components. The ETS method, also a decomposition-based method, will be applied on the original data only. The difference between the applications of VAR and ARIMA to the set of decomposed components is that while VAR uses IMF components for prediction, it also incorporates the relationship between the components into the process. The implementations of ARIMA and ETS are from the R package *forecast* developed by [119] and the VAR implementation is from the R package *VARS* by [120]. In the subsequent figures and tables of this section, the different forecast combinations are denoted as follows: EMD/VAR, Original Data/ARIMA, Original Data/ETS, and EMD/ARIMA.

5.2.1 Forecasting Sample Sinusoidal Functions

From the analysis in Chapter 4, it was shown that EMD can successfully decompose the sample functions into IMFs that resemble the functions' components. The sample functions used are:

$f_1(x) = sin(x) + sin(2x) + sin(3x) + 0.5x$

$f_2(x) = sin(x) + sin(7x) + sin(20x) + 0.5x$

$f_3(x) = sin(x) + sin(4x) + sin(8x) + sin(12x) + sin(16x) + sin(24x) + sin(168x) + 0.5x$

Since EMD can decompose these functions into IMFs that resemble the functions' sinusoidal components, the forecasting process using these IMFs should be more accurate when compared to the traditional methods using unprocessed data. In other words, the functional fitting process done

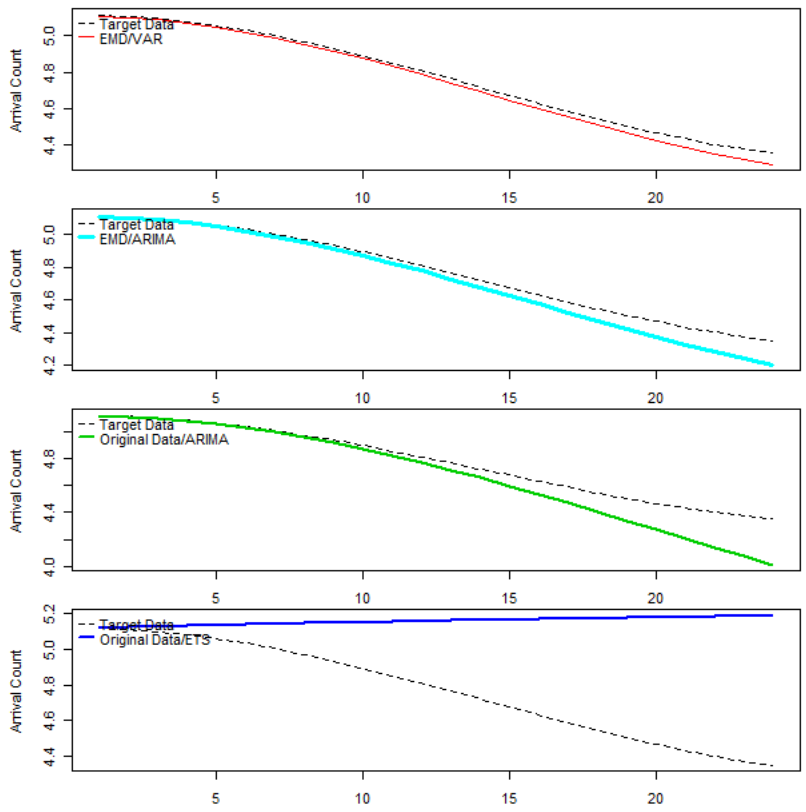

Figure 5.7: *Compare the forecasting results of EMD-based and traditional methods on function* f_1.

on simple functions (the decomposed IMF components that match the original sinusoidal components) should be easier than the fitting process done on a complex function (the original functions). Results of the forecasting for f_1, f_2, and f_3 are shown in Figures 5.7, 5.8, and 5.9 and Table 5.2. The columns of Table 5.2 show the MAPE values on the different forecast window ranges: 3, 6, 12, and 24 hours. In the Table and Figures, the forecasting results from the EMD/VAR are significantly better throughout the experiments. The results of EMD/ARIMA are also better than the traditional methods (ARIMA and ETS), although as the forecast range goes further into the future, this level of accuracy quickly decreases. In addition, the ARIMA method on unprocessed data and the EMD/ARIMA perform well in this case also, particularly within the first 10 forecast steps into the future.

Table 5.2: *Accuracy Measurement Comparisons using Mean Averaged Percentage Error*

	3 Hrs forecast window	6 Hrs forecast window	12 Hrs forecast window	24 Hrs forecast window
Forecast Approach				
$f_1(x) = sin(x) + sin(2x) + sin(3x) + 0.5x$				
EMD/VAR	0.71%	0.91%	1.42%	3.11%
EMD/ARIMA	0.72%	0.95%	1.81%	47.8%
ARIMA	0.04%	0.24%	1.69%	13.7%
ETS	2.07%	5.96%	18.2%	47.84%
$f_2(x) = sin(x) + sin(7x) + sin(20x) + 0.5x$				
EMD/VAR	11.72%	11.79%	13.28%	14.97%
EMD/ARIMA	13.94%	21.59%	32.19%	144.5%
ARIMA	21.35%	55.79%	52.58%	119.3%
ETS	61.21%	165.7%	160.1%	131.3%
$f_3(x) = sin(x) + sin(4x) + sin(8x) + sin(12x) + sin(16x) + sin(24x) + sin(168x) + 0.5x$				
EMD/VAR	2.61%	6.04%	12.64%	35.62%
EMD/ARIMA	49.37%	106.9%	223.7%	311.6%
ARIMA	12.42%	190.6%	200.0%	243.8%
ETS	68.44%	95.81%	198.0%	220.4%

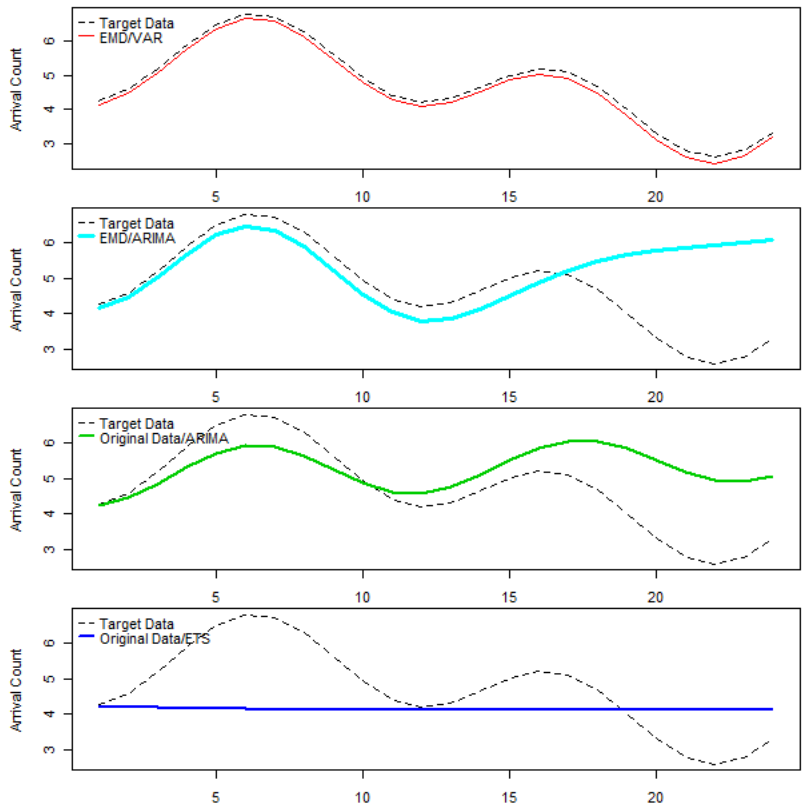

Figure 5.8: *Compare the forecasting results of EMD-based and traditional methods on function* f_2.

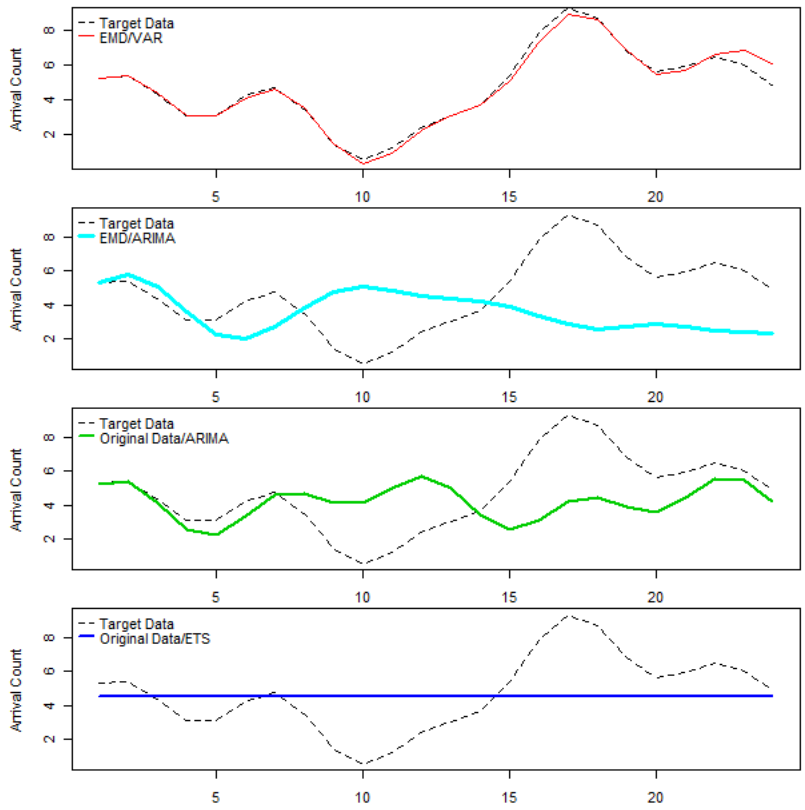

Figure 5.9: *Compare the forecasting results of EMD-based and traditional methods on function* f_3.

5.2.2 Forecasting Acxiom data

In this initial set of experiments, the forecast window varies from 12, 24, and 48 hours. Based on the autocorrelation analysis, the history window is chosen to be at least one week (168 hours). In other words, one week of data is used to predict 12 to 48 hours into the future. Additionally, due to the differences between weekdays and weekends, the forecasting experiments are divided further into three cases:

- Case 1: Using a previous week (weekdays and weekends) to forecast the next 48 hours
- Case 2: Using seven previous weekdays (Thursday, Friday, then Monday to Friday) to forecast the next 48 hours of weekdays
- Case 3: Using seven previous weekends days(Saturdays and Sundays) to forecast the next 48 hours of weekends.

We stop at the two-week mark, since the autocorrelation function has clearly indicated a lack of correlation beyond three weeks. As all techniques utilized here begin by fitting the known data to a functional form before using this form for forecasting purpose, it is necessary to show how well the data are fitted. This, in a sense, similar to the training phase of the naive weighted forecasting algorithm. Figures 5.10, 5.11, and 5.12 demonstrate the initial fits for the three cases.

In all three cases, the forecasting combinations using ARIMA and ETS fit the known data values better than those of the VAR. Among the combinations using the ARIMA/ETS, the combinations that run on the original data fit better than the one that run on the decomposed data (IMFs). On the case of the VAR technique, this could be explained that while there are relationship among the IMFs (basis for the VAR technique to be applied), these relationships are not strong and do not last throughout the 168-hour long data set. The optimal lag used in the VAR is 48-hour.

For accuracy measurements, a number of standard methods are used: RMSE (Root Mean Squared Error), sMAPE (Symmetrical Mean Absolute Percentage Error), and MASE (Mean Absolute Scaled Error). While RMSE and sMAPE have been in used for a long time, MASE is a more recent method of measuring forecasting accuracy and was developed by [121]. MASE can

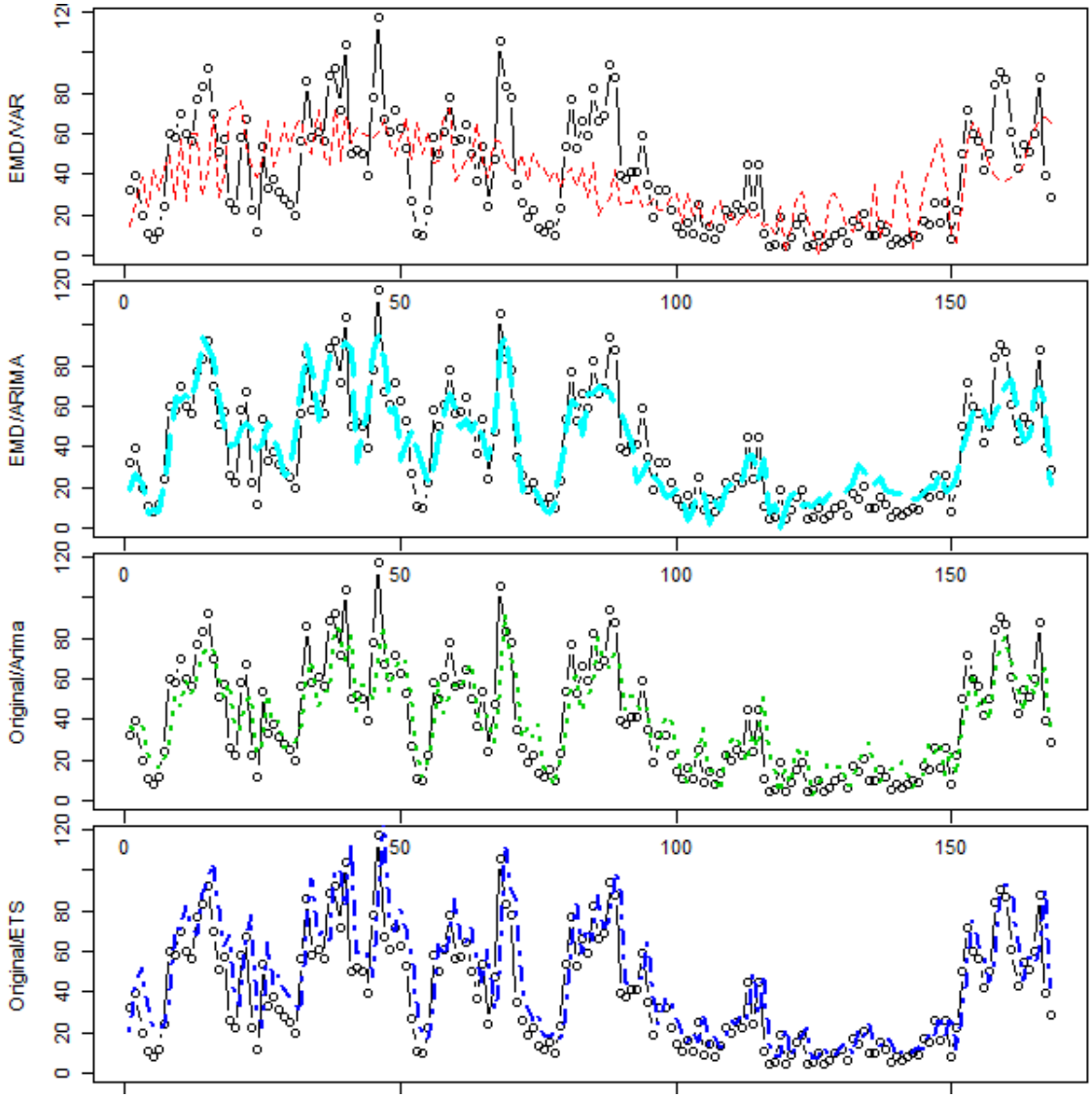

Figure 5.10: *Comparing initial fitting accuracy of different forecasting techniques in case 1 where the look back range includes all days of week*

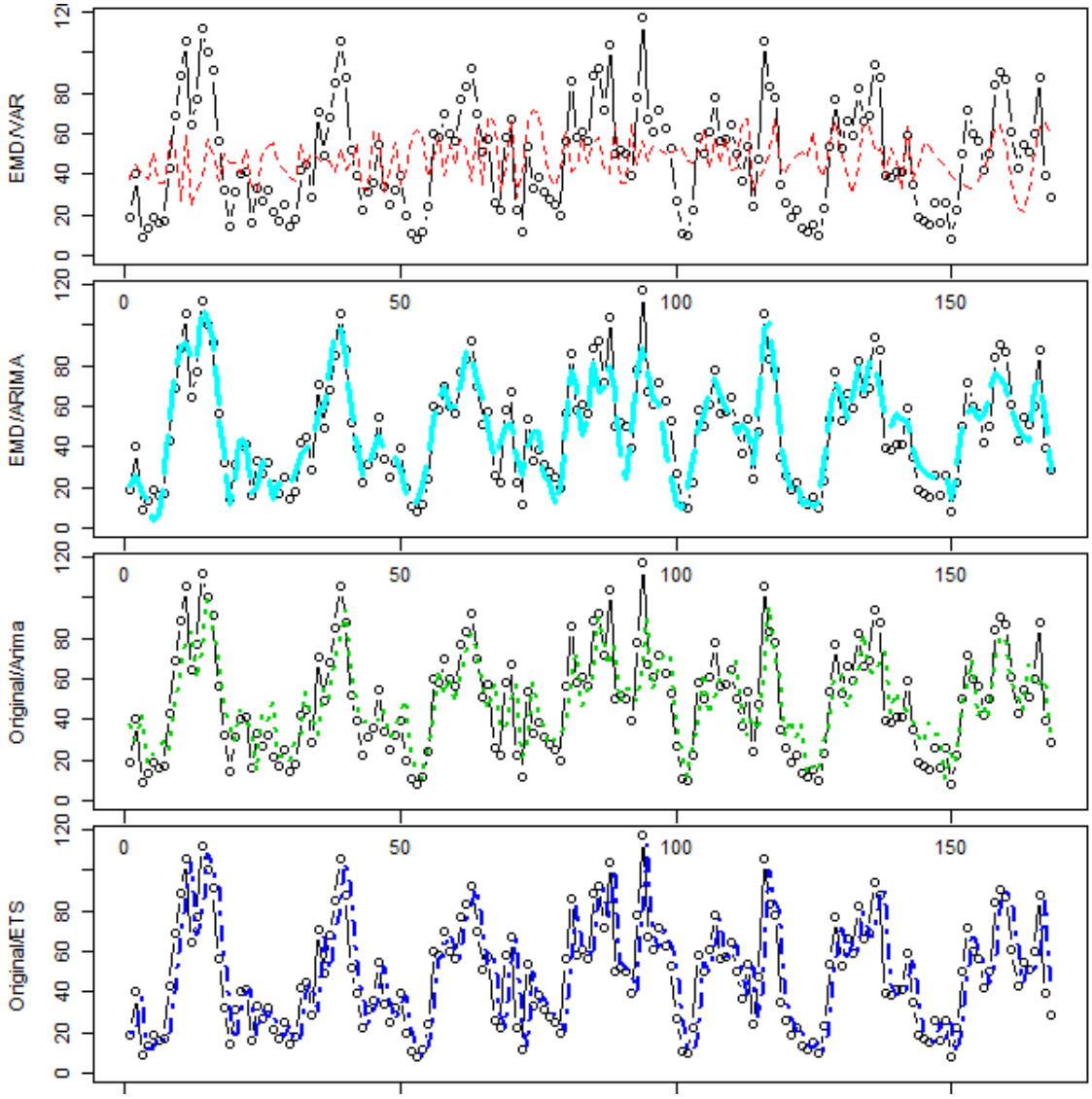

Figure 5.11: *Comparing initial fitting accuracy of different forecasting techniques in case 2 where the look back range includes only weekdays*

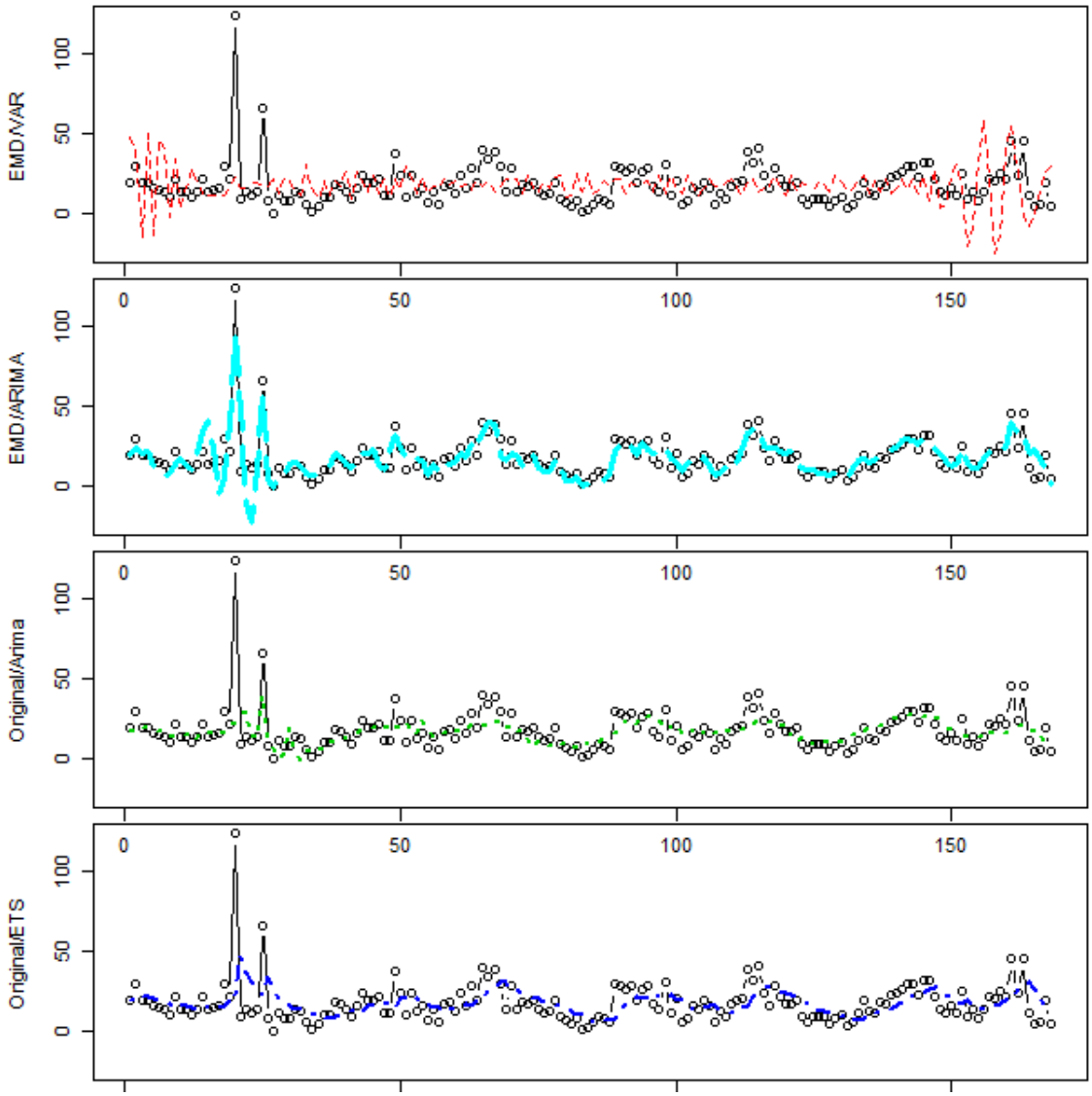

Figure 5.12: *Comparing initial fitting accuracy of different forecasting techniques in case 3 where the look back range includes only weekends*

	Case 1: All Week			Case 2: Weekdays			Case 3: Weekends		
Error Msm.	12	24	48	12	24	48	12	24	48
RMSE									
EMD/VAR	46.47	55.92	125.84	32.70	43.51	39.22	15.56	22.44	30.84
EMD/ARIMA	32.77	39.58	82.47	27.74	29.78	28.32	9.61	8.36	28.00
ARIMA	33.13	39.63	33.15	34.90	36.24	31.13	8.23	9.16	28.17
ETS	27.77	33.27	54.74	34.65	44.88	38.03	6.19	5.66	29.64
sMAPE									
EMD/VAR	0.35	0.34	0.49	0.36	0.36	0.33	0.41	0.42	0.38
EMD/ARIMA	0.36	0.33	0.42	0.33	0.26	0.25	0.73	0.46	0.46
ARIMA	0.37	0.34	0.30	0.39	0.31	0.28	0.34	0.33	0.36
ETS	0.33	0.27	0.36	0.36	0.38	0.34	0.26	0.23	0.34
MASE									
EMD/VAR	2.61	2.87	5.82	2.13	2.30	1.85	2.25	3.58	1.85
EMD/ARIMA	2.07	2.00	3.60	1.77	1.52	1.30	1.24	1.31	1.58
ARIMA	2.08	2.019	1.53	2.24	1.84	1.47	1.29	1.76	1.64
ETS	1.78	1.63	2.63	1.97	2.19	1.7	0.97	1.06	1.62

Table 5.3: *Accuracy Measurement Comparisons (closer to zero is better)*

be calculated as:

$$MASE = \frac{1}{n}\left|\frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}(Y_i - Y_{i-1})}\right|$$

In the above equation, in a period t from 1 to n, e_t is the absolute value of the difference between the actual value Y_t and the forecast value F_t. The calculation of MASE is basically the comparison between the error of the forecast technique to be analyzed and the error of a one-step naive forecast method. The usage of MASE is very useful in comparing different forecast methods [121]. While the forecast range is extended to 48 hours, this does not affect the accuracy of the shorter forecasts. Therefore, the two other shorter ranges (12-hour and 24-hour) are also observed. Figures 5.13, 5.14, and 5.15 demonstrates the visual comparisons between the forecast combinations. Table 5.3 gives a numerical comparison based on the accuracy measurements.

From the table and figures, the following observations can be made:

- With the exceptions of the 12-hour forecast of case 2, numerically, the EMD/VAR com-

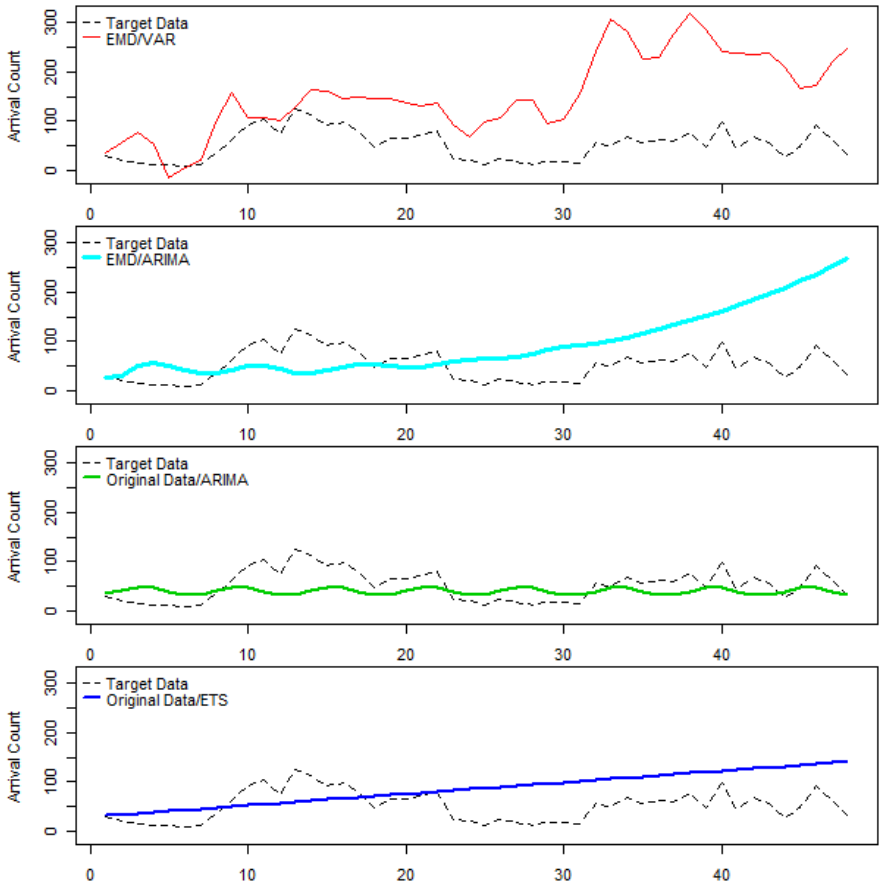

Figure 5.13: *48-hour forecasting results in case 1 where the look back range includes all days of week*

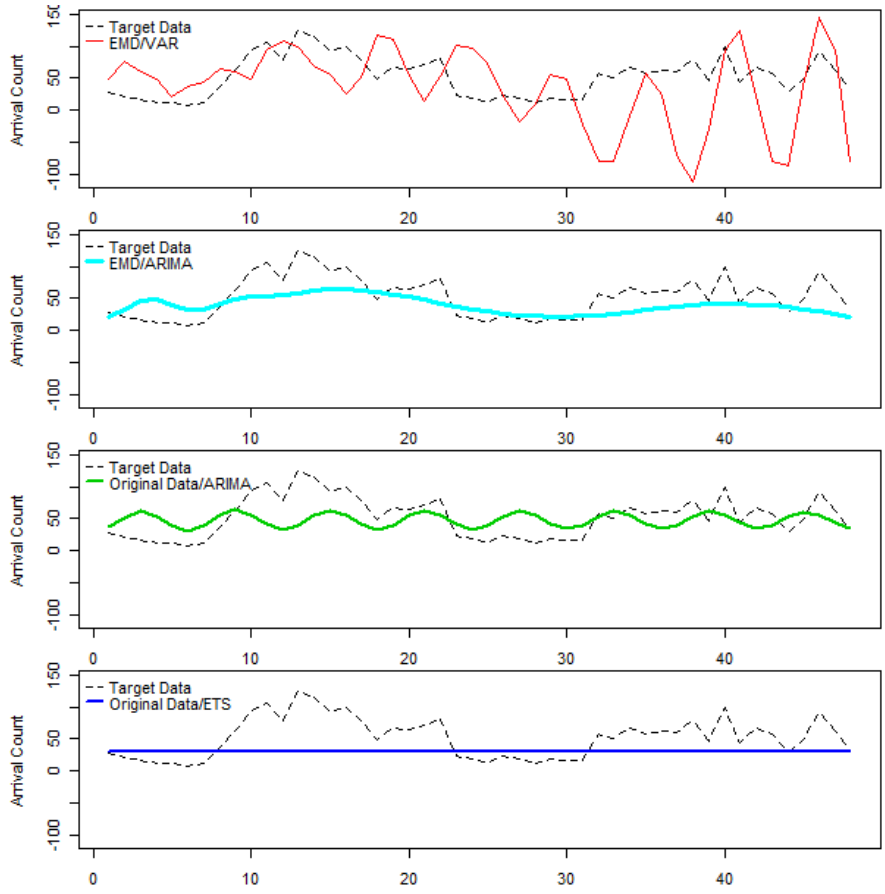

Figure 5.14: *48-hour forecasting results in case 2 where the look back range include only weekdays*

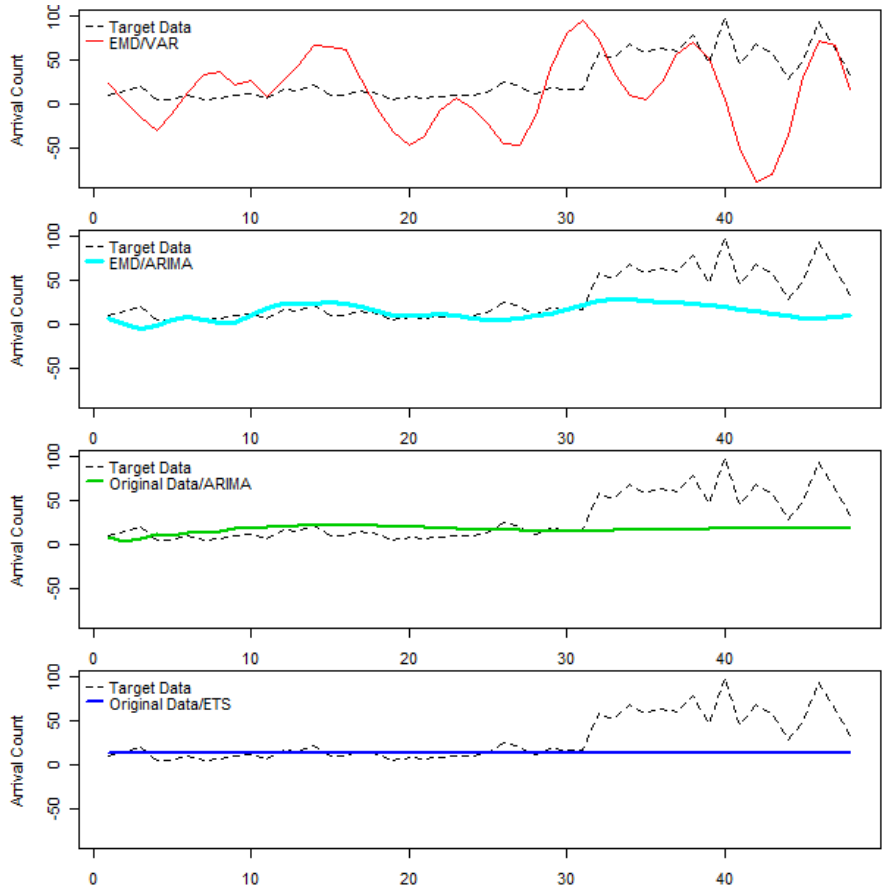

Figure 5.15: *48-hour forecasting results in case 3 where the look back range includes only weekends*

bination is outperformed by all other forecasting combinations. However, the EMD/VAR combination is better than the other forecasting combinations in preserving the differences in bursts from different periods of the forecast window

- The Original/ETS combination can only capture the trends of the cases. In the first case, the forecast trend is correct up to approximately 24 hours. Beyond 24 hours, this combination follow the linear trend upward and cannot reset for the next 24 hour forecast. On the other hand, in the second and third case, the Original/ETS combination is numerically accurate due to the steady nature of these cases. However, it misses all the burstiness of the hourly job arrival counts.
- The Original/ARIMA combination seems to capture an average burstiness and maintain this burstiness throughout the forecasting periods. Similar to Original/ETS, but better due to the capturing of an average burstiness, the Original/ARIMA combination also works well on the second and third case rather than the first case.
- The last combination, EMD/ARIMA, while outperforms EMD/VAR, also suffers the same setbacks as the ETS and Original/ARIMA combinations in case 1. However, in the second case, the EMD/ARIMA combination was able to capture both the burstiness as well as the overall up-down daily cycles of the original data. In the third case, with the lack of burstiness data, EMD/ARIMA does not perform as well as Original/ARIMA.
- Among the accuracy measurement methods, the measured sMAPEs of the forecasting approaches are quite similar to each other. This similarity is also maintained across the different forecast windows. Consequently, sMAPE does not offer a lot of insights into the comparison of these approaches. The RMSE method highlights the fact that all the forecasting approaches do well in the first 24 hours and deteriorate quickly beyond that point. However, a disadvantage of the RMSE method is that it magnifies the errors (due to being based on the euclidean distance calculation). Consequently, a forecast technique that captures the trend but does not follow the data closely will not do as well in RMSE as a technique that follow the averages of the data. This leaves the MASE method as a more reasonable solution, as the

forecasting approaches' performance are being compared against a default technique first, and only the resulting numerical rankings are compared against each other.

As the earlier experiments did not provide a definite answer for the performance comparison of the forecast approaches, we next perform a set of exhaustive experiments in order to have a more comprehensive understanding about how well each forecasting approach performs. In particular, these experiments will attempt to determine whether it is possible to identify a definitive forecasting approach that is proved to be superior to the others with regard to the Acxiom data set? If the answer for this question is "No", then next we will attempt to investigate more in-depth questions such as:

- If there is no particular "best" approach, then what are the circumstances under which a particular approach would work best?
- Is there a relationship between the days of week and the performance of the approaches?

These experiments are set up as followed:

1. The forecasting approaches are performed on the entire Acxiom dataset except for the initial look back range. The forecast window is set to contain four forecasting ranges: 3, 6, 12, and 24 hours. The look back range is set to be two weeks (336 hours). To better compare among the forecasting approaches, the forecasts are divided according to the days of week, from Monday to Sunday.
2. The error measurement is chosen to be MASE (Mean Average Scaled Error). For the collection of each day of week, MASE for individual predictions will be calculated. Next, depending on the MASE measurements, these predictions will be sorted into five groups (EMDVAR, EMDARIMA, ARIMA, ETS, and Undecided) based on the best forecasting approach as indicated by MASE. The first four groups are self explanatory, and the last group, Undecided, is for the cases where either the day to be predicted contains significant periods of down time or all the predictions are so incorrect that their inclusion will be considered

	Days of Week						
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
EMD/VAR	39%	42%	38%	34%	40%	26%	17%
EMD/ARIMA	23%	17%	25%	20%	2%	29%	13%
ARIMA	18%	27%	22%	39%	37%	31%	26%
ETS	5%	0%	0%	0%	0%	0%	0%
Undecided	15%	14%	15%	7%	7%	22%	44%

Table 5.4: *Percentages among the forecasting groups for each day of week*

outliers. The number of predictions in each group is counted, and the mean and standard deviation of the measured MASEs are also calculated.

3. In addition to the calculated MASEs, the graphs comparing each prediction with its intended target data are also drawn and compared visually.

The results of the experimental set are recorded in the tables of Appendix A. The tables show that on average, there are 60 forecasting observations for each day of week. While the predictions are sorted into five different groups based on their MASEs, there is a problem with the ETS approach. In particular, while it is not possible to identity the "best" approach, ETS can easily be considered having the worst performance. "Worst" here should be understood as an extremely unrealistic prediction despite the fact that the MASEs of ETS instances are among the best. An example of a "worst" performance of ETS with best calculated MASE is shown in Figure 5.16. In this example, the MASE of ETS is 1.40, which is better than 1.97 of EMD/VAR, 2.22 of EMD/ARIMA, and 2.45 of ARIMA. However, the ETS prediction is a straight positive horizontal line, which is completely incorrect. Consequently, when the sorting is performed, visual analysis is also taken into account in order to identify such pathological cases.

In table 5.4, the experimental results show that during the weekdays, the EMD-based approaches performances perform better than ARIMA. This is not true for Saturdays when ARIMA performs slightly better. A significant number of observations cannot be predicted correctly on Sundays (44% of the total Sundays).

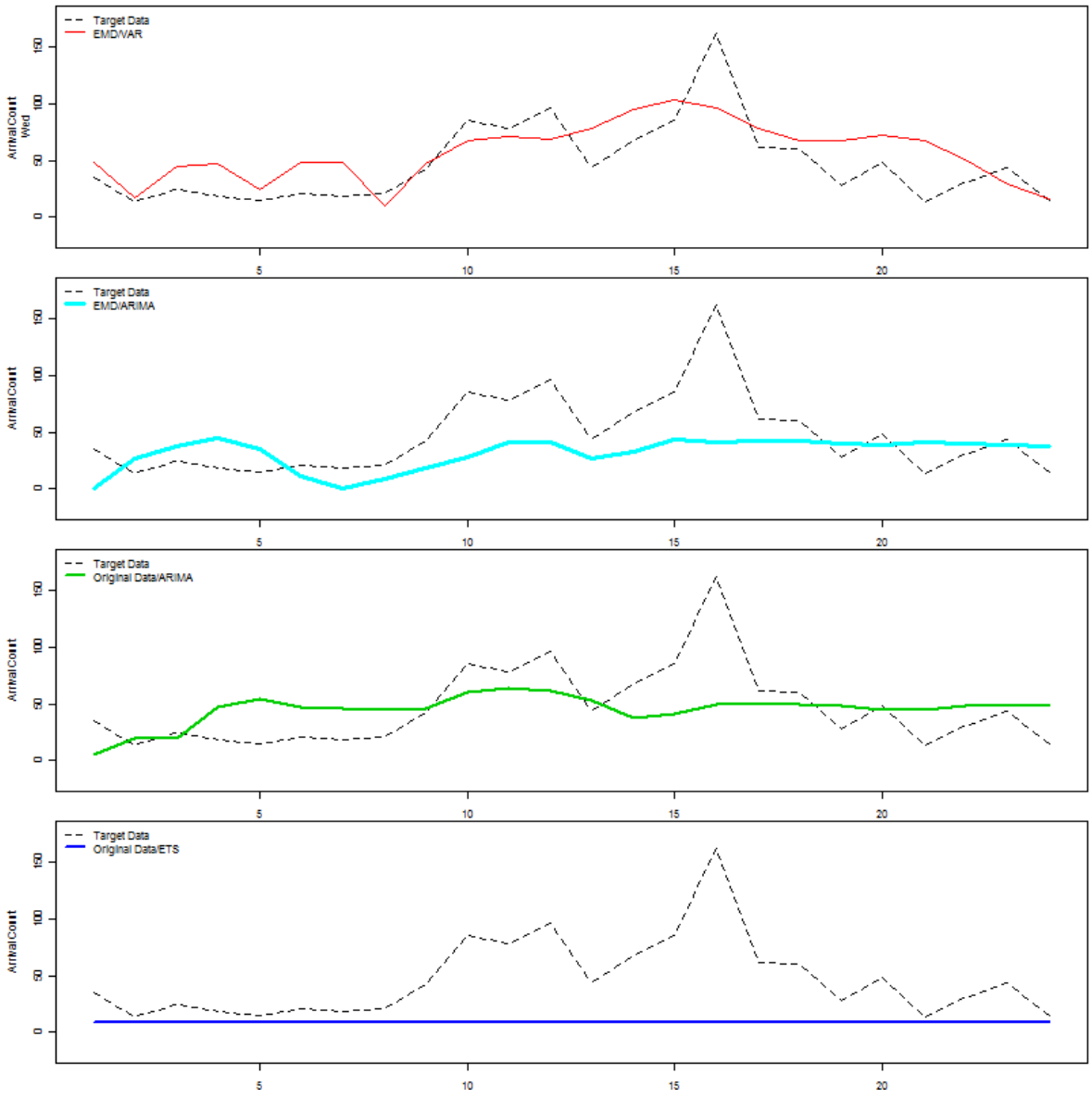

Figure 5.16: *Example of the case where the MASE of ETS is the best, yet it is not a realistic prediction.*

Another observation is made with respect to the recorded graphs. That is, in all the buckets, while the overall MASE for 24 hours can be different, ARIMA is better than EMD/VAR during the first 6-12 hours of the forecast window. These differences are reduced when compared with EMD/ARIMA. On the other hand, EMD/VAR is better than ARIMA in the prediction of the 12-24 hours of the window. Again, EMD/ARIMA is better than ARIMA, but worse than EMD/VAR. Particularly, there exists a number of cases where EMD-based can predict the changes in the trend (downward and then upward and vice versa) while ARIMA cannot.

An additional experiment is also performed, in which the forecast of a day of week is calculated as the average of the previous four same days of week. The MASE measurements for the 3, 6, 12, and 24 hours forecast window are recorded in table A.8 of the Appendix A. The results show that the 4-week averaged forecast is inferior compared to all other forecast approaches. This is because even for the same day of week, the arrival patterns do fluctuate through different weeks. These fluctuations cancel out each other in the average and reduce the burstiness of the calculated average.

With the observations from the above experiments, the following conclusions can be made:

- While there are correlations among the IMFs, it is not high enough to allow an effective and accurate application of VAR.
- Although there is visual evidence of seasonal components, standard statistical decomposition methods can not isolate them. This explains the inaccuracies of the ETS combination.
- Among the remaining three forecasting approaches, EMD/VAR, EMD/ARIMA, and ARIMA, while the EMD-based approaches can be better in some particular cases, the statistical evidence is not significant enough to declare it a winner.
- The performance of the forecast is affected by the day of week. The weekdays can be predicted more accurately, in all three approaches above, due to the expected daily patterns. On the other hand, the hourly arrival schedule of the weekends does not have a particular pattern, and thus is more difficult to predict.
- Having the IMF components helps to generate a forecast that correctly predict bursts in data

as well as the trend. On the other hand, the use of the components also affects the accuracy of EMD/VAR, when the target to be predicted is not of the same overall pattern as the look back data.

5.3 Conclusion

In conclusion, the work in this chapter has shown that it cannot be said for certain that the application of EMD as a preprocessing tool improves the output of the forecasts of Acxiom data. In the cases where EMD does improve the forecasting results, it is due to the empirical fact that EMD can decompose the input data into meaningful components, and the fact that while there exist seasonal components, these components cannot be extract by statistical decomposition methods. On the cases when either the target forecast data or the look back data are not following the perceived patterns, then EMD will fail to improve the results and might even be worse. This is true not only for EMD-based approaches but also for the approaches using traditional time series analysis techniques such as ARIMA or ETS. It could be said that even for seasonal data, EMD-based techniques and others are imperfect at best. Contrary to the preliminary study, the autocorrelation analysis shows that the hidden patterns within the original data weaken after one week and do not last beyond two weeks, thus invalidating the need to acquire more data.

Chapter 6

Workload Forecasting: Comparing Decomposition Approaches

In this Chapter, we attempt to compare the performance of EMD's sifting technique with another well known technique, wavelet-based decomposition. Since EMD has no mathematical foundation, it is difficult to compare EMD against wavelet decomposition. As a result, not many studies have been done to compare these two techniques with respect to the effect of their decomposition steps, and those who do only compare these techniques empirically. Flandrin and Goncalves highlight the comparability of EMD to wavelet in tasks such as denoising or detrending fractional Gaussian noises [122]. In particular, EMD can perform as a "wavelet-like" filter bank [123]. On the other hand, [122] states there is an important difference between EMD and wavelet's definition of the original signal. For EMD, a signal is a combinations of components with fast oscillations superimposed to components with slow oscillations. For Wavelet, a signal is a combinations of components with fast frequency detail superimposed to components with low-frequency approximation. This implies that the a signal component of EMD can be either harmonic or not while a signal component of wavelet is likely to have a regular frequency across the time range. Further work on the comparison between EMD and Wavelet shows that when the noise level is low, an EMD-based solution can perform better then wavelet, and vice versa [124]. In all of these studies, the authors have been careful to emphasize the fact that their studies are likely to be dependent on the experimental data and not general results. Consequently, the comparison between EMD and Wavelet in this section will also be empirical. That is, the wavelet-based decomposition will be utilized to decompose a sinusoidal function as well as to decompose and forecast the Acxiom data. The results from these experiments will be compared against the results of EMD-based decomposition.

6.1 Forecasting Sinusoidal Functions with Wavelet

In this section, we analyze and compare wavelet-based forecasting approaches with EMD-based ones in forecasting a set of functions with sinusoidal components. The sample functions are the same as the functions used in Chapter 5:

$f_1(x) = sin(x) + sin(2x) + sin(3x) + 0.5x$

$f_2(x) = sin(x) + sin(7x) + sin(20x) + 0.5x$

$f_3(x) = sin(x) + sin(4x) + sin(8x) + sin(12x) + sin(16x) + sin(24x) + sin(168x) + 0.5x$

For convenience, the forecasting results of EMD-based approaches (EMD/VAR and EMD/ARIMA) from Chapter 5 are presented again. The wavelet components of the functions f_1, f_2, and f_3 are used in the VAR and ARIMA forecasting approaches. However, VAR fails to work with the wavelet components for the functions. This is because the first few components produced by wavelet are essentially horizontal lines, $y = 0$, or contain segments that have all the y values converging toward 0. Consequently, when VAR performs the regression steps based on ordinary least squares, the coefficient estimation leads to the case of $0/0$, which invalidates the results. Only the forecasting results from wavelet/ARIMA are available to be recorded. These results are compared to the EMD-based forecast results in Figures 6.1, 6.2, and 6.3 and Table 6.1. The columns of Table 6.1 show the MAPE values for the different forecast window ranges: 3, 6, 12, and 24 hours. In the Table and Figures, the forecasting results from the EMD/VAR are significantly better than the EMD/ARIMA and wavelet/ARIMA. The results of EMD/ARIMA are also better than the wavelet/ARIMA for all ranges except the 12-hour and 24-hour predictions for f_3.

6.2 Acxiom Data Decomposition

In this section, we will look at how EMD and wavelet decompose the Acxiom data and highlight the differences between the resulting components of the two approaches. The monthly data of March 2006 is chosen for its two days with no arrival. Figure 6.4 demonstrates this comparison. There are two key differences between EMD and wavelet in decomposing Acxiom data. While there is no

	3 Hrs forecast window	6 Hrs forecast window	12 Hrs forecast window	24 Hrs forecast window
Forecast Approach				
$f_1(x) = sin(x) + sin(2x) + sin(3x) + 0.5x$				
EMD/VAR	0.71%	0.91%	1.42%	3.11%
EMD/ARIMA	0.72%	0.95%	1.81%	47.8%
Wav/ARIMA	12.31%	9.3%	13.8%	44.17%
$f_2(x) = sin(x) + sin(7x) + sin(20x) + 0.5x$				
EMD/VAR	11.72%	11.79%	13.28%	14.97%
EMD/ARIMA	13.94%	21.59%	32.19%	144.5%
Wav/ARIMA	141.4%	137.6%	101.4%	123.2%
$f_3(x) = sin(x) + sin(4x) + sin(8x) + sin(12x) + sin(16x) + sin(24x) + sin(168x) + 0.5x$				
EMD/VAR	2.61%	6.04%	12.64%	35.62%
EMD/ARIMA	49.37%	106.9%	223.7%	311.6%
Wav/ARIMA	183.8%	176.6%	170.1%	278.0%

Table 6.1: *Accuracy Measurement Comparisons between EMD-based and Wavelet-based Approaches using Mean Averaged Percentage Error*

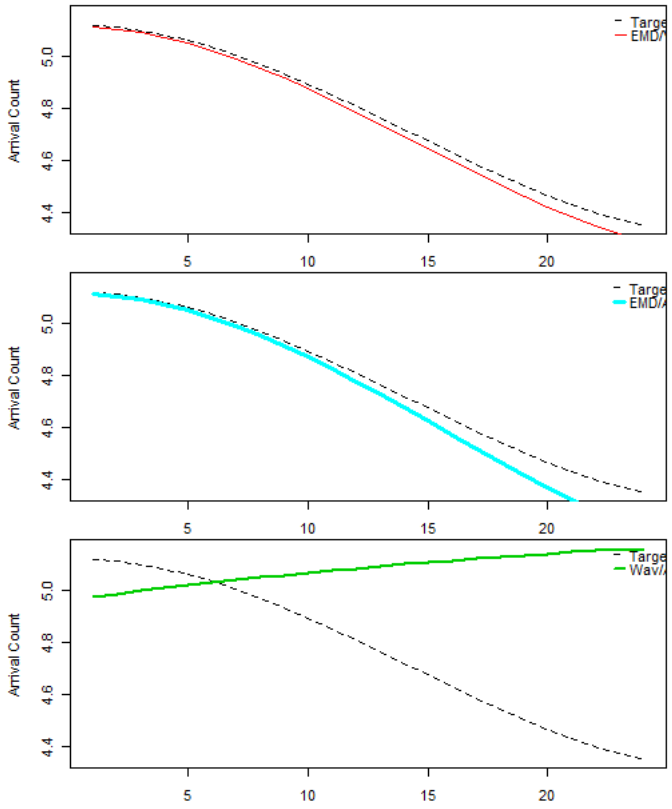

Figure 6.1: *Comparison of the forecasting results of EMD-based and wavelet-based approaches for function* f_1.

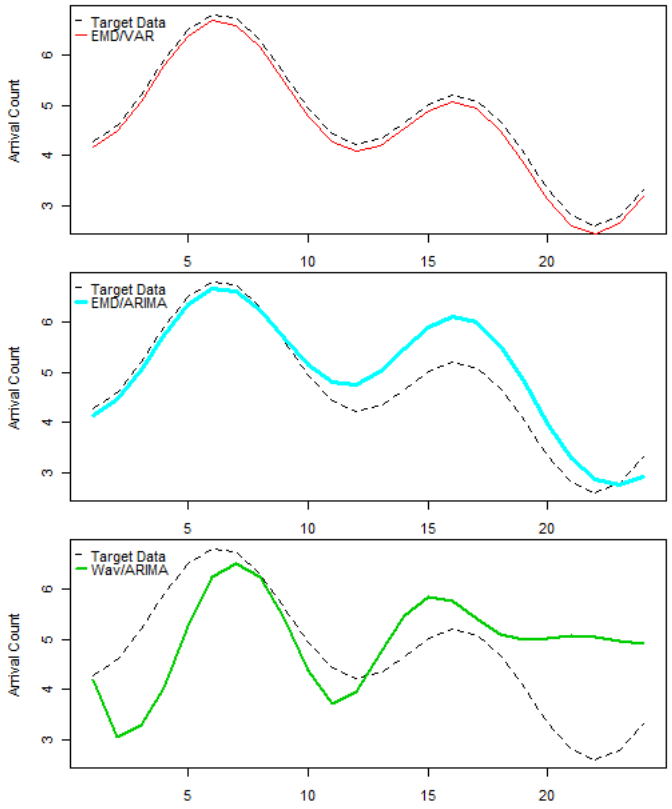

Figure 6.2: *Comparison of the forecasting results of EMD-based and wavelet-based approaches for function* f_2.

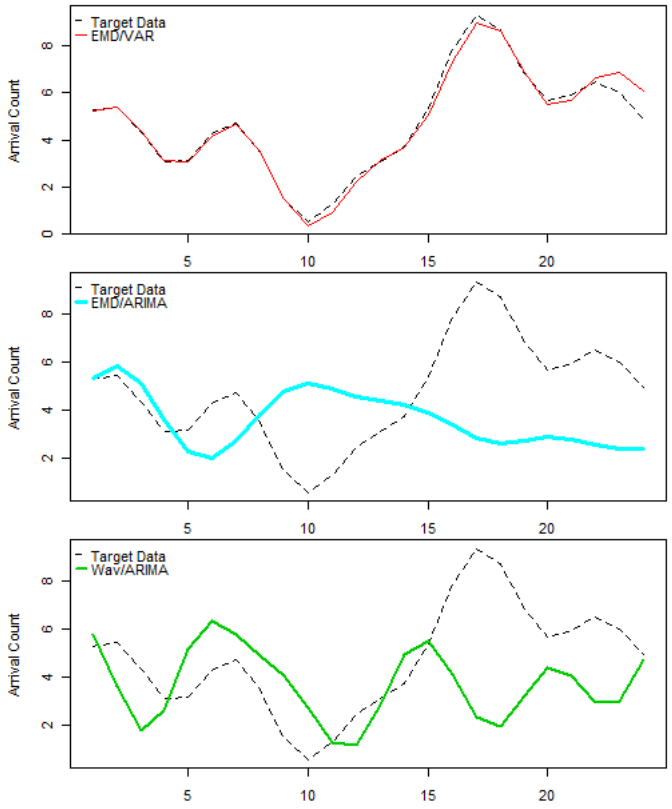

Figure 6.3: *Comparison of the forecasting results of EMD-based and wavelet-based approaches for function* f_3.

	EMD		wavelet	
Components	Mean	Standard Deviation	Mean	Standard Deviation
1	2.93	1.26	2.80	0.90
2	6.12	1.13	6.01	0.23
3	19.625	6.14	24.07	0.75
4	24.07	0.86	24.15	0.55
5	38.12	8.45	34.44	8.20
6	82.75	8.45	168	single point data
7	170	single point data	172	single point data
8	336	single point data	164.5	26.16

Table 6.2: *Comparison of the average distance between group of high value at fixed interval of the auto-correlation functions of IMF components and wavelet components (in hours)*

limitation to the number of decomposed components of wavelet, additional wavelet decomposition on the data beyond the number of IMF components already decomposed by EMD only yields a monotonically increasing trend. First, due to the lack of similar structural limitation conditions as IMFs, the wavelet's components are not as symmetric as the IMFs. This will affect the capability of wavelet components to be used in characterization, as the piecewise fitting method will not match up as well, and standard Fourier fitting will take longer. Second, wavelet components are disrupted at the indexes where the original data are zeros (the abnormal segment).

Table 6.2 compares the averaged periodic patterns of the IMF components and the wavelet components. Table 6.2 shows that the decomposed components of both wavelet and EMD are similar except for components 3 and 6. In order to have a more complete understanding on the components' patterns, the EMD and wavelet decompositions are performed on all the months of the Acxiom data. Overall, the results show that for all the months, the average number of components produced is the same for both techniques (8 components). Next, the averaged periodic pattern for each component of each month is calculated. These patterns are then averaged together to create the averaged patterns for the entire set of Acxiom data. The results, shown in Table 6.3, indicate that the perceived patterns of Acxiom data by both EMD and wavelet are similar, with exception of components 3 and 5. Without the internal knowledge of the Acxiom working schedule, it is not possible to decide which of these components reflect Acxiom's actual patterns better.

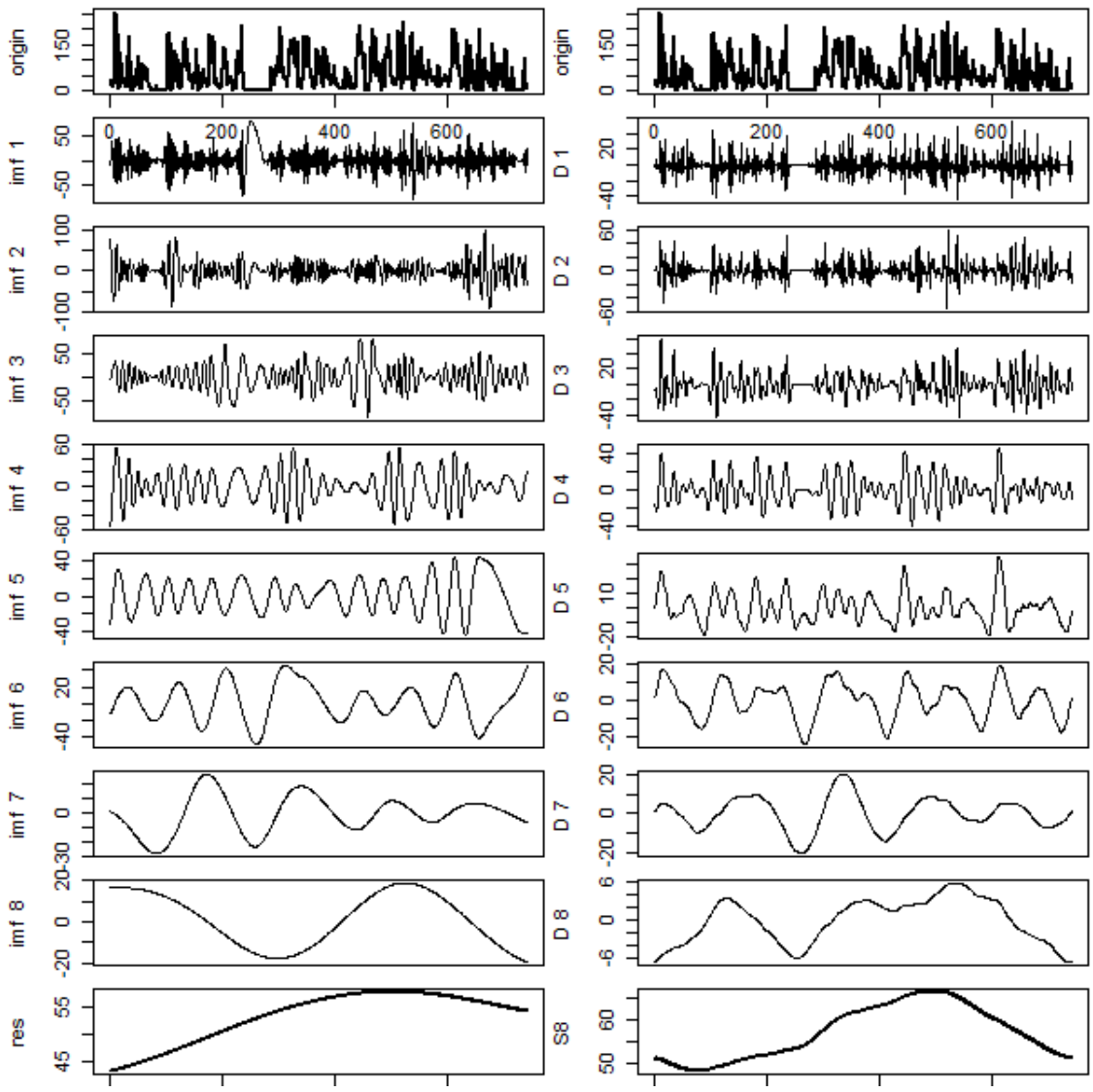

Figure 6.4: *Comparison the decompositions of EMD- and wavelet-based methods on Acxiom data. EMD decomposition is shown on the left. wavelet decomposition on the right*

	Components						
	1	2	3	4	5	6	7
EMD	3.25	6.66	15.64	24.01	46.53	92.72	171.4
wavelet	2.68	5.94	20.36	24.01	33.16	95.82	167.56

Table 6.3: *Averaged periodic patterns of the components produced by EMD and wavelet for the Acxiom data set (hours)*

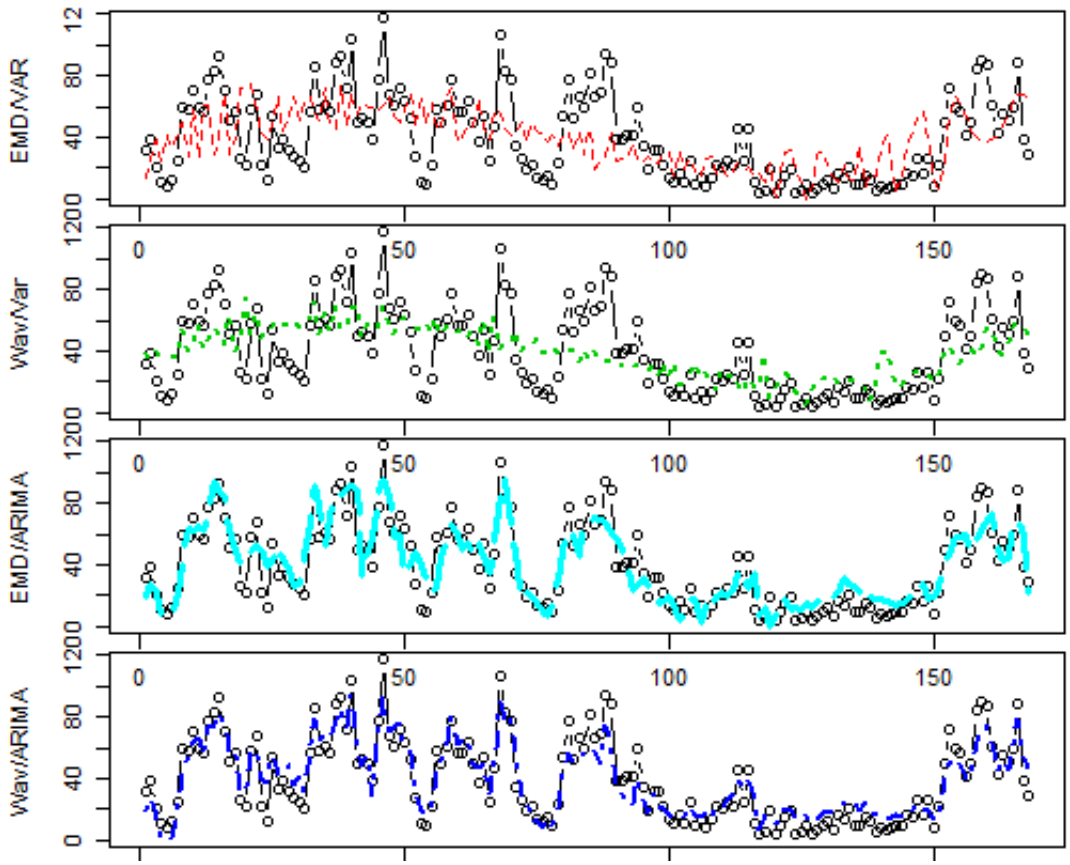

Figure 6.5: *Compare the fittings of EMD- and wavelet-based methods on Acxiom data on during the case that all days of week are used in the look back range*

6.3 Forecasting Acxiom Data

EMD and wavelet are applied to the same data and forecasting procedures as in section 5.1. Since we are not comparing with the original data, there are only four combinations: EMD/VAR, wavelet/VAR, wavelet/ARIMA, and EMD/ARIMA. First, we perform an initial experiment on random dates with three cases: forecasting using all days of the week, forecasting using only Monday to Friday, and forecasting using only Saturday and Sunday.

In the fitting data from Figures 6.5, 6.6, and 6.7, both EMD and wavelet perform on-par with each other with respect to the initial fit of the known data. The forecasting results are shown in Figures 6.8, 6.9, and 6.10. On the VAR combinations, the wavelet-based approach is less accurate than the EMD approach. In the case that the wavelet-based approach works, it predicts only the

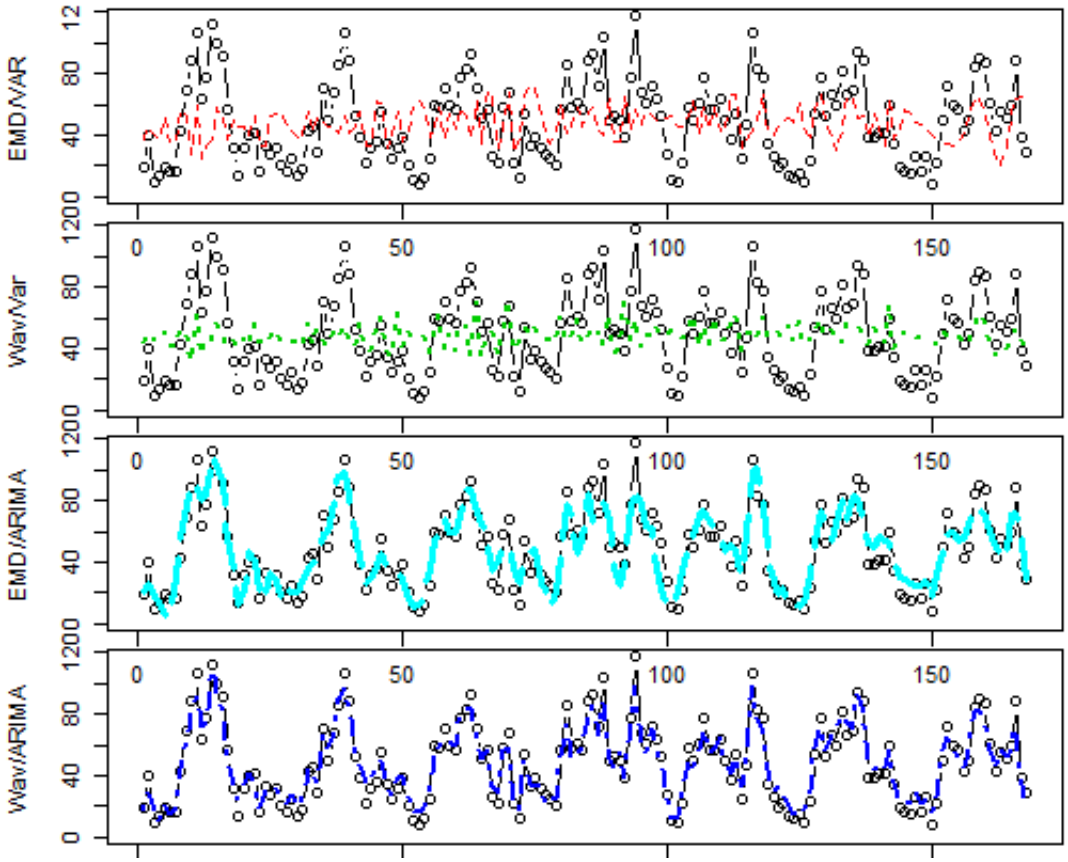

Figure 6.6: *Compare the fittings of EMD- and wavelet-based methods on Acxiom data on during the case that only weekdays are used in the look back range*

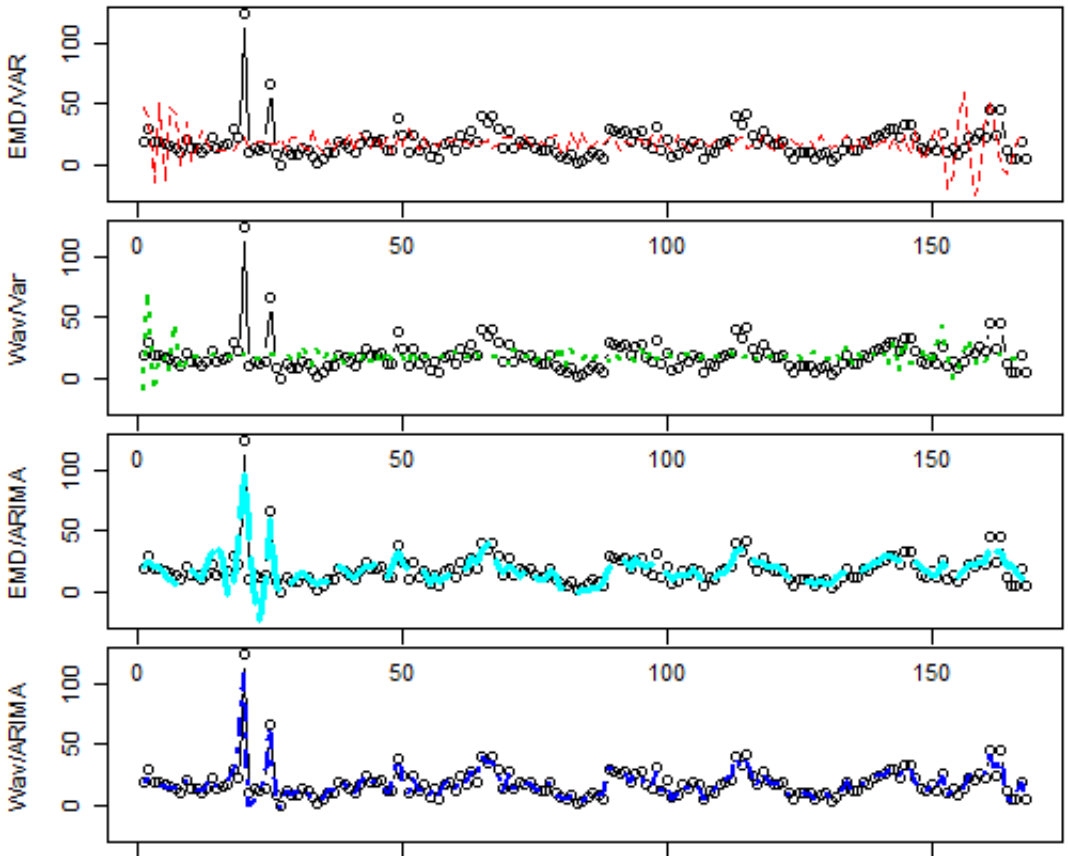

Figure 6.7: *Compare the fittings of EMD- and wavelet-based methods on Acxiom data on during the case that only weekends are used in the look back range*

upward or down trend of the data. On the ARIMA combinations, with the exception of the first case, the wavelet-based approach fails to correctly forecast on the remaining cases. On the first case, the wavelet-based approach does not imitate the burstiness of the data in the first twelve hours as well as the EMD-based approach.

Next, we next perform a more exhaustive set of comparison experiments. These experiments are set up similarly to the set of exhaustive experiment in the previous chapter. The combinations of decomposition techniques and forecasting techniques include EMD/VAR, wavelet/VAR, EMD/ARIMA, wavelet/ARIMA. In this experimental set, there are no groups, and the averages of each combination's forecasting observation for each days of week are recorded. The comparative graphs are also saved. The results of the experimental set are shown in the Table B.1 of Appendix B. The first observation from the results is that there is no MASE recorded for the combination of wavelet and VAR. This is due to the fact that there are many individual observations where the wavelet decomposition works, but the VAR will not produce a prediction. A common characteristics of these cases is that the look back ranges contain a period of down time that lasts more than a few hours. As there are many such cases for wavelet/VAR, removing these days from the Acxiom data set will lead to an unfair experiment. Therefore, it is reasonable to conclude that for the Acxiom data set, the combination of wavelet and VAR is not a feasible option. The collected MASEs show that the combination of wavelet/ARIMA is more accurate than both EMD/VAR and EMD/ARIMA in all days of week. However, with the experience from the ETS situations, the comparison graphs are also studied, and they show that while EMD/VAR does not oscillate as close to the target data as wavelet/ARIMA, EMD/VAR appears to capture the instantaneous changes in trend and the daily burstiness better than wavelet/ARIMA.

6.4 Conclusions

In conclusion, for the case of Acxiom data, EMD and wavelet are comparable with each other in the decomposition process. However, the IMF components are smoother and continuous as compared to the jagged curves of the wavelet components. The two conditions of IMF (about zero crossings

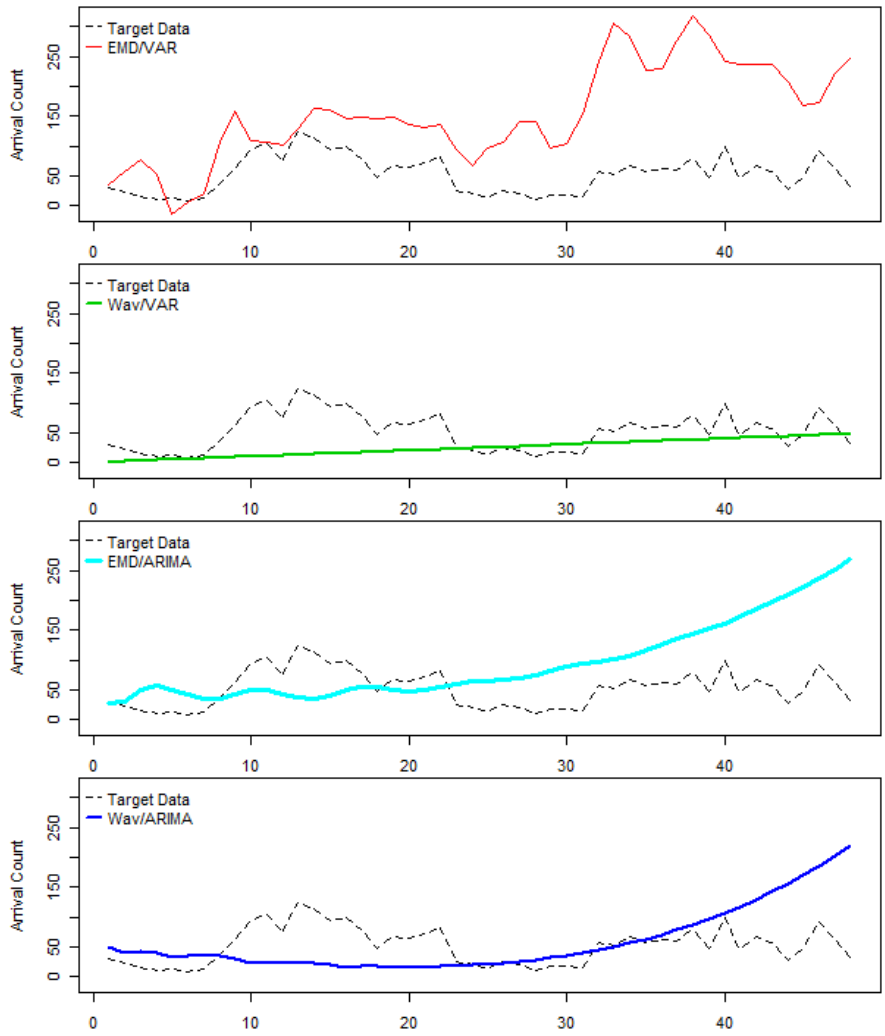

Figure 6.8: *Comparing between wavelet-based and EMD-based 48-hour forecasting results in the case where the look back data contains all days of week*

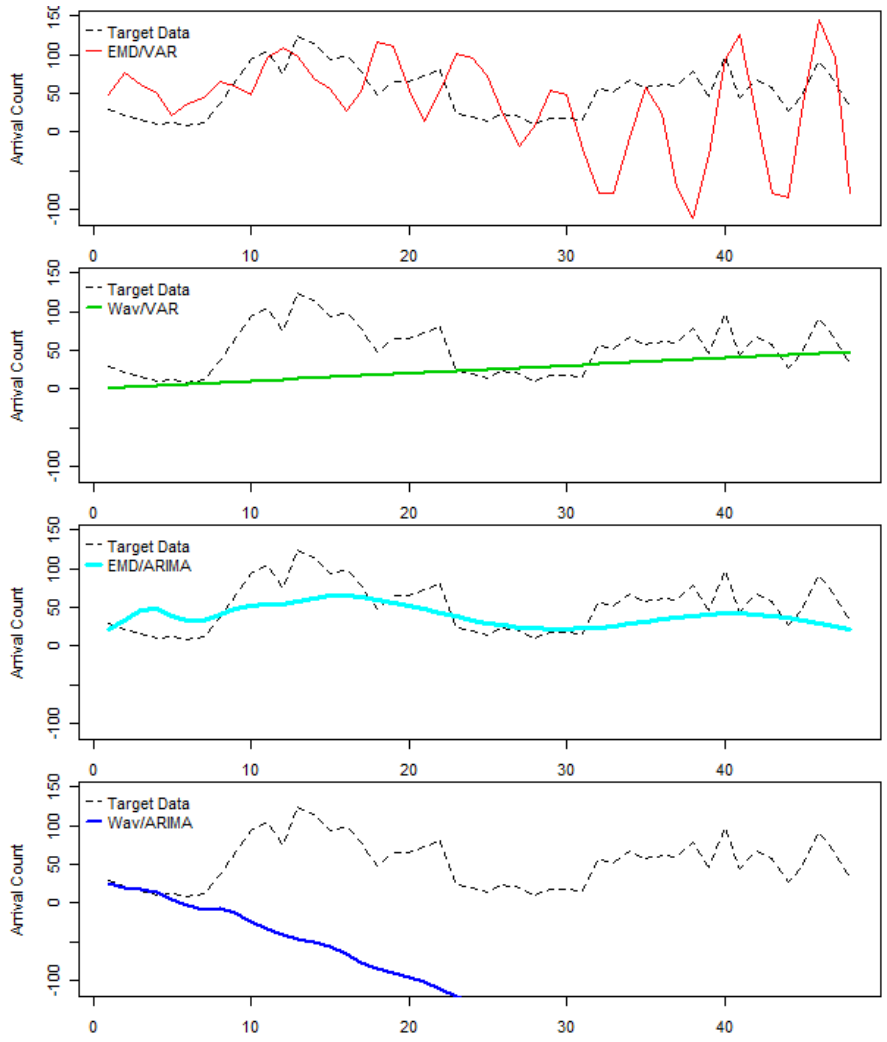

Figure 6.9: *Comparing between wavelet-based and EMD-based 48-hour forecasting results in the case where the look back data contains only weekdays*

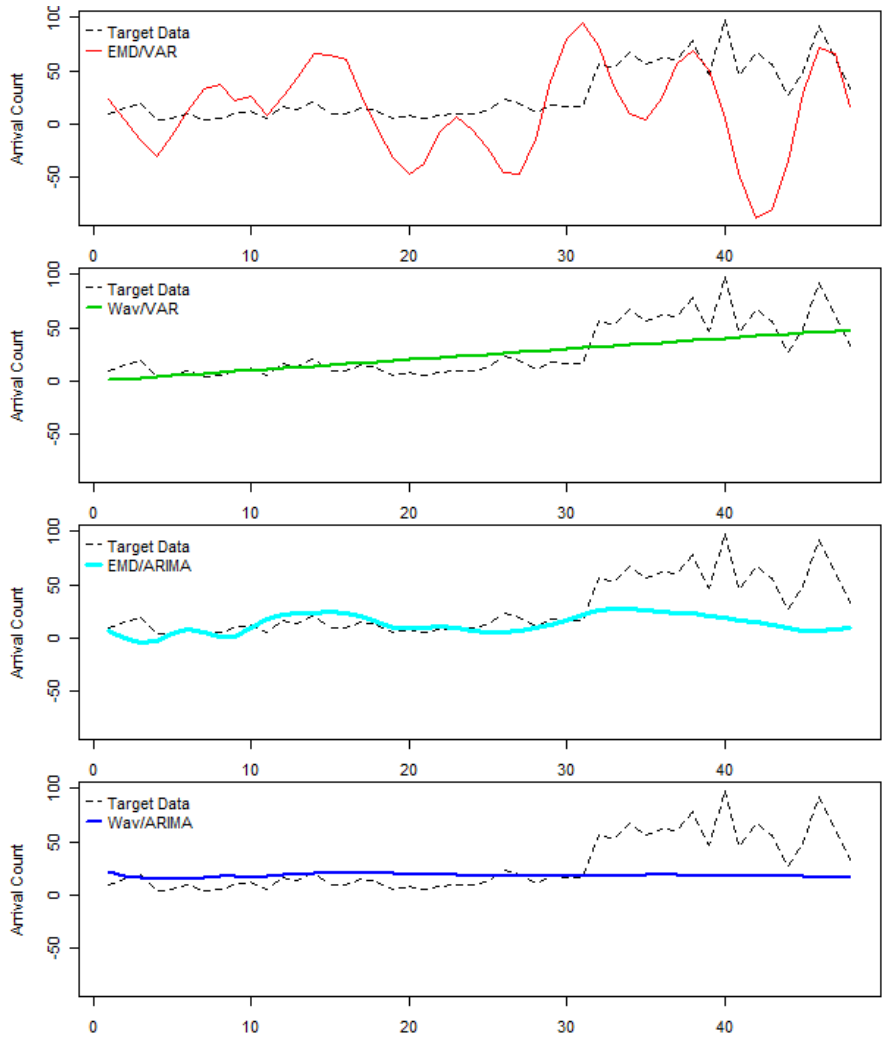

Figure 6.10: *Comparing between wavelet-based and EMD-based 48-hour forecasting results in the case where the look back data contains only weekend*

and mean value of the upper and lower envelopes) also help IMFs to have a more symmetric structure than the wavelet components. In the case of downtime periods, wavelet retains the empty intervals. These characteristics make it more difficult to fit the component for characterization for wavelet than EMD.

In terms of forecasting approaches, wavelet only works with ARIMA and dose not work well with VAR. The wavelet/ARIMA approach has the better MASE measurement than EMD/VAR and EMD/ARIMA. Still, this is not enough to declare wavelet a winner, due to the fact that the graphical evidence suggests EMD is better in capturing the instantaneous changes in the daily trend and the burstiness of the arrival data. Given the differences in deciding the better decomposition approaches from the calculated errors and the graphical observations, there is not a conclusive answer for the comparison, and a better error measurement method is needed.

Chapter 7

Conclusions

In this chapter, we first summarize the contents of this dissertation. Next, the findings and contributions of the research are mentioned. Finally, we discuss the future research.

7.1 Summary

This dissertation focuses on the application of the Empirical Mode Decomposition technique (EMD) to the arrival data of an enterprise cluster. First, the uses of Empirical Mode Decomposition (EMD) as well as traditional approaches relying on hierarchical characterization and hyper-exponential distribution (HED) as methods of arrival time characterize in performance analysis are demonstrated. The sifting procedure of the EMD was applied to the original arrival streams to produce a set IMF components, which in turn carry patterns that can be used for characterization. To characterize these IMFs, a standard Fourier fitting and a novel piecewise sine fitting was chosen. Due to the high accuracy as well as a significant improvement in fitting time, the piecewise sine fitting were chosen. The resulting synthetic job arrivals generated from a set of these fitted functions and the fitted residue follow a distribution that is highly similar to the original arrival stream's. The EMD-based arrivals capture the characteristics of the original arrival stream better as compared to the synthetic arrivals produced by the hyper-exponential distribution (HED) approach. However, the EMD-based approach method requires a nontrivial amount of manual calibration, such as choosing the most fitted combination of IMFs, in order to achieve optimal results. In addition, the modification of the piecewise-fitted sinusoidal functions for the purpose of capacity planning is both difficult and redundant.

Second, the application of EMD as a data preprocessing tool for forecasting purposes is examined. A preliminary study was performed to confirm the potential of EMD to be used in forecasting. The observations of this study have shown promises in the utilization of EMD. Again,

manual calibrations are needed despite the evidence of EMD's capability to decompose sinusoidal components from the data stream. The separation of the original time plot into the intrinsic mode functions creates an improved set of data for the prediction purposes. With a simple weighted forecast algorithm, it is observed that the accuracy of the prediction can be increased by increasing the range of data decomposition.

An analysis of the autocorrelation functions of the original data as well as the IMF components shows that there are hidden patterns within the original data that weaken after one week and do not last beyond two weeks. This is in contrast to the preliminary study and invalidates the need to use more data in developing the prediction. This is the basis for the next set of experiments to compare the performances of several forecasting approaches (EMD/VAR, EMD/ARIMA, ARIMA, ETS) for the Acxiom data. The results of these experiments show that there is no conclusive evidence that using EMD as a preprocessing tool will significantly improve the forecasting results. While the EMD-based approaches are typically better, the percentages are not statistically significant. In addition, the accuracy of the EMD-based approaches is reduced when trying to predict days that have no clear pattern, such as weekend days. Combining this with the results from the preliminary study, it is possible that while the EMD process does extract IMFs that resonate with the perceived patterns of Acxiom data, either these IMFs are not entirely accurate or there might be noise that negatively affects the prediction outcomes.

In comparing EMD against wavelet, contradicting conclusions are observed. While EMD outperforms wavelet in decomposing standard functions with sinusoidal components, the MASE measurements show that the combination of wavelet/ARIMA outperforms EMD/VAR and EMD/ARIMA. On the other hand, wavelet is shown to not work well with VAR, and the graphical observations indicate that EMD/VAR performs better in capturing the instantaneous changes in trend.

7.2 Contributions

A detailed study on the application of EMD to the characterization and forecasting of arrival times in an enterprise cluster has been performed in this research. The study is comprehensive and is

streamlined with a large body of R scripts that have been modularized and can be reused for the analysis of other data type. The contributions of this research are as follows:

- The usefulness of EMD in characterization is proven. A piecewise sine fitting approach was proposed and proven to be effective for characterization purposes. The synthetic data generated from this fitting matches closely with the original data.
- The comprehensive preliminary study has demonstrated EMD's capability in isolating patterns of a function. In comparison to wavelet, EMD can not only denoise and detrend but also extract the possible patterns, if there are any.
- In term of forecasting, an extensive set of experiments has been performed on the entire range of Acxiom data, with different decomposition techniques as well as forecasting techniques are analyzed. While EMD has not been proven to be a definitive choice for forecasting, the experiments offer a better understanding on how well each forecasting technique performs with different arrival conditions.
- A set of procedures, expressed as R scripts, has been developed that helps with the initial analysis and subsequent forecasts of the enterprise data. These procedures are modularized and can easily be modified to be used for other data types. While R cannot produce an enclosed executable program with a GUI as with Matlab or Java, it is open source and user-friendly with a low learning curve.

7.3 Future Work

The ongoing research of this topic will continue to explore the applicability of EMD in the field of forecasting. An important aspect of this is the development of an appropriate error measurement technique that will evaluate the accuracy of EMD-based forecasting techniques. Additional forecast techniques will be studied, particularly the neural-network approaches. The use of neural networks will allow the analysis of complicated relationships among the data set as well as a more dynamic forecasting mechanism and thus can utilize the EMD technique better and create an opportunity for more patterns to be identified.

Different types of datasets will also be investigated. The data used in this research are enterprise data and therefore follow a regular schedule. What happens if the data from an environment with less seasonal schedules, such as those of an academic institution, are analyzed?

In addition, one of EMD's biggest disadvantage is the execution time. It is necessary to parallelize the sifting procedure, so that very large data sets can be decomposed in reasonable time for analytical purposes.

Bibliography

[1] E. O. Joslin, "Application benchmarks: The key to meaningful computer evaluation," in *Proceedings of the ACM Annual Conference*, 1965.

[2] N. E. Huang, Z. Shen, S. R. Long, M. L. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A*, vol. 454, 1998.

[3] L. Ngo, H. Bui, A. Apon, N. Hamm, L. Dowdy, D. Hoffman, and D. Brewer, "Application of empirical mode decomposition to the arrival time characterization of a parallel batch system using system logs," in *Proceedings of the 2009 International Conference on Modeling, Simulation, and Visualization Methods*, 2009.

[4] L. Ngo, A. Apon, and D. Hoffman, "A forecasting capability study of empirical mode decomposition for the arrival time of a parallel batch system," in *Proceedings of the Seventh International Conference on Information Technology: New Generations*, 2010.

[5] H. Lucas, "Synthetic program specifications for performance evaluation," in *Proceedings of the ACM Annual Conference*, 1972.

[6] K. Sreenivasan and A. J. Kleinman, "On the construction of a representative synthetic workload," *Communications of the ACM*, vol. 17, 1974.

[7] A. K. Agrawala, J. M. Mohr, and R. M. Bryant, "An approach to the workload characterization problem," *IEEE Computer*, vol. 9, 1976.

[8] D. Ferrari, *Computer Systems Performance Evaluation*. Prentice Hall, 1978.

[9] ——, "On the foundation of artificial workload design," *ACM SIGMETRICS Performance Evaluation Review*, vol. 12, 1984.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 3, 1999.

[11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceeding of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[12] M. Calzarossa and G. Serazzi, "Workload characterization: A survey," *Proceedings of the IEEE*, vol. 81, 1993.

[13] J. M. Mohr, "The time varying nature of computer workloads," in *Proceedings of the 1979 CMG Conference*, 1979.

[14] M. Calzarossa and G. Serazzi, "A characterization of the variation in time of workload arrival patterns," *IEEE Transactions on Computers*, vol. c34, 1985.

[15] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *Proceedings of the ACM SIGComm*, 1993.

[16] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, 1995.

[17] C. Ruemmler and J. Wilkes, "Unix disk access patterns," in *Proceedings of Winter USENIX*, 1993.

[18] G. R. Ganger, "Generating representative synthetic traces: An unsolved problem," in *Proceedings of the 1995 CMG Conference*, 1995.

[19] ——, "System-oriented evaluation of i/o subsystem performance," Ph.D. dissertation, University of Michigan at Ann Arbor, 1995.

[20] M. Gomez and V. Santonja, "Self-similarity in I/O workload: Analysis and modeling," in *Proceedings of the 1998 Workload Characterization: Methodology and Case Studies*, 1998.

[21] ——, "Analysis of self-similarity in I/O workload using structural modeling," in *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 1999.

[22] M. Wang, A. Ailamaki, and C. Faloutsos, "Capturing the spatio-temporal behavior of real traffic data," *Performance Evaluation*, vol. 49, 2002.

[23] C. E. Shannon and W. Weaver, *Mathematical Theory of Communication*. University of Illinois Press, 1963.

[24] M. Wang, T. Madhyastha, N. H. Chan, S. Papadimitrou, and C. Faloutsos, "Data mining meets performance evaluation: Fast algorithm for modeling bursty traffic," in *Proceedings of the 2002 International Conference on Data Engineering*, 2002.

[25] K. C. Sevcik, "Characterization of parallelism in applications and their use in scheduling," *Performance Evaluation Review*, vol. 17, 1989.

[26] M. Calzarossa, M. Haring, G. Kotsis, A. Merlo, and D. Tessera, "A hierarchical approach to workload characterization for parallel systems," in *Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking*, 1995.

[27] A. Downey, "A parallel workload model and its implications for processor allocation," in *Proceedings of the 6th IEEE International Symposium on High Performance Distributed Computing*, 1997.

[28] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riordan, "Modeling of workload in MPPs," in *Proceedings of the Job Scheduling Strategies for Parallel Process*, 1997.

[29] R. Schassberger, "Insensitivity of steady-state distributions of generalized semi-markov processes I," *The Annals of Probability*, vol. 5, 1977.

[30] ——, "Insensitivity of steady-state distributions of generalized semi-markov processes II," *The Annals of Probability*, vol. 6, 1978.

[31] ——, "Insensitivity of steady-state distributions of generalized semi-markov processes with speeds," *Advanced in Applied Probability*, vol. 10, 1978.

[32] H. Franke, J. Jann, J. E. Moreira, and P. Pattnaik, "An evaluation of parallel job scheduling for ASCI Blue-pacific," in *Proceedings of 1999 Supercomputing*, 1999.

[33] A. Downey and D. G. Feitelson, "The elusive goal of workload characterization," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, 1999.

[34] A. Snavely, L. Carrington, N. Wolter, J. Labarta, R. Badia, and A. Purkayastha, "A framework for performance modeling and prediction," in *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, 2002.

[35] Y. Li, T. Li, T. Kahveci, and J. Fortes, "Workload characterization of bioinformatics applications," in *Proceedings of the 2005 IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005.

[36] N. Azeemi, A. Sultan, and A. A. Muhammad, "Parameterized characterization of bioinformatics workload on SIMD architecture," in *Proceedings of the International Conference on Information and Automation*, 2006.

[37] R. Cheveresan, M. Ramsay, C. Feucht, and I. Sharapov, "Characteristics of workloads used in high performance and technical computing," in *Proceedings of the 21st ACM International Conference on Supercomputing*, 2007.

[38] L. Cherkasova, "Analysis of enterprise media server workloads: Access patterns, locality, content evolution, and rates of change," *IEEE/ACM Transactions on Networking*, vol. 12, 2004.

[39] H. Li, D. Groep, and L. Wolters, *Workload Characteristics of a Multi-cluster Supercomputer*. Springer Verlag, 2004, pp. 176–193.

[40] B. Lu, "Integrated capacity planning environment," Ph.D. dissertation, University of Arkansas, 2008.

[41] M. S. Squillante, D. D. Yao, and L. Zhang, "The impact of job arrival patterns on parallel scheduling," *SIGMETRICS Performance Evaluation Review*, vol. 26, 1999.

[42] D. G. Feitelson, "Packing schemes for gang scheduling," in *Proceeding of the Workshop on Job Scheduling Strategies for Parallel Processing (IPPS'96)*, 1996.

[43] T. L. Lo, "The evolution of workload management in data processing industry: A survey," in *Proceedings of the 1986 ACM Fall Joint Computer Conference*, 1986.

[44] D. D. Agro and S. Preston, "The linear projection model: An event driven forecasting mode," in *Proceedings of the 1984 CMG Conference*, 1984.

[45] T. M. O'Donovan, *Short Term Forecasting: An Introduction to the Box-Jenkins Approach*. Springer, 1985.

[46] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing*. Springer, 2008.

[47] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

[48] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. Prentice Hall, 1978.

[49] E. S. Gardner and E. McKenzie, "Forecasting trends in time series," *Management Science*, vol. 31, 1985.

[50] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management Science*, vol. 6, 1960.

[51] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of Forecasting*, vol. 18, 2002.

[52] P. M. Lee, *Bayesian Statistics: An Introduction*. Oxford University Press, 1989.

[53] P. Harrison and C. Stevens, "Bayesian forecasting," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 38, 1976.

[54] A. S. Goldberger, *Topics in Regression Analysis*. The MacMillan Company, New York, 1968.

[55] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, 1943.

[56] T. Kohonen, *Associative Memory: A System-Theoretical Approach*. Springer-Verlag, 1977.

[57] M. S. Seidenberg and J. L. McClelland, "Encoding sequential structure in simple recurrent networks," in *Advances in Neural Information Processing System I*, D. Tourezky, Ed. New York: Morgan Kaufman, 1989.

[58] D. R. Rumelhart, G. E. Hilton, and R. J. William, "Learning presentation by back-propagating errors," *Nature*, vol. 323, 1986.

[59] S. Grossberg, "Some networks that can learn, remember, and reproduce any number of complcated space-time patterns II," *Studies in Applied Mathematics*, vol. 49, 1970.

[60] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, 1991.

[61] D. Lowe and A. R. Webb, "Time series prediction by adaptive networks: A dynamic system perspective," *IEE Proceedings F: Radar and Signal Processing*, vol. 138, 1991.

[62] S. S. Rao, S. Sethuraman, and V. Ramamurti, "A recurrent neural network for nonlinear time series prediction - a comparative study," in *Proceedings of the 1992 IEEE Workshop on Neural Networks for Signal Processing*, 1992.

[63] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural network and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 3, 1994.

[64] R. Drossu and Z. Obradovic, "Rapid design of neural networks for time series prediction," *IEEE Computational Science & Engineering*, vol. 3, 1996.

[65] N. Toda and S. Usui, "A numerical approach for estimating higher order spectra using neural network autoregressive model," in *Proceedings of the 1995 IEEE Workshop on Neural Networks for Signal Processing*, 1995.

[66] T. Matsumoto, H. Hamagishi, and Y. Chonan, "A hierarchical Bayes approach to nonlinear time series prediction with neural nets," in *Proceedings of the 1997 International Conference on Neural Networks*, 1997.

[67] S. Chen, "Nonlinear time series modeling and prediction using gaussian RBF networks with enhanced clustering and RLS learning," *Electronic Letters*, vol. 31, 1995.

[68] X. Lu and C. Chen, "A new hybrid recurrent neural network," in *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, 1999.

[69] A. A. Khalaf and K. Nakayama, "A learning algorithm for a hybrid nonlinear predictor applied to noisy nonlinear time series," in *Proceedings of the 1999 International Joint Conference on Neural Networks*, 1999.

[70] E. Gomez-Ramirez, A. Poznyak, A. Gonzalez-Yunes, and M. Avila-Alvarez, "Adaptive architecture of polynomial artificial neural network to forecast nonlinear time series," in *Proceedings of the 1999 Congress on Evolutionary Computation*, 1999.

[71] S. Hong, S. Oh, M. Kim, and J. Lee, "Nonlinear time series modeling and prediction using gaussian RBF network with structure optimisation," *Electronics Letters*, vol. 37, 2001.

[72] M. Lehtokangas, J. Saarinen, P. Huuhtanen, and K. Kaski, "Neural network optimization tool based on predictive MDL principle for time series prediction," in *Proceedings of the 1993 IEEE International Conference on Tools with AI*, 1993.

[73] A. Kalos, "Automated heuristic growing of neural networks for nonlinear time series models," in *Proceedings of the 2005 International Joint Conference on Neural Networks*, 2005.

[74] X. Jianyuan, T. Yun, and L. Xin, "A novel forecasting model of contaminated insulator flashover voltage," in *Proceedings of the 2009 IEEE Conference on Industrial Electronics and Applications*, 2009.

[75] S. Soltani, S. Canu, and D. Boichu, "Time series prediction and the wavelet transform," in *Proceedings of the 1998 International Workshop on Advanced Black Box Techniques for Nonlinear Modeling: Theory and Applications*, 1998.

[76] T. Shin and I. Han, "A hybrid system using mutiple cyclic decomposition methods and neural network techniques for point forecast decision making," in *Proceedings of the 2000 Hawaii International Conference on System Sciences*, 2000.

[77] P. Yu, A. Goldenberg, and Z. Bi, "Time series forecasting using wavelets with predictor-corrector boundary treament," in *Proceedings of the Temporal Data Mining Workshop at the 2001 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.

[78] H. Xuefeng and X. De, "Time series prediction based on non-parametric regression and wavelet-fractal," in *Proceedings of the 2004 International Conference on Signal Processing*, 2004.

[79] G. Mao, "Real-time network traffic prediction based on a multiscale decomposition," in *Lecture Notes in Computer Science*. Singler Berlin/Heidelberg, 2005.

[80] P. Goupillaud, A. Grossmann, and J. Morlet, "Cycle-octave and related transforms in seismic signal analysis," *Geoexploration*, vol. 23, 1984.

[81] R. C. Sharpley and V. Vatchev, "Analysis of the intrinsic mode function," *Constructive Approximation*, vol. 24, 2006.

[82] S. Kizhner, K. Blank, T. Flatley, N. E. Huang, D. Petrick, and P. Hestnes, "On certain theoretical development underlying the Hilbert-Huang transform," in *Proceedings of the 2006 IEEE Aerospace Conference*, 2006.

[83] E. Delechelle, J. Lemoine, and O. Niang, "Empirical mode decomposition: An analytical approach for sifting process," *IEEE Signal Processing Letters*, vol. 12, 2005.

[84] T. Zheng, L. Yang, and D. Huang, "An alternative analytical definition to the mean envelope for empirical mode decomposition," in *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition*, 2007.

[85] G. G. S. Pegram, M. C. Peel, and T. A. McMahon, "Empirical mode decomposition using rational splines: An application to rainfall time series," *Proceedings of the Royal Society A*, vol. 464, 2008.

[86] Y. Kopsinis, "Investigation and performance enhancement of the empirical mode decomposition method based on a heuristic search optimization approach," *IEEE Transactions on Signal Processing*, vol. 56, 2008.

[87] K. T. Coughlin and K. K. Tung, "11-year solar cycle in the stratosphere extracted by the empirical mode decomposition method," *Advances in Space Research*, vol. 34, 2004.

[88] G. Rilling, P. Flandrin, and P. Golcalves, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2003.

[89] M. C. Peel, G. E. Amirthanathan, G. G. S. Pegram, T. A. McMahon, and F. H. S. Chiew, "Issues with the application of empirical mode decomposition analysis," in *Proceedings of the International Congress on Modeling and Simulation*, 2005.

[90] Z. Zhidong and W. Yang, "A new method for processing end effect in empirical mode decomposition," in *Proceedings of the 2007 International Conference on Communication, Circuits, and Systems*, 2007.

[91] H. Liang, Q. Lin, and J. D. Z. Chen, "Application of the empirical mode decomposition to the analysis of esophageal manometric data in gastroesophageal reflux disease," *IEEE Transactions on Biomedical Engineering*, vol. 52, 2005.

[92] D. Rouvre, D. Kouame, F. Tranquart, and L. Pourcelot, "Empirical mode decomposition (EMD) for multi-gate, multi-transducer ultrasound doppler fetal heart monitoring," in *Proceedings of the 2005 IEEE International Symposium on Signal Processing and Information Technology*, 2005.

[93] G. Souretis, D. Mandic, M. Grisseli, T. Tanaka, and M. V. Hulle, "Blood volume analysis with empirical mode decomposition," in *Proceedings of the 2007 International Conference on Digital Signal Processing*, 2007.

[94] A. Karagiannis and P. Constantinou, "Noise components identification in biomedical signals based on empirical mode decomposition," in *Proceedings of the 9th International Conference on Information Technology and Applications in Biomedicine*, 2009.

[95] D. Chen, D. Li, M. Xiong, H. Bao, and X. Li, "Gpgpu-aided ensemble empirical mode decomposition for eeg analysis during anesthesia," *IEEE Transactions on Information Technology in Biomedicine)*, vol. 14, 2010.

[96] L. Liang and Z. Ping, "An edge detection algorithm of image based on empirical mode decomposition," in *Proceedings of the 2008 International Symposium on Intelligent Information Technology Application*, 2008.

[97] G. Guangtao, S. Enfang, L. Zhuofu, and Z. Beibei, "Underwater acoustic feature extraction based on bidimensional empirical mode decomposition in shadow field," in *Proceedings of the 2007 International Workshop on Signal Design and Its Application in Communication*, 2007.

[98] X. Gan, W. Huang, J. Yang, and B. Fu, "Internal wave packet characterization from SAR images using empirical mode decomposition(EMD)," in *Proceedings of the 2008 Congress on Image and Signal Processing*, 2008.

[99] M. Ahmed and D. P. Mandic, "Image fusion based on fast and adaptive bidimensional empirical mode decomposition," in *Proceedings of the 13th Conference on Information Fusion*, 2010.

[100] K. Hamad, "Hybrid empirical mode decomposition-neural model for short term travel time prediction on freeways," Ph.D. dissertation, University of Delaware, 2004.

[101] R. Li and Y. Wang, "Short-term wind speed forecasting for wind farm based on empirical mode decomposition," in *Proceedings of the 2009 International Conference on Electrical Machines and Systems*, 2008.

[102] X. Fan and Y. Zhu, "The application of empirical mode decomposition and gene expression programming to short term load forecasting," in *Proceedings of the 6th International Conference on Natural Computation*, 2010.

[103] R. Zhang, Y. Bao, and J. Zhang, "Forecasting erratic demand by support vector machines with ensemble empirical mode decomposition," in *Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences*, 2010.

[104] R, "The R project for statistical computing," 2010.

[105] J. Ramsay, G. Hooker, and S. Graves, *Functional Data Analysis with R and Matlab*. Springer, 2009.

[106] D. Hiebeler, "Matlab/R reference, by David Hiebeler," http://www.math.umaine.edu/ hiebeler/comp/matlabR.html, 2010.

[107] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000.

[108] M. C. Little, "JavaSim user's guide," http://javasim.ncl.ac.uk/manual/javasim.pdf, 1999.

[109] D. G. Feitelson, "The standard workload format," http://www.cs.huji.ac.il/labs/parallel/workload/swf.html, 2006.

[110] B. Lu, L. Ngo, H. Bui, A. Apon, N. Hamm, L. Dowdy, D. Hoffman, and D. Brewer, "Workload modeling for performance management," in *Computer Measurement Group International Conference*, 2008.

[111] D. Tsafarir and D. G. Feitelson, "Instability in parallel job scheduling simulation: The role of workload flurries," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium*, 2006.

[112] J. Shiskin and H. Eisenpress, "Seasonal adjustments by electronic computer methods," *Journal of The American Statistical Association*, vol. 52, 1957.

[113] W. P. Cleveland and G. C. Tiao, "Decomposition of seasonal time series: A model for the census X-11 program," *Journal of the American Statistical Association*, vol. 71, 1976.

[114] S. C. Hiller and G. C. Tiao, "An ARIMA-model-based approach to seasonal adjustment," *Journal of the American Statistical Association*, vol. 77, 1982.

[115] L. Simmons, "Time-series decomposition using the sinusoidal model," *International Journal of Forecasting*, vol. 6, 1990.

[116] V. Assimakopoulos and K. Nikopoylos, "The theta model: A decomposition approach to forecasting," *International Journal of Forecasting*, vol. 16, 2000.

[117] R. J. Hyndman and B. Billah, "Unmasking the theta method," *International Journal of Forecasting*, vol. 19, 2003.

[118] J. S. Armstrong, F. Collopy, and J. Yokum, "Decomposition by causal forces: A procedure for forecasting complex time series," *International Journal of Forecasting*, vol. 21, 2005.

[119] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software*, vol. 26, 2008.

[120] D. Kim and H. Oh, "EMD: A package for empirical mode decomposition and hilbert spectrum," *The R Journal*, vol. 1, 2009.

[121] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, 2006.

[122] P. Flandrin and P. Goncalves, "Empirical mode decomposition as a data-driven wavelet-like expansions," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 2, 2004.

[123] P. Flandrin, G. Rilling, and P. Goncalves, "Empirical mode decomposition as a filter bank," *IEEE Signal Processing Letters*, vol. 11, 2004.

[124] T. Kijewski-Correa and A. Kareem, "Performance of wavelet transform and empirical mode decomposition in extracting signals embedded in noise," *Journal of Engineering Mechanics*, vol. 133, 2007.

Appendix A

Exhaustive Prediction Experimental Results for the Comparisons among Different Forecasting Approaches: EMD/VAR, EMD/ARIMA, ARIMA, and ETS

The Tables in this appendix contain the experimental results of chapter 5. For Tables A.1 through A.7, each Table contains the calculated MASE for each day of week. In particular, each Table contains the mean and standard deviation of the MASEs from the five groups as described in chapter 5 (EMD/VAR, EMD/ARIMA, ARIMA, ETS, and Undecided). The last row of the Tables contains the count of observations for each of the groups, The remaining rows of the Tables are grouped into four horizontal groups, each represents a window range: 3, 6, 12, and 24 hours. Table A.8 contains the MASE measurements of the 4-week average forecast technique with the columns are the days of week and the rows are the 3, 6, 12, and 24 hours forecast window.

The Figures A.1, A.2, A.3, and A.4 demonstrate the example of the cases where EMD/VAR, EMD/ARIMA, ARIMA, and ETS offer the best forecast results, respectively.

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	3.01	2.00	2.65	3.05	2.92	1.43	5.64	7.72	3.11	3.42
EMD/ARIMA	3.40	3.17	1.45	1.69	2.82	1.60	4.05	4.49	2.89	2.09
ARIMA	3.38	2.69	1.22	0.70	3.14	3.83	4.43	4.85	3.29	2.62
ETS	2.48	2.40	1.27	1.28	1.60	1.06	0.79	0.51	2.20	1.52
	6-hour window									
EMD/VAR	2.48	1.48	2.35	1.49	2.29	0.86	4.41	3.22	2.51	1.60
EMD/ARIMA	2.47	1.42	1.74	1.57	2.27	0.80	3.96	2.52	2.47	1.25
ARIMA	2.62	1.54	1.61	1.02	2.16	1.17	4.34	2.44	2.71	1.28
ETS	2.76	1.68	1.97	1.98	1.43	0.79	0.89	0.50	1.94	1.03
	12-hour window									
EMD/VAR	1.90	0.58	1.97	0.86	2.26	0.66	2.72	0.26	2.20	0.61
EMD/ARIMA	2.31	0.65	1.82	0.79	2.35	0.71	2.63	0.41	2.09	0.69
ARIMA	2.35	0.73	2.22	0.80	2.10	0.76	2.75	0.30	2.26	0.72
ETS	2.04	0.75	1.86	0.80	1.91	0.49	1.33	0.45	1.84	0.52
	24-hour window									
EMD/VAR	1.92	0.63	2.05	0.61	2.39	0.57	2.67	0.30	2.14	0.72
EMD/ARIMA	2.46	0.87	1.82	0.70	2.56	0.66	2.78	0.58	2.06	0.76
ARIMA	2.80	1.52	2.28	0.61	2.20	0.56	2.90	0.42	2.24	0.77
ETS	2.27	0.97	1.80	0.65	2.17	0.59	1.64	0.11	1.93	0.70
Percentage	24		14		11		3		9	

Table A.1: *Error in Forecast Results as Measured by MASE for Mondays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	3.20	2.66	6.31	6.11	6.24	6.49			1.58	1.48
EMD/ARIMA	2.85	2.78	5.46	7.12	5.01	5.66			1.78	0.97
ARIMA	4.74	4.54	7.43	5.72	2.74	1.92			1.78	1.20
ETS	2.85	3.77	2.54	2.83	2.75	2.44			1.24	0.77
	6-hour window									
EMD/VAR	2.73	1.65	6.04	7.00	5.68	4.54			2.54	2.35
EMD/ARIMA	2.21	1.24	5.86	9.14	4.86	3.78			2.75	2.10
ARIMA	5.76	4.80	8.90	8.07	3.01	2.75			2.78	2.56
ETS	2.62	2.27	3.19	3.01	2.44	1.84			1.98	1.91
	12-hour window									
EMD/VAR	1.68	0.63	2.01	1.04	2.56	1.29			2.05	0.51
EMD/ARIMA	2.18	0.64	2.16	0.78	3.01	1.36			2.07	0.43
ARIMA	2.46	1.02	3.40	1.39	1.77	0.80			2.16	0.83
ETS	1.85	0.59	1.97	0.49	2.00	0.61			1.80	0.38
	24-hour window									
EMD/VAR	1.65	0.69	2.85	2.47	2.51	1.89			2.10	0.35
EMD/ARIMA	2.38	0.82	2.12	0.90	3.52	4.19			2.01	0.40
ARIMA	2.48	0.84	4.00	1.95	1.91	1.30			2.15	0.61
ETS	1.95	1.22	2.10	0.44	2.24	1.56			1.82	0.29
Percentage	25		10		16		0		8	

Table A.2: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Tuesdays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	5.33	5.56	2.69	1.64	5.49	6.45			2.93	1.86
EMD/ARIMA	4.72	5.40	1.95	1.56	4.85	6.10			2.31	1.32
ARIMA	5.04	4.45	3.09	3.31	4.77	5.41			3.22	2.31
ETS	2.92	3.72	2.24	2.24	3.09	3.19			1.26	0.67
	6-hour window									
EMD/VAR	3.84	3.54	3.53	1.89	5.98	6.98			2.94	1.77
EMD/ARIMA	3.51	3.55	2.50	1.05	5.54	8.07			2.78	1.57
ARIMA	4.62	3.37	4.61	3.90	6.36	8.39			5.31	5.43
ETS	2.20	1.38	2.98	2.61	3.29	3.06			1.89	1.66
	12-hour window									
EMD/VAR	1.87	0.97	2.15	0.68	2.24	1.16			1.86	1.04
EMD/ARIMA	2.41	1.23	2.42	0.84	2.29	0.73			2.28	0.73
ARIMA	2.34	1.35	2.58	0.84	1.97	0.70			2.68	1.41
ETS	1.77	0.37	2.18	0.76	1.79	0.43			1.87	0.60
	24-hour window									
EMD/VAR	1.95	1.52	1.93	0.76	2.56	0.99			2.54	2.73
EMD/ARIMA	2.92	3.56	2.32	0.82	2.38	0.76			2.38	0.73
ARIMA	2.50	1.39	2.36	0.95	2.58	1.12			2.83	2.14
ETS	2.05	1.30	2.23	1.52	1.89	0.57			1.81	0.40
Percentage	23		15		13		0		9	

Table A.3: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Wednesdays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	2.03	2.77	3.57	4.96	5.54	12.77			9.00	0.93
EMD/ARIMA	1.84	0.77	1.26	1.05	5.19	12.12			7.20	0.56
ARIMA	1.71	1.04	1.39	1.04	1.99	2.47			5.86	1.43
ETS	1.70	1.36	0.99	0.59	2.85	3.81			6.59	1.69
	6-hour window									
EMD/VAR	2.48	3.65	4.69	4.04	7.75	19.04			11.72	1.45
EMD/ARIMA	2.19	1.04	2.04	1.85	7.70	21.08			8.53	0.91
ARIMA	2.55	2.01	1.88	0.78	2.63	2.56			6.38	1.08
ETS	1.88	1.21	1.57	0.71	3.63	4.31			6.62	1.29
	12-hour window									
EMD/VAR	1.67	0.73	2.51	1.70	2.43	0.92			8.36	0.51
EMD/ARIMA	2.01	0.52	1.59	0.45	2.48	0.63			8.70	0.19
ARIMA	1.78	0.66	1.75	0.42	2.24	3.10			7.40	0.29
ETS	1.78	0.50	1.85	0.50	2.11	1.45			7.43	0.38
	24-hour window									
EMD/VAR	1.39	0.27	2.38	1.48	2.26	0.80			8.00	0.21
EMD/ARIMA	1.93	0.53	1.44	0.38	2.43	0.65			7.67	0.71
ARIMA	1.83	0.69	1.74	0.44	1.71	0.77			6.92	0.24
ETS	2.03	0.96	1.92	0.71	1.96	0.78			6.41	0.57
Percentage	20		12		23		0		4	

Table A.4: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Thursdays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	2.38	1.97	1.00		4.44	3.93			12.79	20.41
EMD/ARIMA	1.92	1.18	0.56		3.00	2.54			5.37	9.18
ARIMA	2.17	1.42	0.80		1.80	1.47			10.00	20.48
ETS	1.99	1.67	0.39		1.78	1.26			4.77	9.11
	6-hour window									
EMD/VAR	2.58	1.37	1.96		5.02	3.93			8.43	7.98
EMD/ARIMA	2.49	1.57	1.00		4.00	3.09			4.32	3.74
ARIMA	3.02	1.90	1.24		2.19	1.40			8.03	10.30
ETS	2.40	2.00	0.52		2.13	1.14			3.68	4.16
	12-hour window									
EMD/VAR	1.64	0.39	2.01		2.85	1.41			3.52	2.41
EMD/ARIMA	2.11	0.76	1.54		2.61	1.21			3.14	2.44
ARIMA	1.94	0.35	1.91		1.78	0.53			3.27	2.79
ETS	1.90	0.43	1.82		1.93	0.75			2.12	1.19
	24-hour window									
EMD/VAR	1.56	0.44	1.75		2.96	1.38			3.26	1.88
EMD/ARIMA	2.07	0.62	1.24		2.45	1.19			2.98	2.64
ARIMA	1.98	0.59	2.04		1.67	0.51			3.15	3.23
ETS	1.90	0.54	1.77		2.06	1.18			2.24	1.48
Percentage	24		1		22		0		13	

Table A.5: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Fridays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	3.62	4.32	2.59	2.56	4.40	5.08			2.17	1.38
EMD/ARIMA	2.83	3.00	2.19	1.86	3.45	2.45			3.30	2.41
ARIMA	3.65	3.06	2.48	2.24	1.66	0.96			3.41	2.37
ETS	2.18	2.58	1.42	1.18	1.58	1.24			2.24	2.00
	6-hour window									
EMD/VAR	1.75	0.72	1.32	0.56	3.08	3.41			3.39	2.46
EMD/ARIMA	2.06	0.88	1.46	1.00	3.31	3.74			3.47	2.53
ARIMA	2.36	1.62	1.68	0.97	1.51	0.75			3.72	2.35
ETS	1.52	0.72	1.27	0.83	1.65	0.98			2.28	1.21
	12-hour window									
EMD/VAR	1.68	1.18	1.20	0.45	2.95	1.74			3.82	2.34
EMD/ARIMA	1.98	1.19	1.22	0.52	2.75	1.88			3.03	1.33
ARIMA	1.98	1.65	1.40	0.52	1.38	0.88			2.90	1.64
ETS	1.64	1.23	1.08	0.32	1.69	1.32			1.67	0.47
	24-hour window									
EMD/VAR	1.97	1.34	1.83	1.08	3.72	1.40			4.20	3.85
EMD/ARIMA	1.99	1.16	1.30	0.42	3.46	1.74			3.10	2.97
ARIMA	2.03	1.27	1.75	0.46	1.65	0.93			2.10	0.72
ETS	2.23	2.05	1.53	0.66	2.09	1.78			1.73	0.89
Percentage	15		17		18		0		8	

Table A.6: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Saturdays*

	EMD/VAR		EMD/ARIMA		ARIMA		ETS		Undecided	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
	3-hour window									
EMD/VAR	3.36	3.64	7.39	4.44	7.13	8.44			5.47	4.87
EMD/ARIMA	4.20	3.88	3.70	3.57	5.60	7.84			3.49	3.83
ARIMA	4.18	4.76	8.64	10.07	3.83	3.32			2.33	2.05
ETS	4.72	7.71	2.48	1.96	3.95	4.89			1.86	1.69
	6-hour window									
EMD/VAR	2.65	1.11	5.97	3.41	7.57	9.92			5.89	5.11
EMD/ARIMA	3.13	2.02	2.21	1.65	6.38	8.45			4.14	4.13
ARIMA	2.59	2.22	7.18	9.01	3.58	2.86			2.43	1.62
ETS	2.88	2.70	2.77	2.42	3.64	3.96			2.24	2.32
	12-hour window									
EMD/VAR	3.35	3.37	4.13	2.91	4.63	3.55			4.66	3.36
EMD/ARIMA	2.50	1.84	1.53	0.35	3.90	2.84			3.91	3.11
ARIMA	2.68	3.11	3.58	2.29	2.38	1.38			2.28	1.53
ETS	2.57	1.01	3.19	3.12	2.73	2.26			2.32	2.29
	24-hour window									
EMD/VAR	1.75	0.34	3.93	2.13	3.49	2.34			3.56	2.39
EMD/ARIMA	2.18	0.80	1.52	0.63	3.77	2.31			3.37	2.20
ARIMA	1.93	0.90	4.00	3.08	1.76	0.47			2.47	2.19
ETS	2.75	2.69	4.81	7.32	3.03	3.91			2.08	2.31
Percentage	9		7		14		0		24	

Table A.7: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Sundays*

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
3 Hours Forecast Window	2.21	22.49	21.54	6.38	18.51	19.77	2.63
6 Hours Forecast Window	2.33	18.75	14.29	11.34	15.20	12.14	5.19
12 Hours Forecast Window	5.60	6.82	5.02	6.06	7.43	5.78	4.67
24 Hours Forecast Window	3.35	5.53	3.21	4.93	5.02	4.51	2.41

Table A.8: *MASE Accuracy Measurements of the 4-week Average Estimation Forecast for Days of Week*

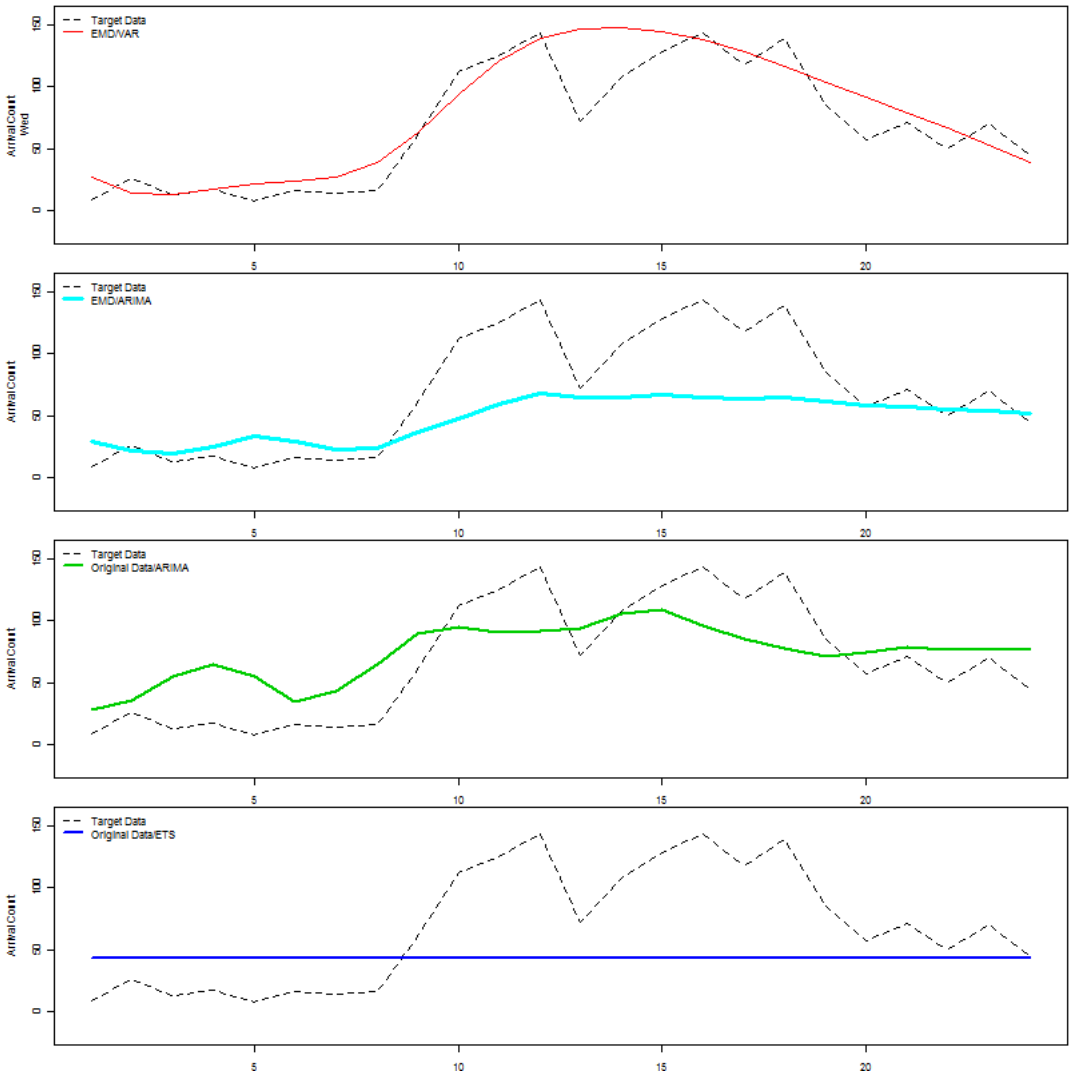

Figure A.1: *Example of the case where EMD/VAR offers the best prediction.*

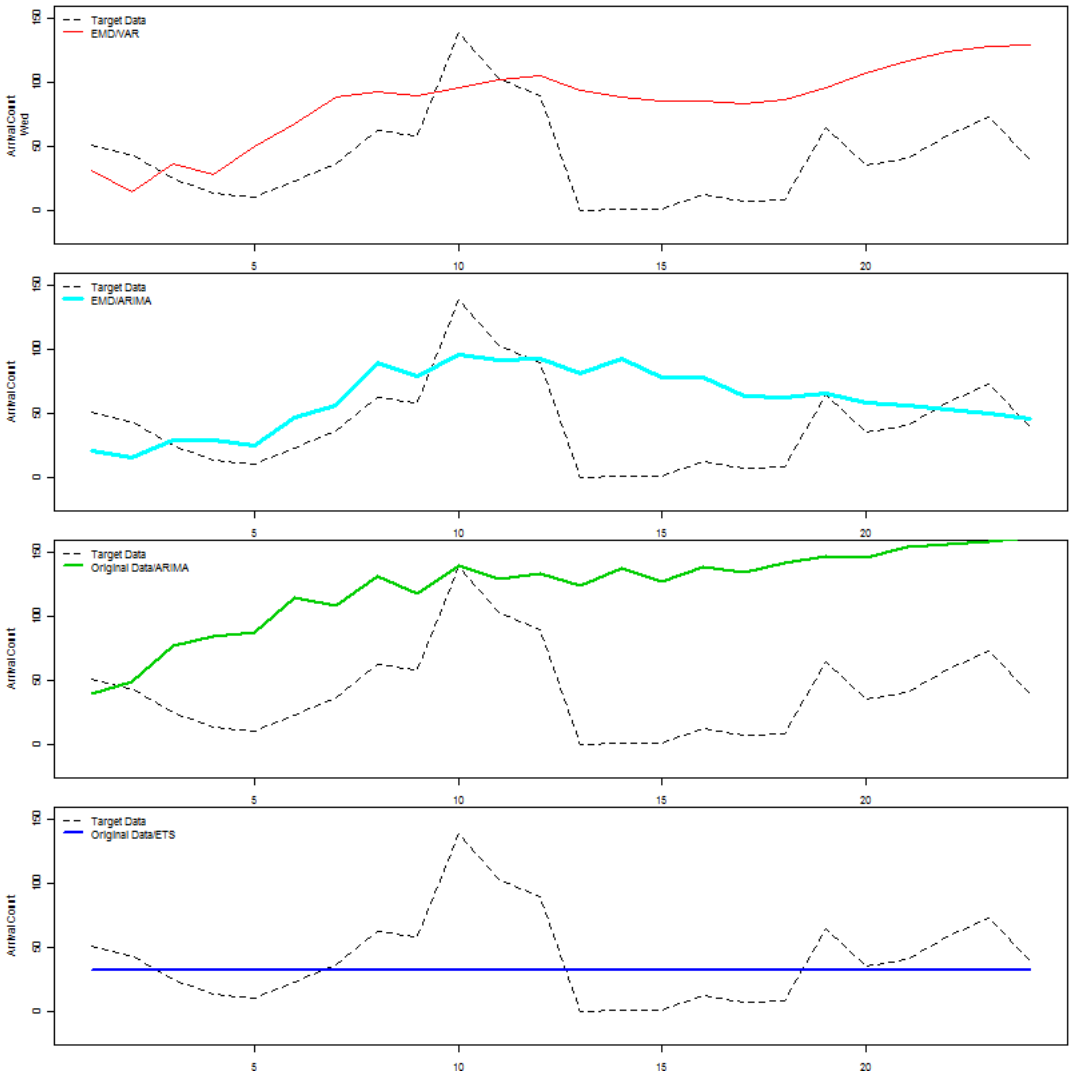

Figure A.2: *Example of the case where EMD/ARIMA offers the best prediction.*

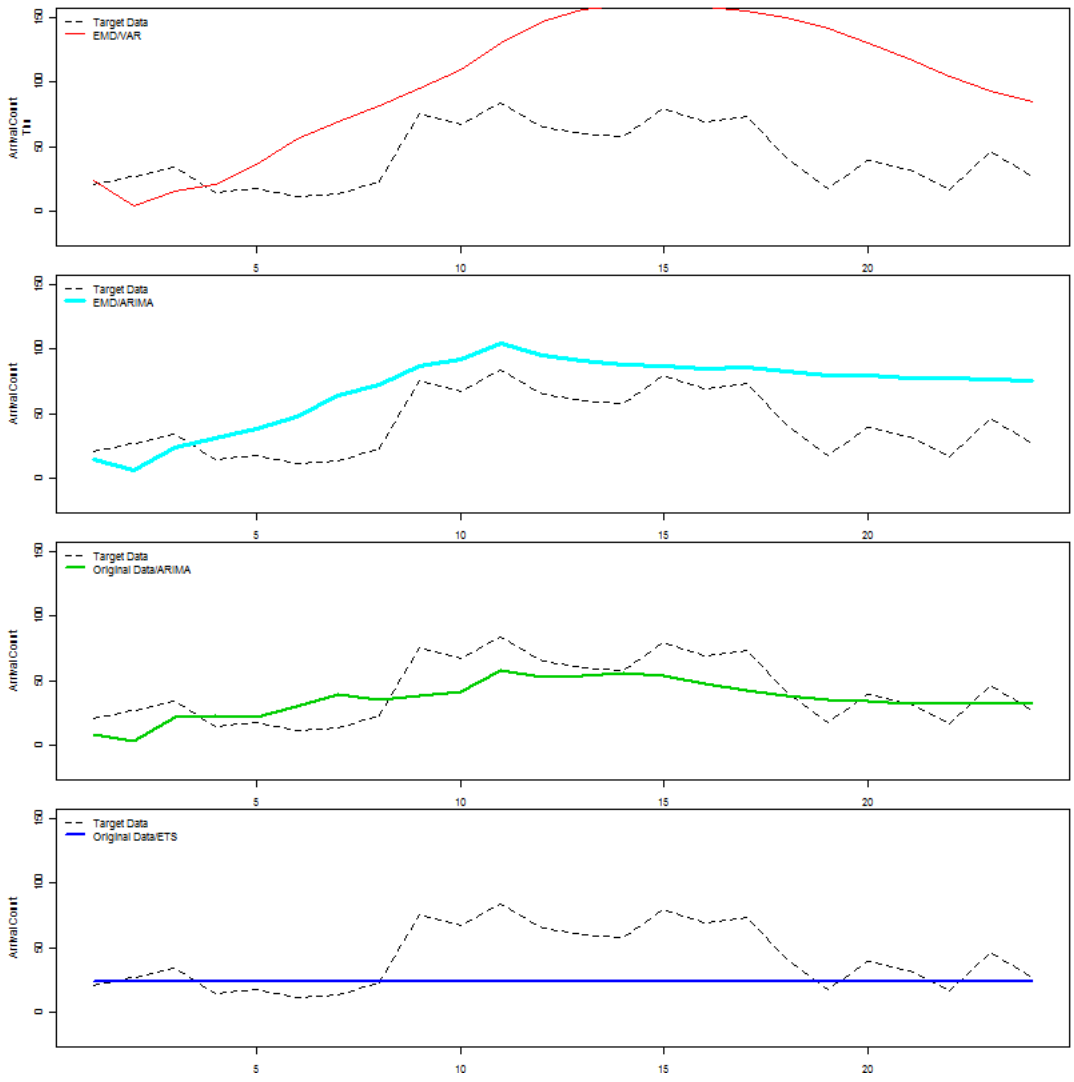

Figure A.3: *Example of the case where ARIMA offers the best prediction.*

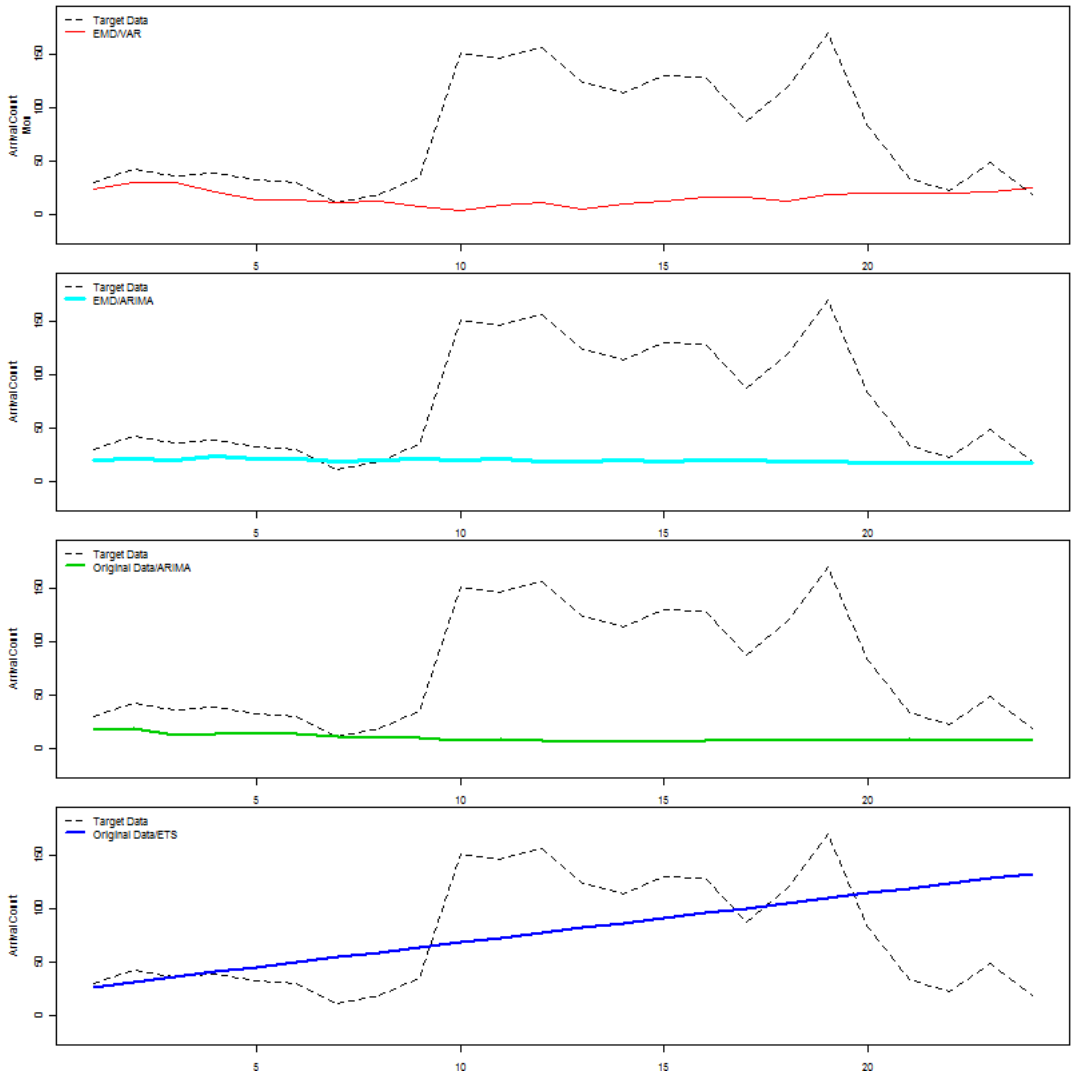

Figure A.4: *Example of the case where ETS offers the best prediction.*

Appendix B

Exhaustive Prediction Experimental Results for the Comparisons among the two data pre-processing approaches: EMD and Wavelet

The table in this appendix contains the experimental results of chapter 6. Each column of table contains the average of the MASEs from each day of week, as described in chapter 6. The rows of the table are grouped into four horizontal groups, each represents a window range: 3, 6, 12, and 24 hours.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	3-hour window						
EMD/VAR	3.06	4.33	4.14	2.71	5.28	3.23	5.80
Wavelet/VAR							
EMD/ARIMA	2.81	3.73	3.33	2.04	3.00	2.89	4.18
Wavelet/ARIMA	2.28	3.19	2.30	2.64	2.70	2.96	2.50
	6-hour window						
EMD/VAR	2.52	4.06	4.00	3.48	4.68	2.24	5.80
Wavelet/VAR							
EMD/ARIMA	2.40	3.62	3.47	2.59	3.39	2.46	4.30
Wavelet/ARIMA	1.85	2.17	1.82	2.69	2.50	1.66	2.04
	12-hour window						
EMD/VAR	2.07	2.03	2.21	2.13	2.48	2.24	4.37
Wavelet/VAR							
EMD/ARIMA	2.19	2.39	2.17	2.08	2.49	2.14	3.37
Wavelet/ARIMA	1.60	1.67	1.67	1.94	1.63	1.21	1.74
	24-hour window						
EMD/VAR	2.11	2.15	2.04	1.95	2.42	2.86	3.29
Wavelet/VAR							
EMD/ARIMA	2.29	2.59	2.15	1.99	2.37	2.42	3.03
Wavelet/ARIMA	1.71	1.65	1.69	1.61	1.57	1.44	1.53

Table B.1: *Accuracy Comparison among Forecast Approaches as Measured by MASE for Sundays*