COLLECTIVE DATA FORECASTING IN DYNAMIC TRANSPORT NETWORKS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

By
Mehmet Güvercin
September 2021

COLLECTIVE DATA FORECASTING IN DYNAMIC TRANSPORT NETWORKS

By Mehmet Güvercin

September 2021

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

İbrahim Körpeoğlu(Advisor)

Buğra Gedik(Co-Advisor)

Özgür Ulusoy

Uğur Güdükbay

İsmail Hakkı Toroslu

Ahmet Coşar

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

COLLECTIVE DATA FORECASTING IN DYNAMIC TRANSPORT NETWORKS

Mehmet Güvercin
Ph.D. in Computer Engineering
Advisor: İbrahim Körpeoğlu
Co-Advisor: Buğra Gedik
September 2021

Forecasting is a crucial tool for intelligent transportation systems and passengers of these systems and critical for transportation planning and management, as the transportation variable (e.g. delay, traffic speed) are among major costs in transportation. Each transportation variable may cause a further propagation in dynamic transport network. Hence, the transportation variable pattern of a node and the location of the node in the transport network can provide useful information for other nodes. We address the problem of forecasting transportation variable of a transport network node, utilizing the network information as well as the transportation variable patterns of similar nodes in the network.

We propose ECFM, Exploratory Clustered Forecasting Modeling, on both static and dynamic transportation network which makes use of graph based features for time-series estimation. ECFM approach builds a representative time-series for each group of nodes in the transport network and fits a common model like Seasonal Autoregressive Integrated Moving Average (SARIMA), Long-Short Term Memory (LSTM), Regression with Autoregressive Integrated Moving Average errors (REG-ARIMA), Regression with Long-Short Term Memory errors (REG-LSTM) for each, using the network based features as regressors. The models are then applied individually to each node data for predicting the node's transportation variable.

We perform a network based analysis of the transport network and identify graph-based features and we represent nodes as vectors that are used for both grouping nodes and as regressors in forecasting models. We evaluate proposed ECFM, Exploratory Clustered Forecasting Modeling, on two datasets (flight delay dataset, traffic speed dataset). The experiments show that ECFM provides accurate forecasts of delays/traffics compared to individual forecasting models. Centrality measure of nodes such as betweenness centrality score is found to be an effective regressor in the clustered modeling. Clustered models built on dynamic

networks performs better compared to static networks.

ECFM, Exploratory Clustered Forecasting Modeling, is an conceptual approach and it is domain independent. Our proposed approach tries to incorporate information, related to estimated variable, exist in similar nodes of the network. Thus, we can achieve to build robust estimation models on enriched data.

Keywords: Time Series Forecasting, Clustered Forecasting Models, Delay Estimation, Traffic Forecasting, Network Clustering, Dynamic Transport Networks.

ÖZET

DİNAMİK ULAŞTIRMA AĞLARINDA KOLEKTİF VERİ TAHMİNLEMESİ

Mehmet Güvercin
Bilgisayar Mühendisliği, Doktora
Tez Danışmanı: İbrahim Körpeoğlu
İkinci Tez Danışmanı: Buğra Gedik
Eylül 2021

Tahminleme işi akıllı ulaşım sistemleri ve bu sistemleri kullananan yolcular için önemli olmakla birlikte ulaşımla ilgili yönetim ve planlamada kritik role sahiptir, gecikme süreleri ve trafik hızı gibi değişkenler ulaşımla ilgili başlıca maliyetleri oluşturmaktadırlar. Her bir ulaşım değişkeni dinamik ulaşım ağlarında ilave yayılmalara sebep olmaktadır. Bunun sonucu olarak, ulaşım ağındaki bir düğümün ulaşım değişkeni örüntüsü ve ağdaki yeri diğer düğümlere yararlı bilgiler sağlayabilir. Biz bu çalışmada bir ulaşım ağındaki düğüme bağlı ulaşım değişkeninin tahminlemesinde ağ bilgisinin ve ulaşım değişkeni ile ilgili benzer düğümlerin davranışlarının ortaklaştırılması gerektiğini dikkate sunmaktayız.

Bu çalışmada statik ve dinamik ulaşım ağlarında zaman serisi tahminlemesinde ağ tabanlı özellikleri kullanan Keşifçi Kümelemeli Tahminleme Modelini (KKTM) önermekteyiz. Keşifçi Kümelemeli Tahminleme Modeli, ulaşım ağındaki her bir grup düğüm için temsili bir zaman serisi oluşturur ve bu zaman serilerinin her biri için ağ tabanlı özellikleri bağlayıcı değişken olarak kullarak mevsimler otoregresif hareketli ortalamalar (SARIMA), uzun kısa vadeli hafıza ağları (LSTM), bağlayıcı değişkenli otoregresif hareketli ortalamalar (REG-ARIMA), bağlayıcı değişkenli uzun kısa vadeli hafıza ağları (REG-LSTM) gibi ortak modeller inşa eder. Bu ortak modeller daha sonra her bir düğümün verisine ayrı ayrı uyarlanarak düğümün ulaşım değişkeninin tahminlemesi yapılır.

Bu tezde biz ulaşım ağındaki düğümleri gruplamak ve aynı zamanda tahminleme modellerine bağlayıcı değişken olarak girdi oluşturmak için ağ tabanlı teorik özellikleri ve düğüm vektörlerini çıkardık. Önerdiğimiz Keşifçi Kümelemeli Tahminleme Modelini (KKTM) iki ayrı veri seti (uçuş gecikmesi veri seti, trafik hızı veri seti) üzerinde test ettik. Deney sonuçlarına bakıldığında uçuş gecikmesi ve trafik hızı değişkenleri için önerilen keşifçi kümelemeli tahminleme modellerinin bireysel tahminleme modellerine göre daha hatasız tahminlemeler ortaya koyduğu

görülmektedir. Kümeleme modellerinde düğümlerin merkezi olma özelliğini ölçen merkeziyet arasındalık skorunun etkili bir bağlayıcı değişken olduğu tespit edilmiştir. Kümelemeli modellerin dinamik olarak oluşturulan ağlarda sitatik ağlara göre daha başarılı olduğu görülmüştür.

Bu tezde önerilen Keşifçi Kümelemeli Tahminleme Modeli uygulama alanı bağımsız kavramsal bir modeldir. Önerdiğimiz yöntem ağdaki benzer düğümlerde var olan tahminleme değişkeniyle ilgilili bilgileri ortaklaştırmayı sağladığı için zenginleştiriltimesi sağlanmış veri sayesinde daha güçlü tahminleme modelleri oluşturmaktadır.

Anahtar sözcükler: Zaman Serisi Tahminleme, Kümelemeli Tahminleme Modelleri, Gecikme Tahmini, Trafik Tahmini, Ağ Kümeleme, Dinamik Ulaşım Ağları.

Acknowledgement

Thanks to Scientific and Technological Research Council of Turkey (TUBİTAK) and Department of Computer Engineering for financially supporting me during my graduate education.

First, I am thankful to Prof. İbrahim Körpeoğlu and Prof. Buğra Gedik for their wisdom, teaching, and patience. I am very lucky that they were my advisors. I would like to thank to PhD committee members Prof. Özgür Ulusoy and Prof. İsmail Hakkı Toroslu. I am very thankful for their guidance, patience and encouragement during committee meetings. I am also thankful to Prof. Ahmet Coşar and Prof. Uğur Güdükbay for accepting being jury members. Their feedback made this thesis better.

Last, but not least, my family and my favorite friends deserve a medal for their support during thesis process and throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Easy access to the dynamic data or big data coming from current technologies makes easier to work on transportation forecasting. Forecasting is a crucial tool for intelligent transportation systems and critical for transportation planning and management. Major factors of delays or traffics include technical problems of vehicles, weather conditions, overuse of capacities, and delay/traffic propagation. While these factors are traditionally well studied, patterns due to the structure of networks have not been enough understood yet. In this thesis, we investigate whether the position of a node in the transportation network and information about similar nodes improve the estimation of delay or traffic patterns. We aim to forecast delays/traffics by incorporating *network information* and *similarity of delay/traffics patterns of nodes* into the estimation models. Accurate forecasting of delays/traffics is essential both for optimization of management operations and capacity planning.

The network of entities that are related to forecasting problem is first represented as a graph structure with each airport/loop as a node, and the number of flights or distance between two entities as the weight of the edge between the nodes. A set of graph-theoric features or vectorial graph features are extracted for each node. In particular, we adapt the measures of hub score, betweenness centrality, articulation point, in-degree, and weighted-in-degree and node2Vec [1]

into the context of these transportation networks.

We then use graph features and time-series patterns of delays/traffics to quantify similarities between airports/loops, and cluster the airports based on these similarities. We finally model each cluster of airports/loops with regression with model with regressors (M-REG) using the extracted features as regressors. The clusters are used to develop a joint model of airports/sensors for delay/traffic estimation. The information aggregated in the clusters helps to remove noise and handle outliers. We refer this approach as ECFM, Exploratory Clustered Forecasting Modeling, which makes use of graph based features for time-series estimation.

An extensive set of experiments is presented on millions of domestic flights between 305 airports in the United States over seven years from `http://www.transtats.bts.gov/` for flight delay forecasting and on 2016 observations (timesteps) of speed records over 207 sensor [2] for traffic forecasting. Developing a joint model for a cluster of airports based on graph features and delay patterns is shown to improve the estimation accuracy for individual airports. The betweenness centrality, which quantifies how important a node is in the routes of other node destination and arrival pairs, is found to be effective both for clustering the node and as a regressor in the M-REG model. Making networks dynamic and building more specific networks affects performance of network based model in positive way.

The presented network based analysis results can contribute to understanding the airport/los networks and their effect on delays. In particular, the analysed measures of network structure of airports and sensors can provide simple explanations to better understand network based delay/traffic behaviors. The proposed approach of using delay/traffic information of similar airports/sensors can help transportation system operators perform more effective planning and budgeting.

We summarize contributions of this thesis as follows.

- We define conceptually Exploratory Clustered Forecasting Modeling (ECFM) in order to incorporate network information to the forecasting models.
- We propose to use network based features as exploratory variables of estimation models.
- We test also proposed model on cases that include dynamically changing networks.
- The model we proposed can be used on all domains in which estimation model needed to be built on sequences related to network nodes.
- We present results of experiments considering a benchmark. Results of experiments illustrates that our methods perform better than baselines in the literature.

1.1 Thesis Outline

The rest of the thesis is organized as follows. Related works in literature for delay estimation, traffic forecasting, network analysis and time series modeling is discussed in Chapter 2. Chapter 3 defines network incorporation and node clustering and Chapter 4 discusses individual estimation models on related time series. Chapter 5 proposed exploratory clustered forecasting models and approaches in comparison. Chapter 6 gives and discusses experimental results for approaches in comparison. Finally, Chapter 7 concludes the thesis.

Chapter 2

Related Work

The proposed approach is related to the areas of flight delay estimation, traffic forecasting, time series modeling, and network analysis. We summarize the related work and how our method is placed in the literature for each these areas.

2.1 Flight Delay Estimation

Flight delay prediction has attracted significant attention both in practice and research literature [3]. Carriers and customers get affected by excess travel times, departure and arrival delays. Around 19% of the US domestic flights have a delay more than 15 minutes [4]. The causes of flight delays are studied from the perspectives of airlines and customers [5]. Airline hubbing and peaking airport concentration due to over-scheduling flights, besides other logistic and economic factors, are found to cause delays. Barnhart et al. study passenger delays as a factor for flight delays and derive findings for its causes [6]. They analyze flight cancellation and missed connections and develop a discrete choice model to estimate historical passenger travels. A taxonomy of flight delay prediction problems and a review of prediction approaches are presented in [3]. Carriers aim to consider airport network effect while deciding to postpone or cancel flights,

such as giving priority to flights that start or end in hub airports [7]. Our work introduces a graph based approach to flight delay estimation by incorporating the network features of the airports and analyzing them as groups in the network.

2.2 Traffic Speed Forecasting

Traffic speed forecasting is critical for planning and capacity management and it is studied considerable in the literature. In order to improve traffic prediction methods a wavelet transform and gated recurrent unit based technique is proposed in [8]. A novel neural network-based traffic forecasting method is proposed in [9] to capture spatial and temporal dependencies simultaneously. Authors generate the temporal graph convolutional network (T-GCN) model that combines graph convolutional network (GCN) and the gated recurrent unit (GRU). By the help of big data processing, work is done in order to select historical traffic data and suitable time series forecasting methods that achieve more accurate prediction [10]. Traffic prediction problem is challenging due to the complicated and dynamic spatio-temporal dependencies between different regions in the road network. Classical and deep learning based techniques are systematically classified and summarized in [11]. Authors also collect publicly available datasets in literature. Deep learning architectures are applied to the road traffic forecasting in [12] and these techniques perform better as demonstrated.

2.3 Time Series Modeling

The proposed method involves regression with ARIMA modeling and clustering. There is extensive work in these areas in the data mining and statistics literature. ARIMA is widely used for many applications, such as forecasting the electricity price [13], predicting the frequency and severity of accidents [14]. Seasonal ARIMA models (SARIMA models) are also well established in the literature

[15]. Time series clustering can be applied either over raw data, or models or features built over raw time series data [16]. Clustering is used to combine forecasts of time-series data [17]. Forecasting on multiple time series is recently studied via clustered models based on time series similarities, which helps to improve the scalability of forecasting methods [18]. Another line of research is to design Long short-term memory (LSTM) type recurrent neural networks [19] for a variety of machine learning problems on time-series data. Adapting these methods for multiple (delay) time-series data considering an underlying (airport) network structure is an interesting problem.

2.4 Network Analysis

Our work utilizes network analysis to better understand the transportation networks and forecasting flight delays and traffic speeds. We contribute to the literature by linking the graph features of the network nodes to their exhibited delay or speed patterns. Network analysis has made a significant impact in Web and social networks [20] [21] [22], starting from the early work by Freeman on measuring the structural centrality [23]. White and Borgatti adapt the centrality measures on undirected graphs to directed graphs [24]. Authoritative and hub scores of node sources in a hyper-linked environment are extensively studied in the literature [20, 25].

The network structure of airports has attracted some attention in air transportation research [26]. Santos and Robin analyze the variables, including the hub-airport variable, that explain the flight delays at the European airports [27]. Kim and Hansen introduce a non-parametric approach to estimate the effects of demand changes and throughput changes on delay [28]. Delay propagation of flights is modeled by considering both local congestion in individual airports and propagation of these delays over connected airports [29]. This approach aims to model the stochastic nature and time-varying behavior of airports. A network-based model is introduced to simulate the effects of aircraft ground movements in apron taxiways to gate assignment operations [30]. Airport network is also used

as an exploratory variable to obtain the global delay state of the entire system [31].

The recent US Federal Aviation Administration (FAA) Strategic Plan discusses to increase the throughput capacity of airports and congested air corridors [32]. Our approach can provide a network based insight to help prioritize certain airports in terms of the planned capacity enhancements. It can complement some of the tasks in FAA's research plan via a better understanding of the air transport networks. Among the functionalities of the Aviation Environmental Design Tool (AEDT), released by the FAA Office of Environment and Energy (AEE), are to model multiple airports in a single study, airplane taxi delay and sequence modelling [33].

Chapter 3

Network Incorporation and Node Clustering

Our proposed ECFM, Exploratory Clustered Forecasting Modeling, is based on incorporation of node features in estimation models. We need network incorporation for two cases while building forecasting models. In order to cluster nodes to have common data for each cluster we need network based features of nodes. Furthermore, in order to enrich estimation models with some network related regressors we also need network based features of nodes.

3.1 Node Clustering

Our proposed methodology needs to have clustered nodes of network from which we incorporate information to the forecasting models. We study four types of node clustering approach in this thesis as graph-theoretic clustering, node2vec clustering, graph partitioning, and time series clustering. We use airport network as a network instance in order to anaylze clustering techniques.

3.1.1 Graph-Theoretic Clustering

The graph-based features that we explore are: hub score, betweenness centrality, articulation point, in-degree and weighted in-degree. These features are fed to a clustering algorithm to obtain the airport clusters. We build SARIMA and REG-M (REG-ARIMA or REG-LSTM) models on a representative time-series for each cluster. We use Graph Theoretic Clustered (GTC) as initial word to define variants. We work graph-theoretic clustering on airport network and detailse are as follows.

Flights between airports are represented by an airport interaction graph: Each airport corresponds to a node and each flight between two airports corresponds to an edge between the nodes. The edge weight is calculated using total number of flights from the origin airport to the destination (i.e. 1/w, w: is the total number of flights from the origin to destination) . The data sets are collected from http://www.transtats.bts.gov/. We use this interaction network to analyze the topological properties of the airports within the global airport network. We visualize the airport network graph of with 305 nodes and 4622 edges using Cytoscape [34]. Every year has its own flight delays, so the airport network edges have different network scores. While doing experiments, we used their own network status for modeling. However, for the graph theoretical analysis later on we presented the scores for only one of the years since aggregating them would not be reasonable.

Delay vs. airport size and connectivity Figure 3.1 illustrates the top-30 airports which have the highest number of delays. The node sizes are shown according to number of delays (e.g., ATL has the highest number of delays). It is not surprising that the number of delays is correlated by the size of airport and the connectivity level of the nodes, as illustrated in Figure 3.2 and Figure 3.3. Selected airports are "ATL, "MWH" which have the highest, and the lowest number of delays in the US. Almost all of the airports are connected to "ATL" as a first neighbor, as depicted in Figure 3.2 in yellow and just one airport is connected to "MWH", as depicted in Figure 3.3. We observe that the airports

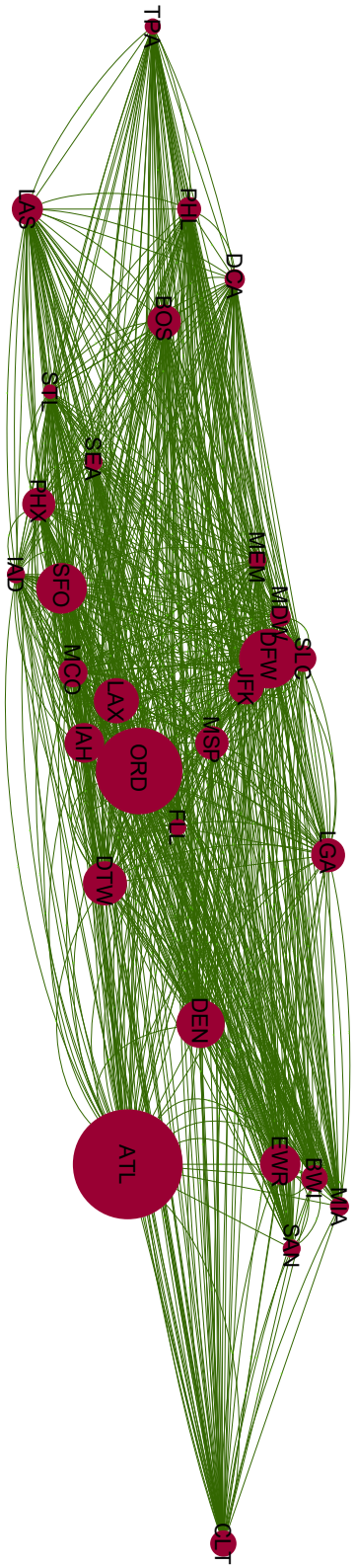

Figure 3.1: Top-30 airports of US aviation system considering number of delays

that have a high flight density is prone to have flight delays.

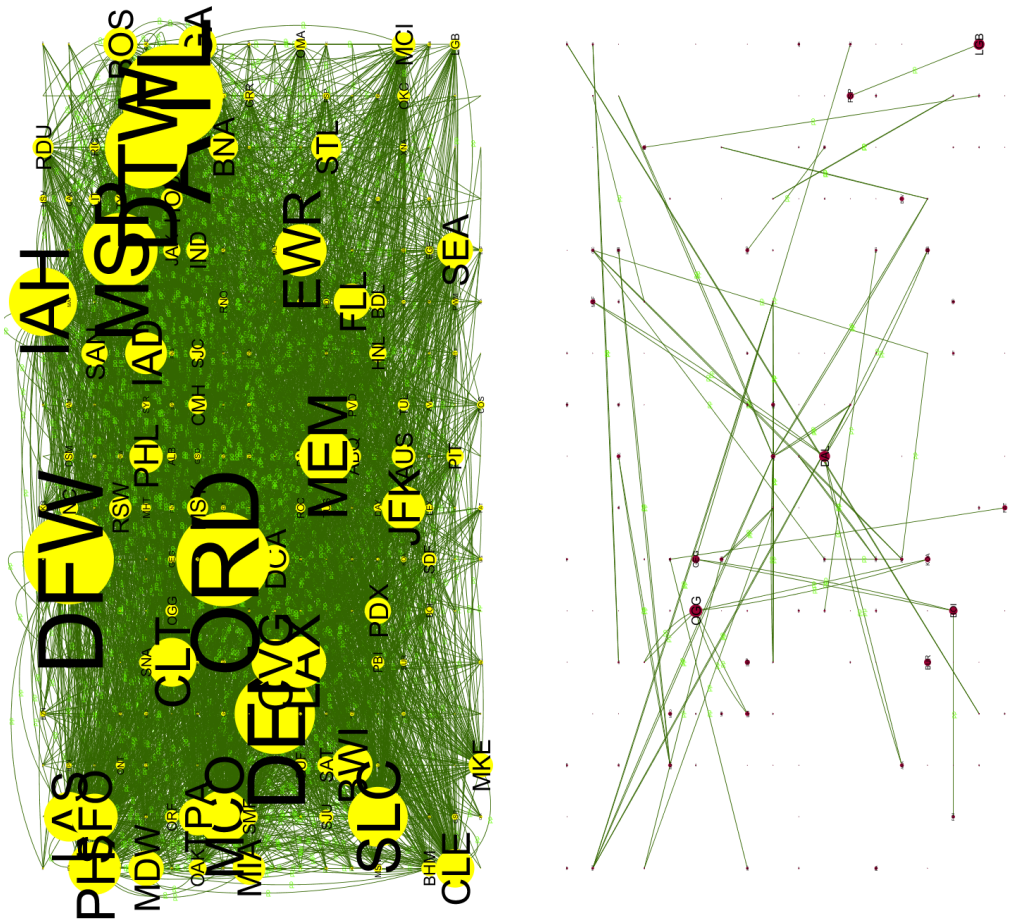

Figure 3.2: Left side: "ATL" first neighbors; right side: without "ATL" first neighbors

Graph-based features. The topological features include: the *hub score* of the airport, the *betwenness centrality* of the airport, and *articulation point(s)* on the graph. The node score features include *in-degree* and *weighted in-degree* of the airports.

Hub Score is the left-singular vectors of the Singular Value Decomposition (SVD) of the adjacency matrix A of a graph, which is used to represent the relative importance of a node in a network [20]. An airport with a high hub score is the origin of many flights to important and large-scale airports, and is naturally

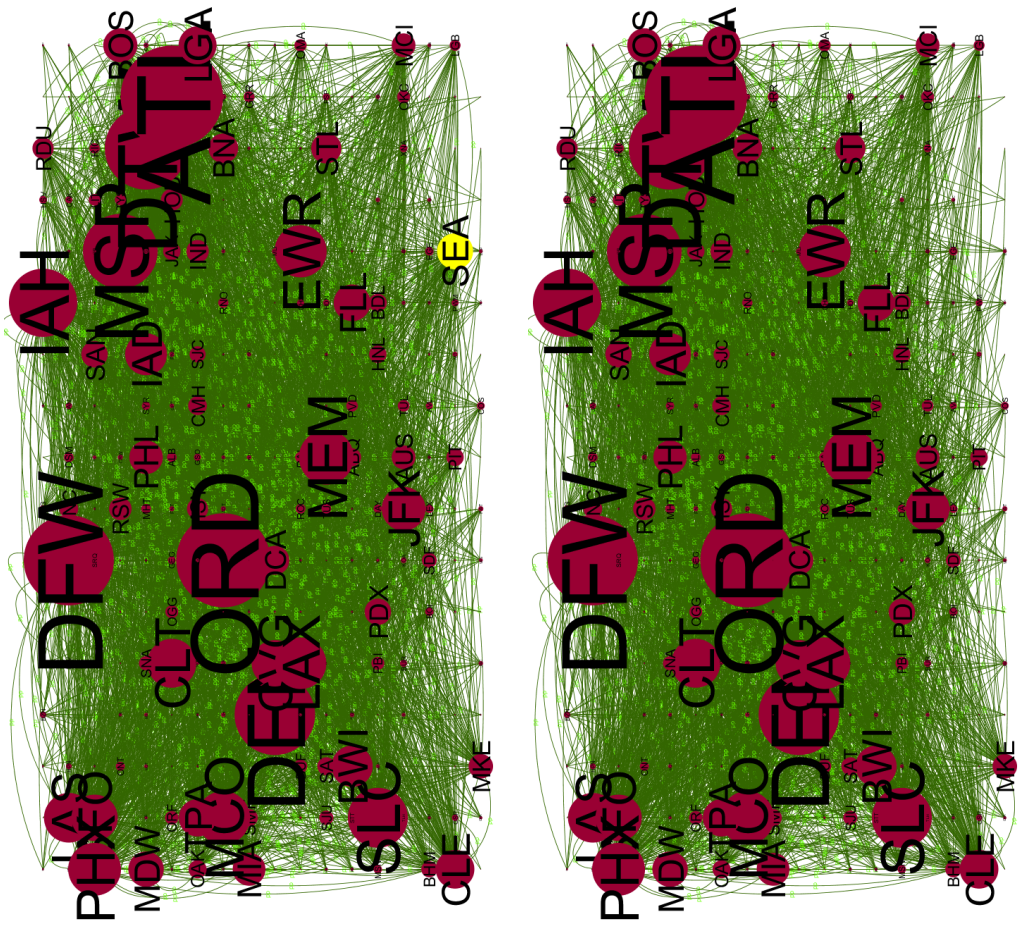

Figure 3.3: Left side: "MWH" first neighbors; right side: without "MWH" first neighbors

more important than the nodes with low hub score. Airports with similar hub scores may be expected to show similar behavior in terms of their arrival delays. Top 5 normalized hub scores can be seen in Table 3.1.

Betweenness centrality of node v in a directed graph $G = (V, E)$ can be represented as:

$$b(v) = \sum_{s \neq v \neq t \in V} \frac{p_{st}(v)}{p_{st}} \tag{3.1}$$

where node s to node t represented as p_{st} and the number of shortest paths that pass through node v total number of shortest paths from represented as $p_{st}(v)$. In

Table 3.1: Top-5 airports in the context of node scores

Rank	**Airport**	**Hub score**	**Airport**	**Betweenness.**
1	Hartsfield-Atlanta I.	1.00	Hartsfield-Atlanta I.	1.00
2	Chicago O'Hare I.	0.883	Dallas-Fort Worth I.	0.777
3	Dallas-Fort Worth I.	0.775	Chicago O'Hare I.	0.654
4	San Francisco I.	0.745	Salt Lake City I.	0.546
5	Denver I.	0.737	Detroit Metropolitan	0.539
Rank	**Airport**	**In-degree**	**Airport**	**W. in-degree**
1	Hartsfield-Atlanta I.	1.000	Hartsfield-Atlanta I.	1.000
2	Chicago O'Hare I.	0.924	Chicago O'Hare I.	0.755
3	Dallas-Fort Worth I.	0.886	Dallas-Fort Worth I.	0.645
4	Detroit Metropolitan	0.810	Denver I.	0.575
5	Denver I.	0.791	Los Angeles I.	0.481

order to apply idea of betweenness centrality to airport network context, we took the edge values as $1/w$ where w is number flights between corresponded nodes.

Algorithm 1: Find-articulation-points

```
dfsnum(v) ← −1, for all v
dfscounter ← 0 r ← |V|
for i ← 1 to r do
    v ← V_i
    if dfsnum(v) ≠ −1 then
        DFS(v)
```

Betweenness centrality (BC) of an airport can quantify its use as a popular transfer node between other airports in the network. An airport with high BC is in the path of many arrival-destination pairs and may denote some relationship for connecting flights. Being a central airport in the network naturally increases the density of the flight traffic. Considering percentage of intersected nodes in Figures 3.1 and 3.4 one can say that BC can serve as a potential indicator for delay behavior.

Articulation point of a graph is a node whose removal causes other nodes to be unreachable. Let $G = (V, E)$ be a directed graph, articulation points of graph G can be found by Algorithm 1 and Algorithm 2. *dfsnum* is a variable that keeps the information whether node v discovered or not. Also *dfscounter* counts *dfs*s for a specific node. We have identified 19 articulation points in the US airport data set that follows this definition.

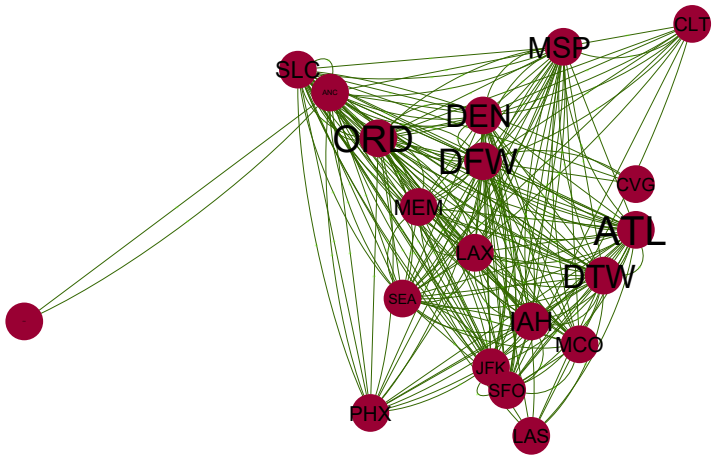

Figure 3.4: Top-20 airports with highest betweenness centrality scores

Articulation points of an airport network may be expected to have similarly high traffic behavior, causing high number of flight delays. 12 of 19 articulation points of the airport network are indeed among the top 19 airports ordered by the number of delayed flights. The majority of articulation points have high flight delays.

Algorithm 2: DFS(v)

$dfsnum(v) \leftarrow dfscounter$
$dfscounter \leftarrow dfscounter + 1$
$low(v) \leftarrow dfsnum(v)$
foreach *edge* (v, x) **do**
  if $dfsnum(x) == -1$ **then**
    DFS(x)
    $low(v) \leftarrow min\{low(x), low(v)\}$
    if $low(x) \geq dfsnum(v)$ **then**
      v is an art. point
  else if x *is not parent of* v **then**
    $low(v) \leftarrow min\{low(v), dfsnum(x)\}$

The number of neighbor airports and the number of flights are naturally related to the traffic and arrival delays. The number of airports that have flights to a node is the in-degree of an airport node. The in-degree of node v in a directed

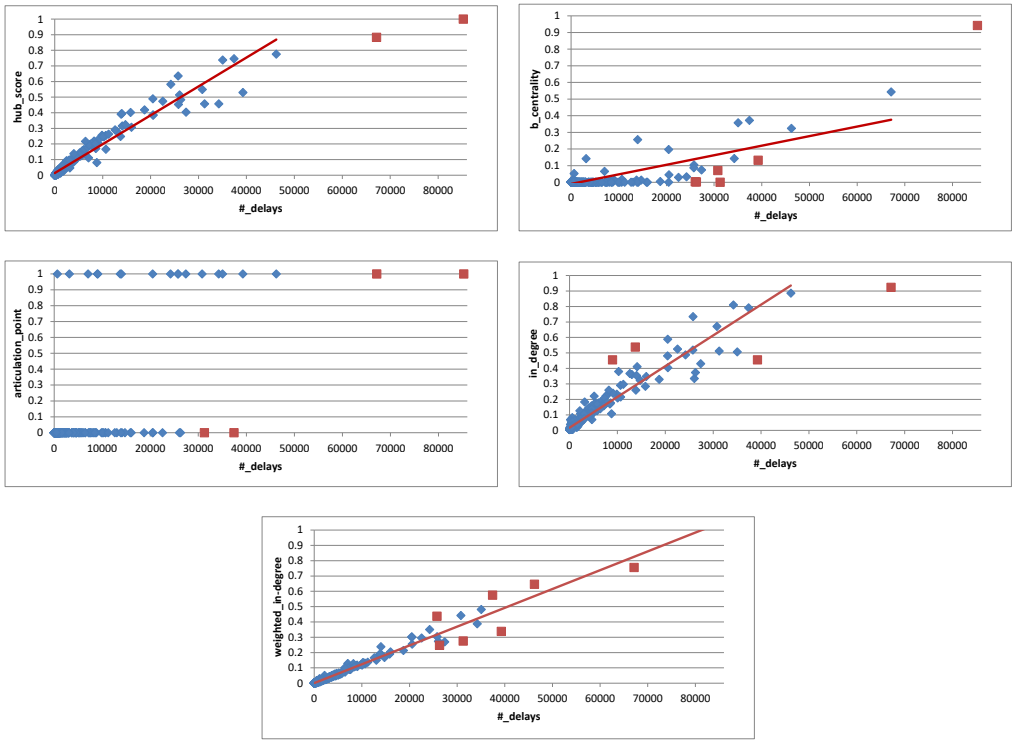

Figure 3.5: Graph-based scores vs. number of delays

graph $G = (V, E)$ can be calculated as:

$$id(v) = |\{u : (u, v) \in E\}| \tag{3.2}$$

The weighted in-degree of node v, the number of incoming flights, in a directed graph $G = (V, E)$ is:

$$wid(v) = \sum_{\{u:(u,v)\in E\}} w_{uv} \tag{3.3}$$

Table 3.1 lists the top 5 airports in terms of the presented scores. The top-30 highest delayed airports are illustrated in Figure 3.1. Figure 3.5 displays the correlation between the graph-based scores and the number of delayed flights. Red points in the figure represent outlier points in that score set and straight lines are obtained through linear regression on scores except outlier points. Scores presented in Table 3.1 and Figure 3.5 are normalized to the largest value.

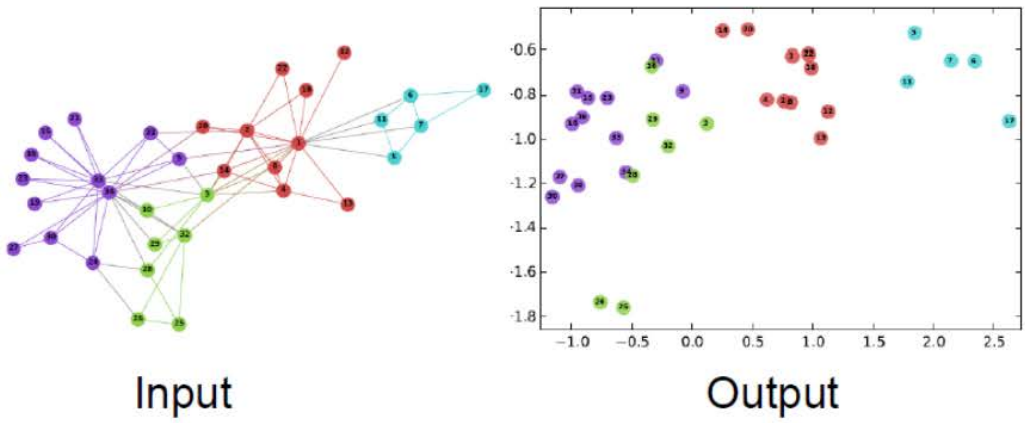

Figure 3.6: Zachary's Karate Club network embedding. Adapted from *"WWW-18 Tutorial Representation Learning on Networks"*, by J. Leskovec, 2018, SNAP, Retrieved from September 13, 2021 from http://snap.stanford.edu/proj/embeddings-www/.

3.1.2 Node2Vec Clustering

Node2Vec is another method of transforming nodes into the feature vectors. Node2Vec method tries to create embedding from nodes that makes model learning easier. Embedding is learnt as same way in [35] using skip-gram model. We use these transformed vectors as instance features while clustering nodes. Node2Vec embedding technique is developed by [1] This technique creates node vectors by optimizing a neighborhood preserving objective and mainly tries to optimize neighborhood preserving objective in order to output low dimensional representations of nodes. As an example, Zachary's Karate Club network's vector representation is ilustrated in Figure 3.6. In this thesis, we use Node2Vec Clustered (N2VC) as initial word to define variants.

3.1.3 Graph Partitioning

Graph partitioning can also be used to group the airports [36]. A partition in a network can be defined as a set of nodes with dense connections internally and sparser connections to outside of the partition. We identify partitions of airports and treat each group of airports as a hard partition. Several methods have been developed especially in the social network literature for partitioning and community detection, such as edge betweenness community [37], walk trap community [38], spin glass community [39], leading eigenvector community [36] fast greedy community [40]. Partitioning algorithms show similar performance in our case so we select the walk trap community algorithm. We use Graph Partitioned (GP) as initial word to define variants.

3.1.4 Time Series Clustering

Another approach we explore for clustering is to utilize signal information of airports' delay time series. We extract features using time series transformation methods, namely Discrete Fourier Transform (DFT) [41] and Discrete Wavelet Transform (DWT) [42]. We use Time Series Clustered (TSC) as initial word to define variants.

3.2 Network Incorporation

Network incorporation to the forecasting models is one of the main part of our proposed ECFM. Network incorporation basically means using information exists in network structure while building estimation models. This come true when network based node features are used as regressors inputs for the estimation model.

3.3 Dynamic Graphs

We firstly build proposed modeling approach on network that is created statically. However, today's big data world makes it possible and mandatory to think about incorporating data coming from dynamic networks. In this study, we first create networks using yearly information. Afterward we create monthly, weekly and daily network in order to investigate effect of dynamism in the networks.

Chapter 4

Estimation Models

In this work, we represent data as time series in order to build estimation models. There several estimation models for time series in literature. We present models that are used both individually and as a part of our proposed exploratory clustered forecasting models.

4.1 Time-Series Representation of Data

Both arrival delays of an airport and traffic speeds of sensors can be represented as a time series, a sequence of numerical points in successive order. We first decide time points and then calculate value of each time point. Considering arrival delays the value of each time point is the maximum (or median) arrival delay, in minutes, of the incoming flights to the airport for the corresponding period. For the purpose of experimentation, we produce the time series for one-year data of length 2920 (8 points for each day) for 305 distinct airports. We do this for seven years that are utilized in experimental section. We then estimate the delays of a period of three-hours in a day. Considering traffic speeds the value of each time point is the speed value measured at a specific sensor. Speed are recorded at every 5 minutes. We produce the time series for one-month data of length of for

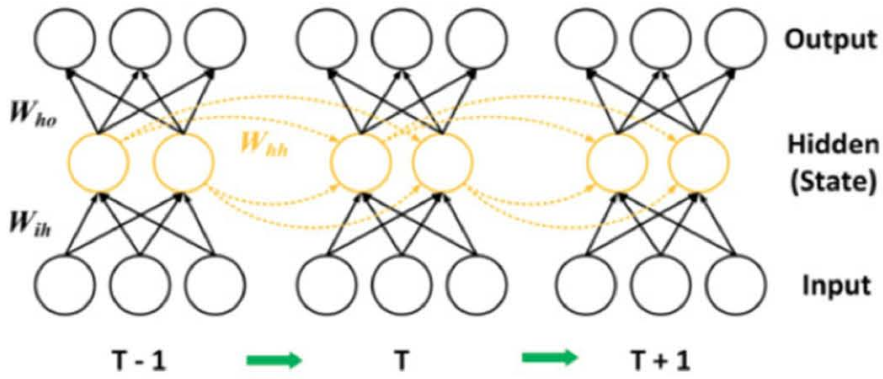

Figure 4.1: Time sequence processing in recurrent neural network

207 distinct sensors. We do this for four months and we estimate the speeds of a period o 5 minutes.

We presents example time series on airport delays. Figure 4.2 illustrates the delay behavior of time series of the selected 8 big airports for a day. IATA codes of airports are used instead of airports' names in graphs (ATL, SFO, LAX, etc.) Each delay point versus time in the graphs represents the maximum (median) delay occurred in a period of three-hours of a day and delays are measured in minutes. The figure shows the maximum and median based delay behaviors of airports, and illustrates that there are airports with similar delay behaviors to each other.

4.2 Individual Models

For the proposed approach we use five types of forecasting models: Multiple regression models, Seasonal Autoregressive Integrated Moving Average (SARIMA) family of models, Regression with ARIMA Errors (REG-ARIMA) models, Long-Short Term Memory models and Regression with LSTM Errors (REG-LSTM) models. We present definition of these models in following sections.

4.2.1 Multiple Regression Model

Multiple regression model represents a dependent variable y by using k multiple independent variables $x_1, x_2, ..., x_k$ as in the form of Equation 4.1. Building a regression model is the problem of finding the model's coefficient set $b_1, b_2, ..., b_k$.

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + ... + b_k * x_k \tag{4.1}$$

4.2.2 SARIMA Modeling

SARIMA modeling represents a time point of a time series as the linear combination of its past time points. A $SARIMA$ model $SARIMA(p, d, q)(P, D, Q)$ is represented by its autoregressive order p, differencing order d, moving average order q, seasonal autoregressive order P, seasonal differencing order D and seasonal differencing order Q. Building a $SARIMA$ model on given data series aims to determine the order of model and a vector of parameters. Further discussions on SARIMA model operators, stationarity of time series, and how to estimate the parameters of a SARIMA model can be found in [18].

4.2.3 REG-ARIMA Model

REG-ARIMA model is a combination of a regression and an Autoregressive Integrated Moving Average (ARIMA) model. To build a REG-ARIMA model on time series X, one builds a regression model on X where residual time-series N of the regression model follows a SARIMA or an ARIMA model. The first part of the REG-ARIMA model is formulated as in Equation 4.2 and N is the remaining time series on which a SARIMA model will be built.

$$X = b_0 + b_1 * x_1 + b_2 * x_2 + ... + b_k * x_k + N \tag{4.2}$$

4.2.4 LSTM Models for Forecasting

Long Short-Term Memory (LSTM) networks are used to predict sequence problems and they are a type of recurrent neural network (RNN). RNN is a recursive neural network approach that can model dynamic performance of systems so it can be used to model sequential data as in Figure 4.1 adapted from [43]. However, long-range dependencies cannot be captured by RNN, RNNs are only capable of process short-term sequential data. LSTM is a type of recurrent neural networks and it can capture long-term information by using memory cell structure like a conveyor belt. LSTM models applied on time sequences are presented in [43].

4.2.5 REG-LSTM for Forecasting

REG-LSTM is a combination of a regression and an Long-Short Term Memory (LSTM) model. To build a REG-LSTM model on time series X, one builds a regression model on X where residual time-series N of the regression model follows a LSTM. The first part of the REG-LSTM model is formulated as in Equation 4.3 and L is the remaining time series on which a LSTM model will be built.

$$X = b_0 + b_1 * x_1 + b_2 * x_2 + ... + b_k * x_k + L \tag{4.3}$$

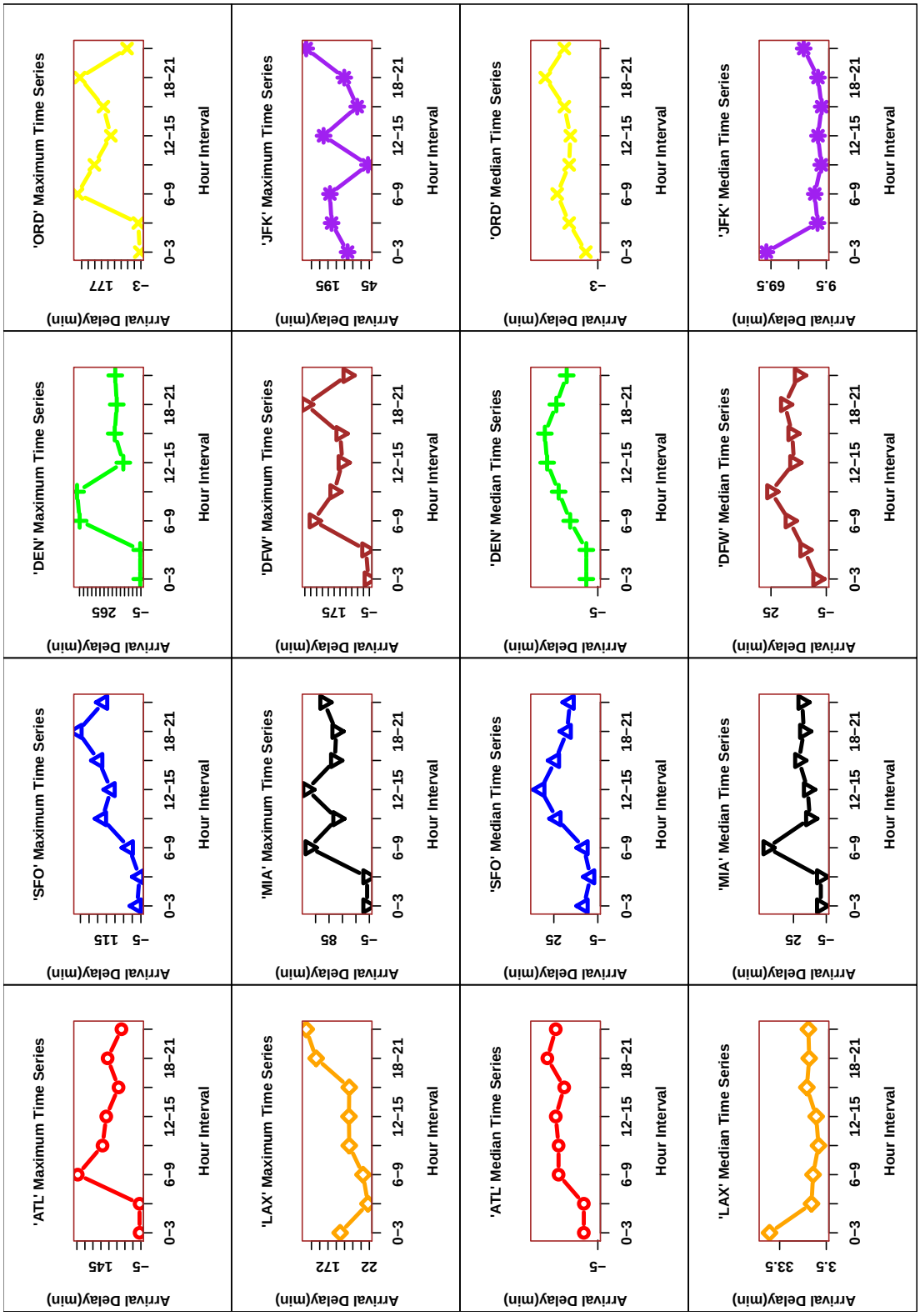

Figure 4.2: Example time series of maximum and median arrival delays for a day

Chapter 5

Exploratory Clustered Forecasting Models

Our approach, Exploratory Clustered Forecasting Modeling (ECFM), clusters the airports/loops and builds a common REG-M (REG-ARIMA or REG-LSTM) model for the aggregate time series of flight delays/traffic speeds for each cluster. Figure 5.1 shows general the steps of ECFM. It first constructs the network, consisting of airports/loops as the nodes and their relations as edges, and extracts the node features for each node. ECFM then clusters the airports/loops (via k-means, PAM) using the node features, node2vec features, and time series patterns of nodes or it basically partition the graph in order to compose sets of nodes.

ECFM applies the clustered modeling using REG-M (or SARIMA) forecasting model. A common regressors set is generated for each cluster. We use the each graph-based feature as regressor in regression model or we use SVDs or PCAs of node2vec features of nodes as regressors. While generating a common time series of a cluster for each time point t_i we find the maximum (median) time series in that cluster and assign the value of time point t_i of maximum (median) time series to the time point t_i of the common time series. To determine the ith value of regressor r which is the corresponding regressor of feature f, we use ith value of feature that belongs to selected maximum (median) time series. A

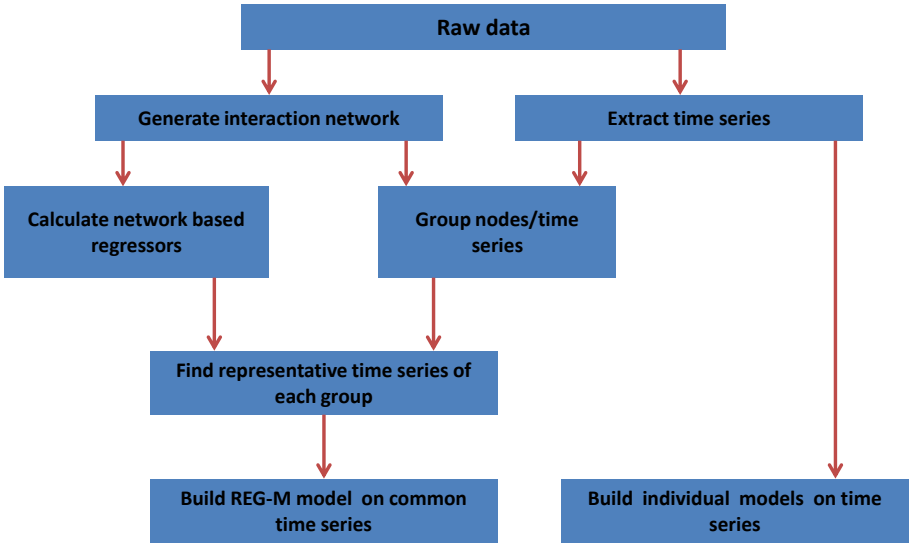

Figure 5.1: Flowchart of exploratory clustered forecasting modeling

REG-M model is developed based on the common time series and regressors set for each cluster. For each cluster we have a common regression model and use this model to find the residual time series, and we build a SARIMA/LSTM model for each residual. To estimate the future values, we use the predicted values of the common regression model and the predicted values of the specific residual time series' SARIMA/LSTM model. As a baseline, we also build a SARIMA/LSTM model for each airport's time series individually.

5.1 Data Scenarios

In this thesis, we propose ECFM as conceptually. We apply this modeling approach on two different datasets in transportation domain. It can be applied for datasets of different domains in which we can incorporate network information into the estimation model. We explain how we construct ECFM on flight delay estimation and traffic speed estimation cases as follows.

5.1.1 Flight Delay Case

To generate the arrival delay time series for each airport, we divide a day into periods (i.e., eight three-hour periods for our experiments) and use the delayed flights in a specific period to calculate the corresponding value. The time point value of the (three-hour) period in the corresponding airport's time series is the maximum (or median) of delays. The signal features of airports' time series are also explored in clustering, besides the graph based features, node2vec features, graph partitioning methods.

The arrival delay of a flight is the amount of time of being late to its destination. A practical task is to build a model that can forecast whether and how much a flight will be delayed. In this thesis, we propose a methodology to utilize the airport network information in forecasting the flight delays. We illustrate our approach using the flight information of 305 US airports for 7 years, collected from Research and Innovative Technology Administration (RITA), absorbed into OST-R. The data set includes the records of millions of commercial domestic flights, each with a set of attributes, e.g., the year, month, day of month, flight number, origin, destination, scheduled time, arrival delay.

The flight delay of an airport is represented by an aggregate time series of all its flights' delays. We develop models, based on regression with ARIMA errors or regression with LSTM errors, that are built on groups of airport time series, as opposed to modeling each airport individually. Our intuition is that the underlying causes of delays can be similar for the airports that have similar features or similar delay patterns. By clustering the delay time series, the model of each airport that might suffer from sparseness or outliers can be enriched with data of other airports. Hence, we use the airport interaction network and the similarities of the airports' delay patterns to cluster the airports and develop a joint representative model for each group of airports.

5.1.2 Traffic Speed Case

In order to generate ECFM on traffic forecasting space we have speeds that are recorded every 5 minutes. We will have 288 observations for a single day. Time series are generated monthly with the length of 8064 and data are collected for 4 months. The data is collected from 207 sensors, so we will have 207 nodes in the graph. In order to create interaction network we use distances between sensors.

The traffic speed of a sensor is represented by a time series and we develop clustered models, based on regression with ARIMA errors or regression with LSTM errors, that are built on groups of sensor time series, as opposed to modeling each airport individually. Our intuition is similar as in the case of flight delay estimation that the underlying causes of traffics can be similar for the sensor that have similar features or similar traffic patterns. By clustering the traffic time series, the model of each sensor that might suffer from sparseness or outliers can be enriched with data of other sensor. Hence, we use the sensor interaction network and the similarities of the sensors' traffic patterns to cluster the sensors and develop a joint representative model for each group of sensors.

5.2 GTC: Graph-Theoretic Clustering

Considering clustering part of the ECFM we use graph-theoretic features of network for GTC version. The graph-based features that we explore are: hub score, betweenness centrality, articulation point, in-degree and weighted in-degree. These features are fed to a clustering algorithm to obtain the node clusters. Beside clustering we also use these graph-theoretic features as regressors in order to build REG-M(REG-ARIMA or REG-LSTM) models. We refer this approach as graph-theoretic clustered SARIMA modeling (GTC-SM), graph-theoretic clustered REG-ARIMA modeling (GTC-RAM), graph-theoretic clustered REG-LSTM modeling (GTC-RLSTMM) in our performance evaluation.

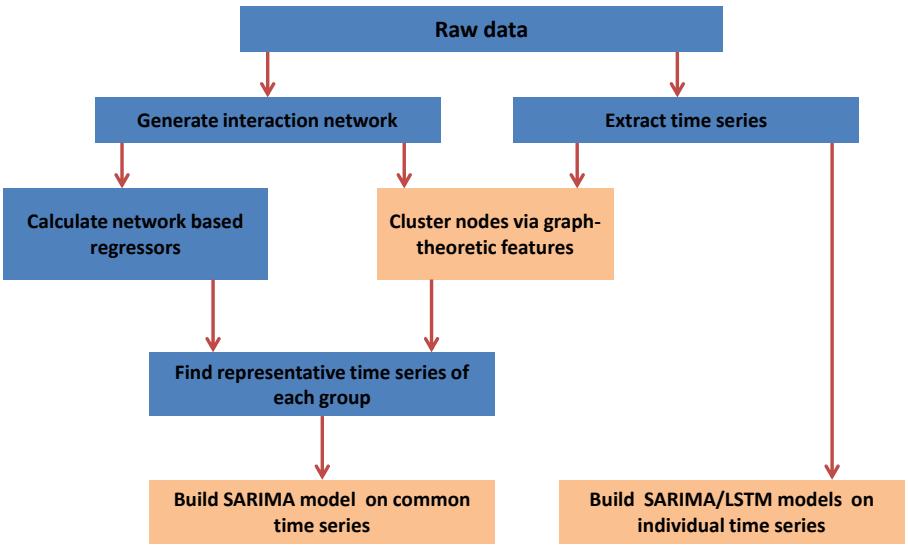

Figure 5.2: Flowchart of proposed GTC-SM

5.2.1 GTC-SM: SARIMA Modeling

As illustrated in Figure 5.2 for GTC-SM we cluster nodes of network via graph-theoretic features and we build SARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.2.2 GTC-RAM: REG-ARIMA Modeling

Figure 5.3 visualize GTC-RAM for which we cluster nodes of network via graph-theoretic features and we build REG-ARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

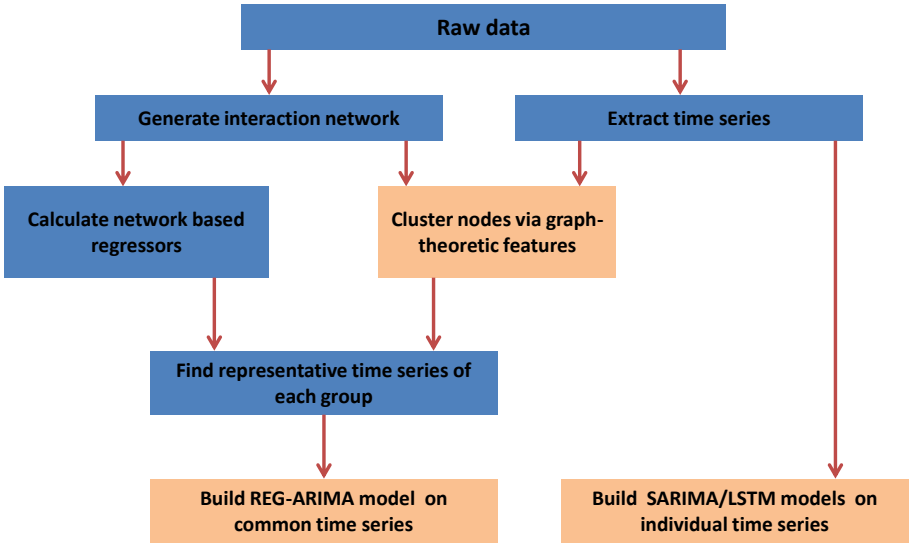

Figure 5.3: Flowchart of proposed GTC-RAM

5.2.3 GTC-RLSTMM: REG-LSTM Modeling

As illustrated in Figure 5.4 for GTC-RLSTMM we cluster nodes of network via graph-theoretic features and we build REG-LSTM estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.3 GP: Graph Partitioning

Graph partitioning can also be used to group the nodes of network [36]. A partition in a network can be defined as a set of nodes with dense connections internally and sparser connections to outside of the partition. We identify partitions of nodes and treat each group of node as a hard partition. Several methods have been developed especially in the social network literature for partitioning and community

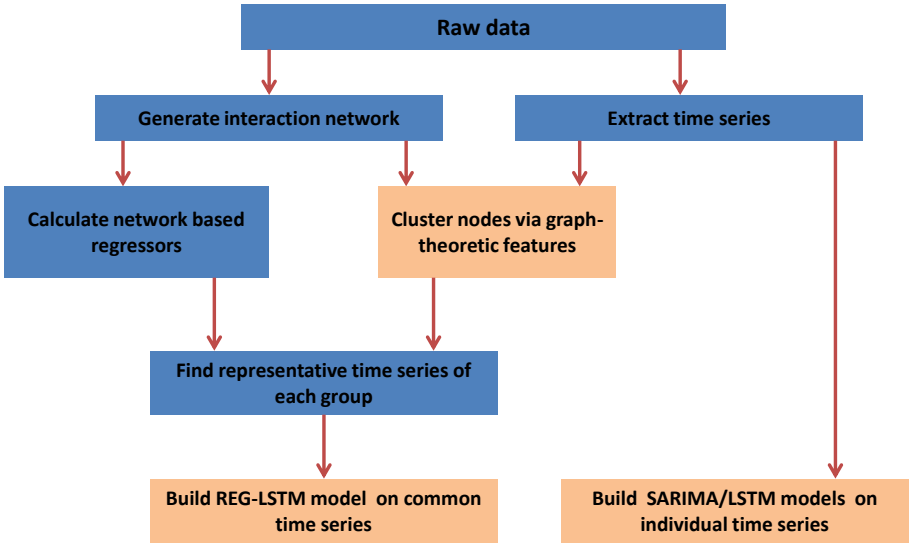

Figure 5.4: Flowchart of proposed GTC-RLSTMM

detection, such as edge betweenness community [37], walk trap community [38], spin glass community [39], leading eigenvector community [36] fast greedy community [40]. Partitioning algorithms show similar performance in our case so we select the walk trap community algorithm. We refer this approach as graph partitioned SARIMA modeling (GP-SM), graph partitioned REG-ARIMA modeling (GP-RAM), and graph partitioned REG-LSTM modeling (GP-RLSTMM) in our performance evaluation.

5.3.1 GP-SM: SARIMA Modeling

As illustrated in Figure 5.5 for GP-SM we obtain groups of nodes via partitioning graph and we build SARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

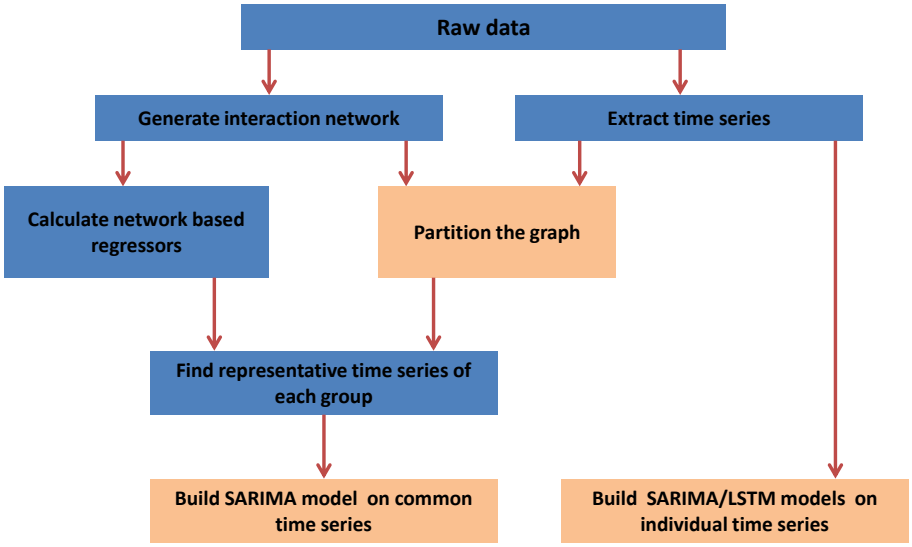

Figure 5.5: Flowchart of proposed GP-SM

5.3.2 GP-RAM: REG-ARIMA Modeling

Figure 5.6 visualize GP-RAM for which we obtain groups of nodes by partitioning graph and we build REG-ARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.3.3 GP-RLSTMM: REG-LSTM Modeling

As illustrated in Figure 5.7 for GP-RLSTMM we obtain clusters of nodes by partitioning graph and we build REG-LSTM estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

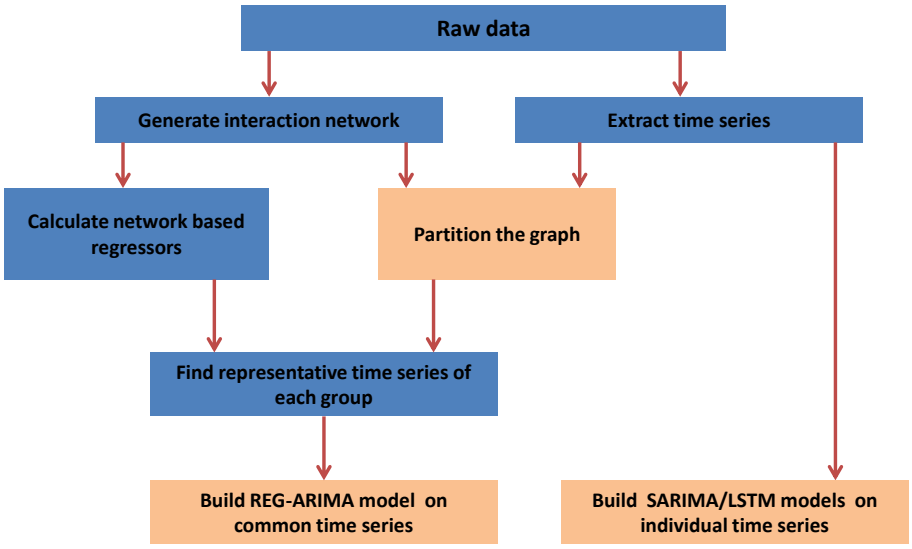

Figure 5.6: Flowchart of proposed GP-RAM

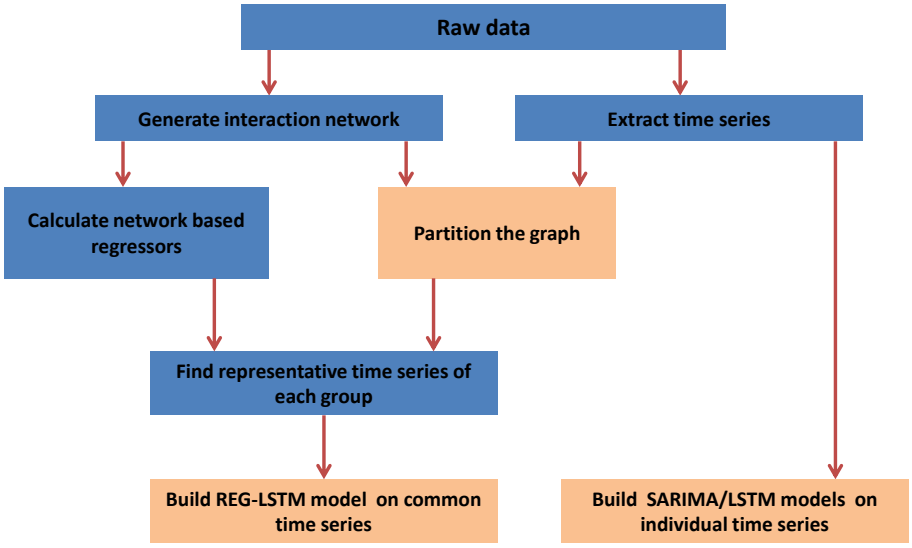

Figure 5.7: Flowchart of proposed GP-RLSTMM

5.4 TSC: Time Series Clustering

Another approach we explore for clustering is to utilize signal information of nodes' delay or traffic speed time series. We extract features using time series transformation methods, namely Discrete Fourier Transform (DFT) [41] and Discrete Wavelet Transform (DWT) [42]. We call these approaches time series clustered SARIMA modeling (TSC-SM), time series clustered REG-ARIMA modeling (TSC-RAM), and time series clustered REG-LSTM modeling (TSC-RLSTMM) in our performance evaluation.

5.4.1 TSC-SM: SARIMA Modeling

As illustrated in Figure 5.8 for TSC-SM we cluster nodes of network by using features extracted from DFT/DWT of time series. We build SARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.4.2 TSC-RAM: REG-ARIMA Modeling

Figure 5.9 visualize TSC-RAM for which we obtain groups of nodes by clustering nodes features obtained from DFT/DWT of time series. We build REG-ARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.4.3 TSC-RLSTMM: REG-LSTM Modeling

As illustrated in Figure 5.10 for TSC-RLSTMM we obtain groups of nodes by clustering nodes features obtained from DFT/DWT of time series we build REG-LSTM estimation model on aggregate time series of each cluster, where individual

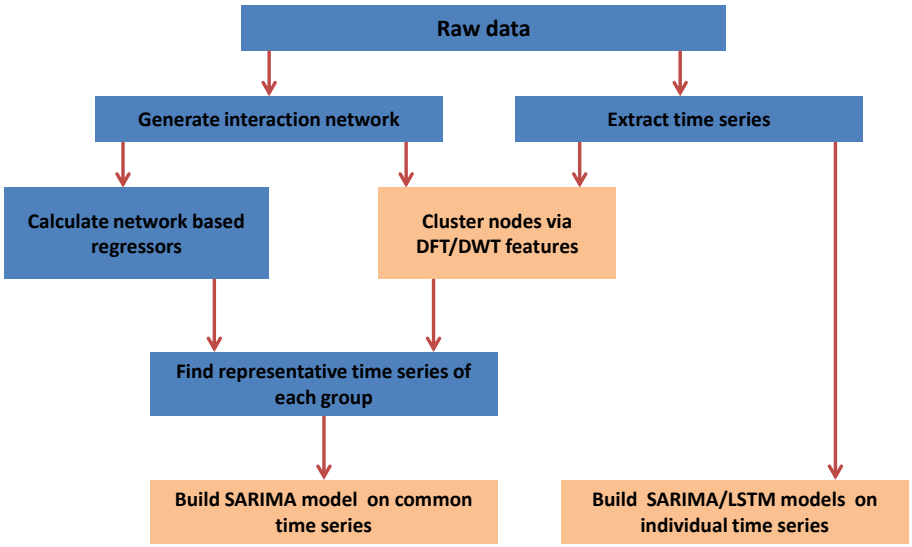

Figure 5.8: Flowchart of proposed TSC-SM

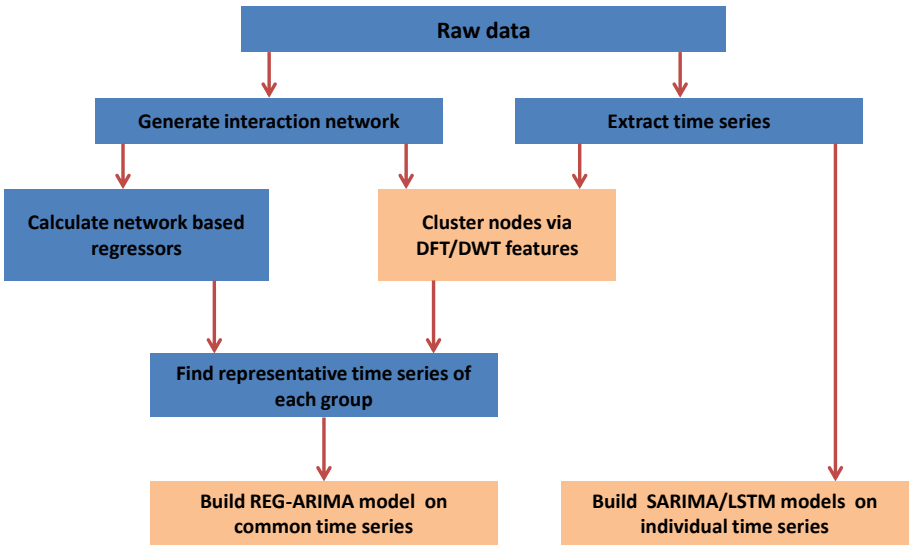

Figure 5.9: Flowchart of proposed TSC-RAM

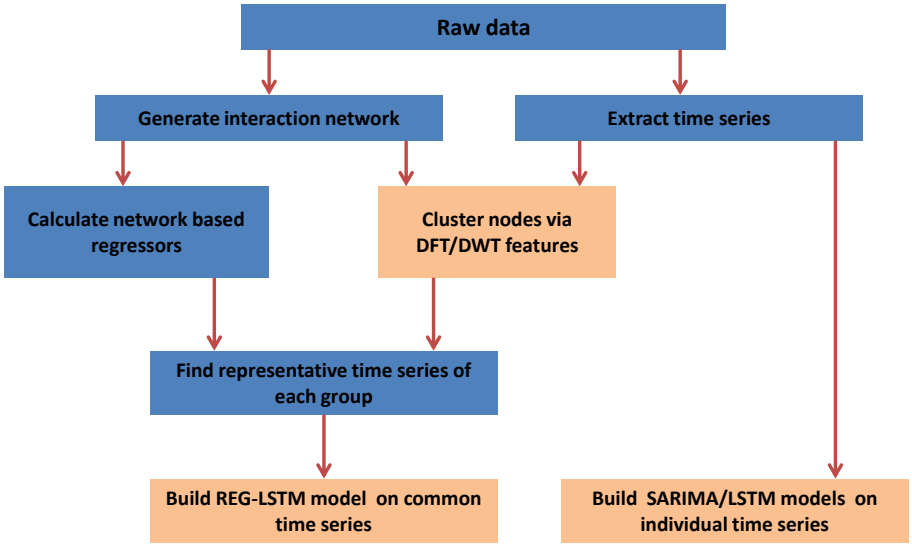

Figure 5.10: Flowchart of proposed TSC-RLSTMM

SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.5 N2VC: Node2Vec Clustering

In this thesis, we process also graphs of airports or sensors by calculating scalable features. For the N2VC version of ECFM we cluster nodes of network by using feature vectors coming from node2vec calculation [1]. While determining regressors for REG-ARIMA or REG-LSTM models we have two alternatives. For the first alternative, we use again graph-theoretic features as regressors for regression models built on aggregate time series of clusters created by using node2vec features. In second alternative, regressors are selected from dimension node2vec clustered SARIMA modeling (N2VC-SM), node2vec clustered REG-ARIMA modeling with graph theoretic regressors (N2VC-RAM-GTR), and node2vec clustered REG-LSTM modeling with graph theoretic regressors (N2VC-RLSTMM-GTR), node2vec clustered REG-ARIMA modeling with dimension reduction regressors

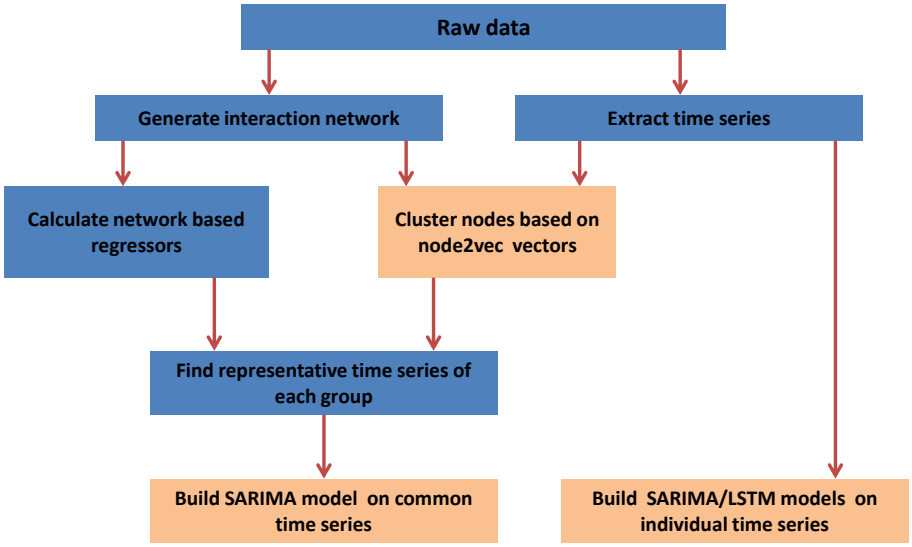

Figure 5.11: Flowchart of proposed N2VC-SM

(N2VC-RAM-DRR), and node2vec clustered REG-LSTM modeling with dimension reduction regressors (N2VC-RLSTMM-DRR) in our performance evaluation.

5.5.1 N2VC-SM: SARIMA Modeling

As illustrated in Figure 5.11 for N2VC-SM we cluster nodes of network by using vectors coming from node2vec [1] calculations. We build SARIMA estimation model on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

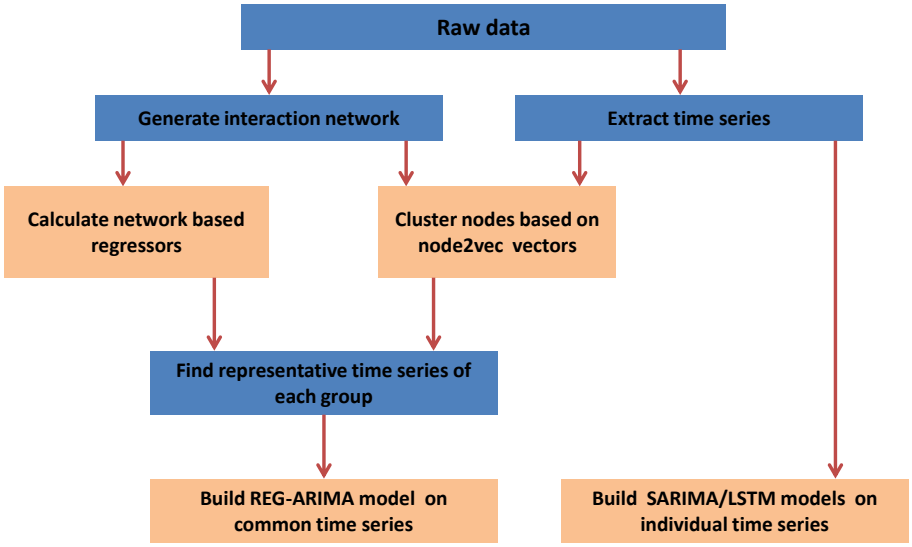

Figure 5.12: Flowchart of proposed N2VC-RAM-GTR

5.5.2 N2VC-RAM-GTR: REG-ARIMA Modeling with Graph Theoretic Regressors

Figure 5.12 visualize N2VC-RAM-GTR for which we obtain groups of nodes by clustering nodes' vectors obtained from node2vec calculations of nodes. We build REG-ARIMA estimation model, in which graph theoretic features used as regressors, on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.5.3 N2VC-RLSTMM-GTR: REG-LSTM Modeling with Graph Theoretic Regressors

As illustrated in Figure 5.13 for N2VC-RLSTMM-GTR we obtain groups of nodes by clustering nodes' vectors obtained from node2vec calculations of nodes and we build REG-LSTM estimation model, in which graph theoretic features used as

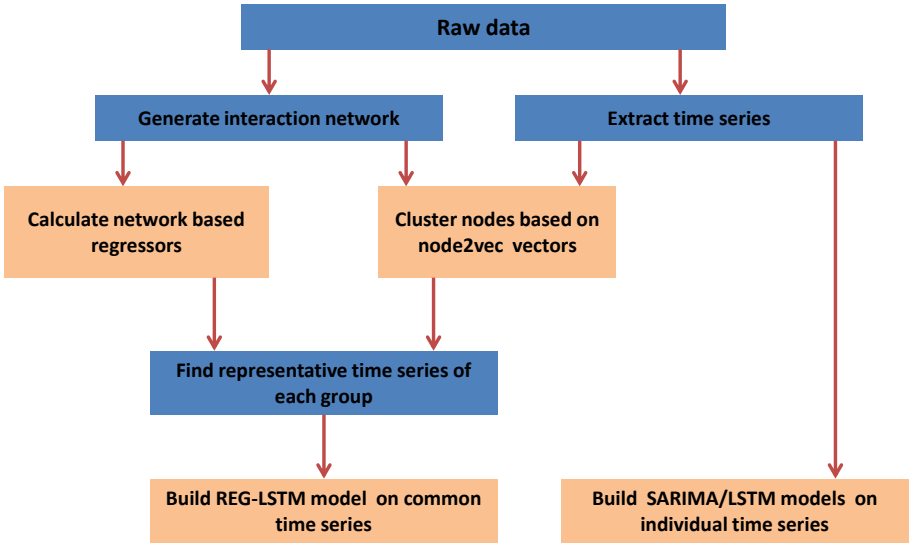

Figure 5.13: Flowchart of proposed N2VC-RLSTMM-GTR

regressors, on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

5.5.4 N2VC-RAM-DRR: REG-ARIMA Modeling with Dimension Reduction Regressor

Figure 5.14 visualize N2VC-RAM-DRR for which we obtain groups of nodes by clustering nodes' vectors obtained from node2vec calculations of nodes. We build REG-ARIMA estimation model, in which SVD or PCA form of node vectors used as regressors, on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

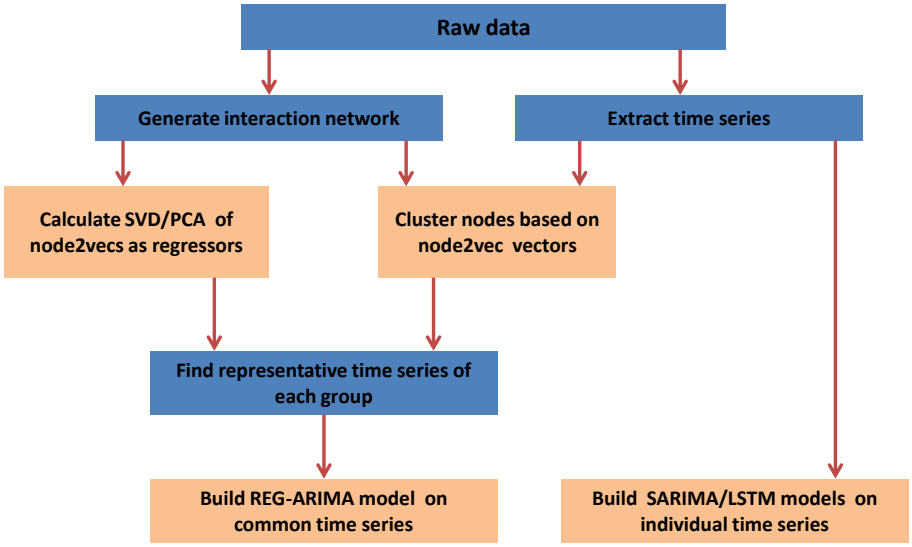

Figure 5.14: Flowchart of proposed N2VC-RAM-DRR

5.5.5 N2VC-RLSTMM-DRR: REG-LSTM Modeling with Dimension Reduction Regressor

As illustrated in Figure 5.15 for N2VC-RLSTMM-DRR we obtain groups of nodes by clustering nodes' vectors obtained from node2vec calculations of nodes and we build REG-LSTM estimation model, in which SVD or PCA form of node vectors used as regressors, on aggregate time series of each cluster, where individual SARIMA modeling and individual LSTM modeling are regarded as baseline.

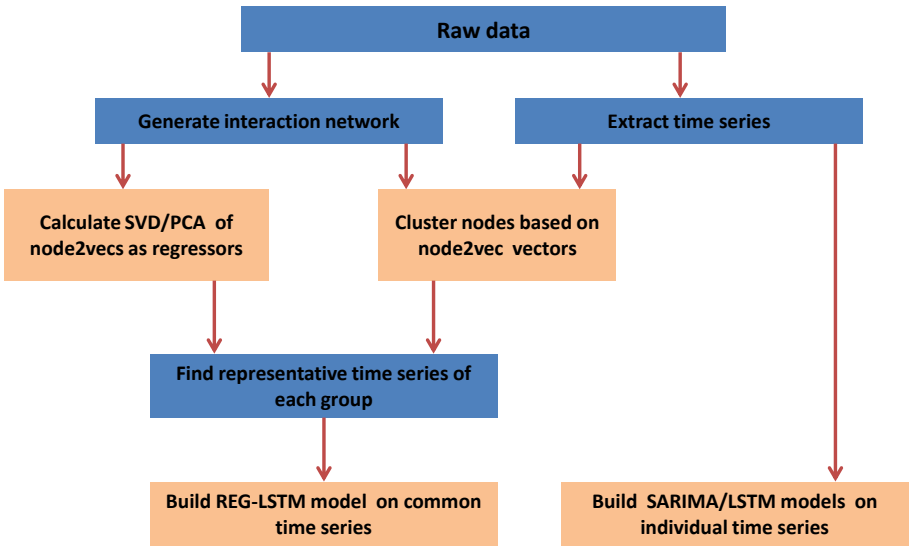

Figure 5.15: Flowchart of proposed N2VC-RLSTMM-DRR

Chapter 6

Experimental Evaluation

We evaluated how the proposed exploratory clustered methodology change the accuracy of forecasting the delays and traffics, compared to modeling each node's time series individually. We compare the different variants of incorporating the clustered network information with the baseline of fitting an individual model for each time series of transportation variable.

6.1 Datasets

We applied proposed idea on two different datasets. While comparing variants proposed clustered forecasting modeling approach with individual baselines we tried to estimate flight delay of airports of US and to estimate traffic speed of sensors located at some point in Los Angeles.

6.1.1 Flight Delay Dataset

We used the data set provided by RITA (Research and Innovative Technology Administration), absorbed into OST-R, that contains 7 years of flight records in

the United States for the years 2006 to 2012. The data include attributes such as origin, destination, arrival time, scheduled arrival time, etc. RITA coordinates the U.S. Department of Transportation research programs.

We constructed the network of the 305 airports in the data set, and generated the flight arrival delay time series of each airport. Note that the arrival delay is defined as the difference between the scheduled arrival time and the actual arrival time, both in local time. The forecasting methods are implemented to predict the results for three-hour periods. A delay time-series of length 2920 for each year is used for each airport. We took the first 2680 time points to build the models and made the forecasts for the remaining 240 points. We present the accuracy results for every week (4 weeks for 240 points) and compare the accuracy performances using the measures of Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) in Equation 6.1, and Equation 6.2 respectively.

6.1.2 Los Angeles Traffic Speed Dataset

In this dataset, we have speeds that are recorded every 5 minutes. We will have 288 observations for a single day. Time series are generated monthly with the length of 8064 and data are collected for 4 months. The data is collected from 207 sensors, so we will have 207 nodes in the graph. In order to create interaction network we use distances between sensors. We took the first 6048 time points to construct models and made the forecasts for the remaining 2016 points. We present the accuracy results for every last week (1 week 2016 time points) of month on average and compare the accuracy performances using the measures of Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) in Equation 6.1, and Equation 6.2 respectively.

$$MAPE = \frac{1}{h}\left(\Sigma_{i=1}^{h}|\frac{x_{n+i} - f_i}{x_{n+i}}|\right) \tag{6.1}$$

$$MAE = \frac{1}{h}\left(\Sigma_{i=1}^{h}|x_{n+i} - f_i|\right) \tag{6.2}$$

where h is the forecasting period, x_{n+i} is the i-th future time point, and f_i is the i-th forecast.

6.2 Validation of Approaches using RAM

Table 6.1 shows the correlation of the pairs of graph-based features. Hub score, in-degree, and weighted-in-degree are highly correlated with each other, while betweenness centrality and articulation have less correlation with them and with each other.

Table 6.1: Correlation coefficients between features

	HScore	**Betw.**	**APoint**	**InDegree**	**WInDegree**
HScore	1.000	0.695	0.618	0.953	0.969
Betw.	0.695	1.000	0.601	0.681	0.822
APoint	0.618	0.601	1.000	0.683	0.663
InDegree	0.953	0.681	0.683	1.000	0.948
WInDegree	0.969	0.822	0.663	0.948	1.000

Table 6.2 illustrates the model summaries for the case of the number of clusters around 50. The summaries of the clusters whose regression models are based only on the intercept, and the clusters with size of 1, are not presented in the table. Model number defines 15 out of 50 models. The p-values for all the models are found to be less than 0.05, i.e., all regression models are statistically significant.

6.3 The Number of Clusters

We use k-means [44] and Partitioning Around Medians (PAM) [45] in our experiments. To determine the number of clusters, we utilize the within-cluster sum of squares and silhouette width as quality measures for k-means and PAM, respectively. The plots for the number of clusters vs. the cluster quality are presented in Figure 6.1. We can see an elbow behavior on all plots which can be used in

Table 6.2: Regression models' summaries

Model	P-value	Adjusted R-squared.
1	2.20E-16	0.224
2	2.20E-16	0.250
3	2.20E-16	0.064
4	2.20E-16	0.107
5	2.20E-16	0.146
6	2.20E-16	0.150
7	2.20E-16	0.100
8	2.20E-16	0.300
9	2.20E-16	0.158
10	2.20E-16	0.224
11	4.43E-10	0.030
12	2.20E-16	0.081
13	0.014	0.004

determining the number of clusters. These represent the qualities of the graph-theoretic clustering. The same procedure is applied for the time series clustering in our experiments. We continue with k-means in experimental evaluation.

6.4 Approaches in Comparison

The baseline method, ISM (Individual SARIMA model) fits an individual model to each time series of delays and traffic speeds. We refer to our methods that follow different variants of graph-theoretic clustering, graph partitioning, clustering, node2vec clustering and time series clustering within the proposed methodology as in Table 6.3. We present performance of the variants as stated in Chapter 5.

6.4.1 GTC-SM Results

We compare GTC-SM with individual baseline SARIMA model Figure 6.2. We cannot conclude that GTC-SM is certainly better than ISM.

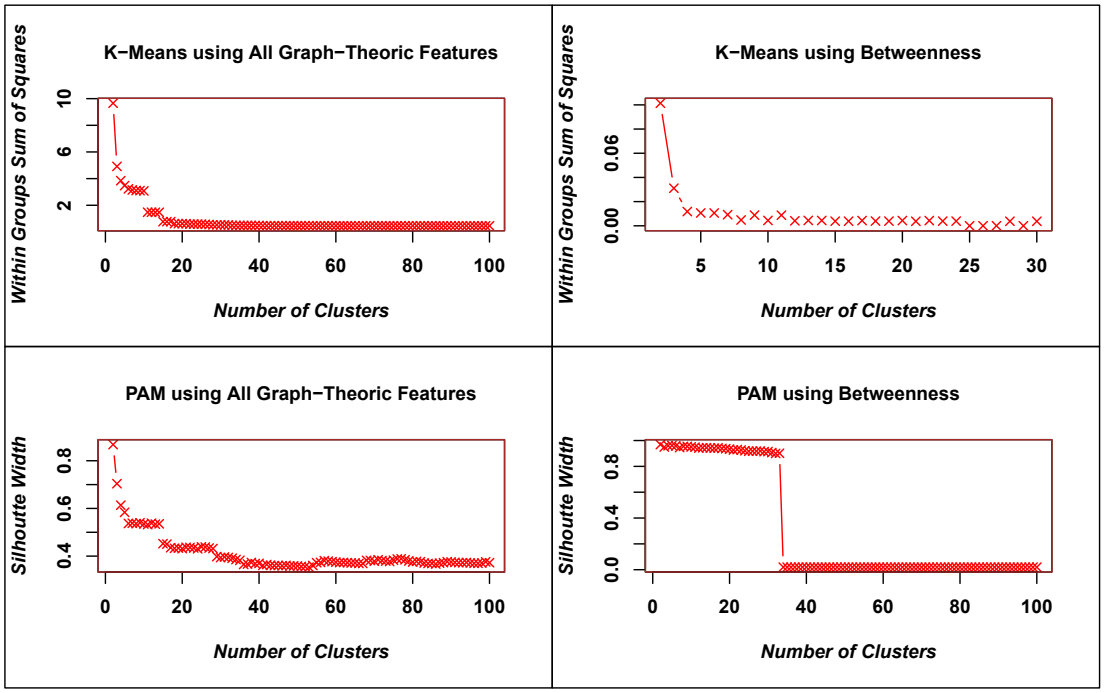

Figure 6.1: Cluster quality behavior changing according to number of clusters

6.4.2 GTC-RAM Results

Results for comparison of GTC-RAM with individual baseline SARIMA model are depicted in Figure 6.3. There is respectable improvement when graph-theoretical features are used for both clustering and as exploratory variables.

6.4.3 GTC-RLSTMM Results

We compare GTC-RLSTMM with individual baseline SARIMA model. Results are showed in Figure 6.4. We can conlude that GTC-RLSTMM beats both baseline individual model and its ARIMA competitor. Improvement made RLSTMM is around 7 compared to ARIMA.

Table 6.3: Approaches in comparison

Variant	Abbreviation	Full Name
1	GTC-SM	Graph-Theoretic Clustered SARIMA Modeling
2	GTC-RAM	Graph-Theoretic Clustered REG-ARIMA Modeling
3	GTC-RLSTMM	Graph-Theoretic Clustered REG-LSTM Modeling
4	GP-SM	Graph Partitionined SARIMA Modeling
5	GP-RAM	Graph Partitionined REG-ARIMA Modeling
6	GP-RLSTMM	Graph Partitionined REG-LSTM Modeling
7	TSC-SM-DFT	Time Series Clustered SARIMA Modeling with Discrete Fourier Transform
8	TSC-RAM-DFT	Time Series Clustered REG-ARIMA Modeling with Discrete Fourier Transform
9	TSC-RLSTMM-DFT	Time Series Clustered REG-LSTM Modeling with Discrete Fourier Transform
10	TSC-SM-DWT	Time Series Clustered SARIMA Modeling with Discrete Wavelet Transform
11	TSC-RAM-DWT	Time Series Clustered REG-ARIMA Modeling with Discrete Wavelet Transform
12	TSC-RLSTMM-DWT	Time Series Clustered REG-LSTM Modeling with Discrete Wavelet Transform
13	N2VC-SM	Node2Vec Clustered SARIMA Modeling
14	N2VC-RAM-GTR	Node2Vec Clustered REG-ARIMA Modeling with Graph Theoretic Regressors
15	N2VC-RLSTMM-GTR	Node2Vec Clustered REG-LSTM Modeling with Graph Theoretic Regressors
16	N2VC-RAM-DRR	Node2Vec Clustered REG-ARIMA Modeling with Dimension Reduction Regressor
17	N2VC-RLSTMM-DRR	Node2Vec Clustered REG-LSTM Modeling with Dimension Reduction Regressor

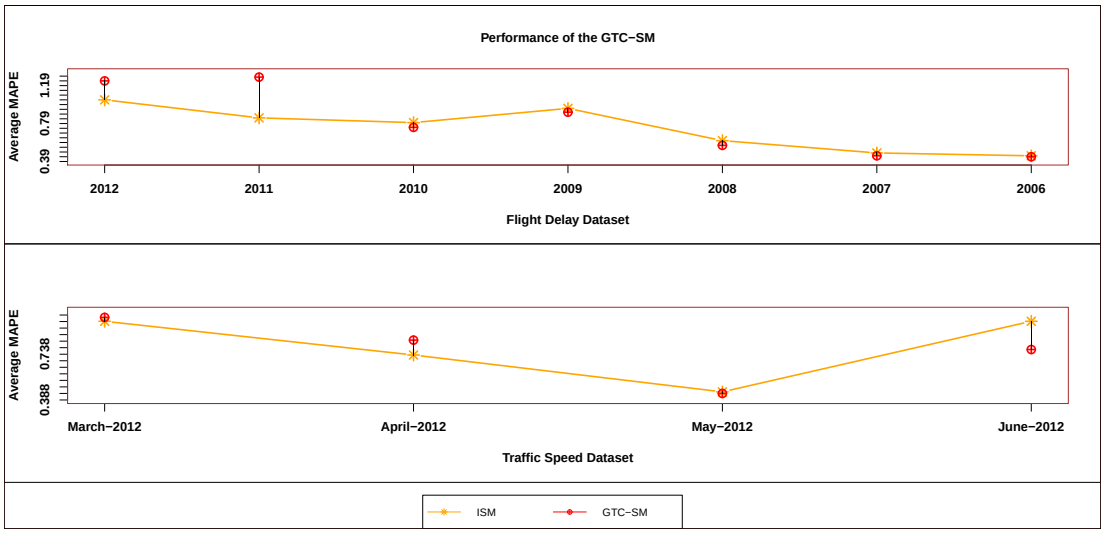

Figure 6.2: Comparison of GTC-SM with individual baseline model

6.4.4 GP-SM Results

We compare GP-SM with individual baseline SARIMA model Figure 6.5. We cannot conclude that GP-SM has better performance than ISM.

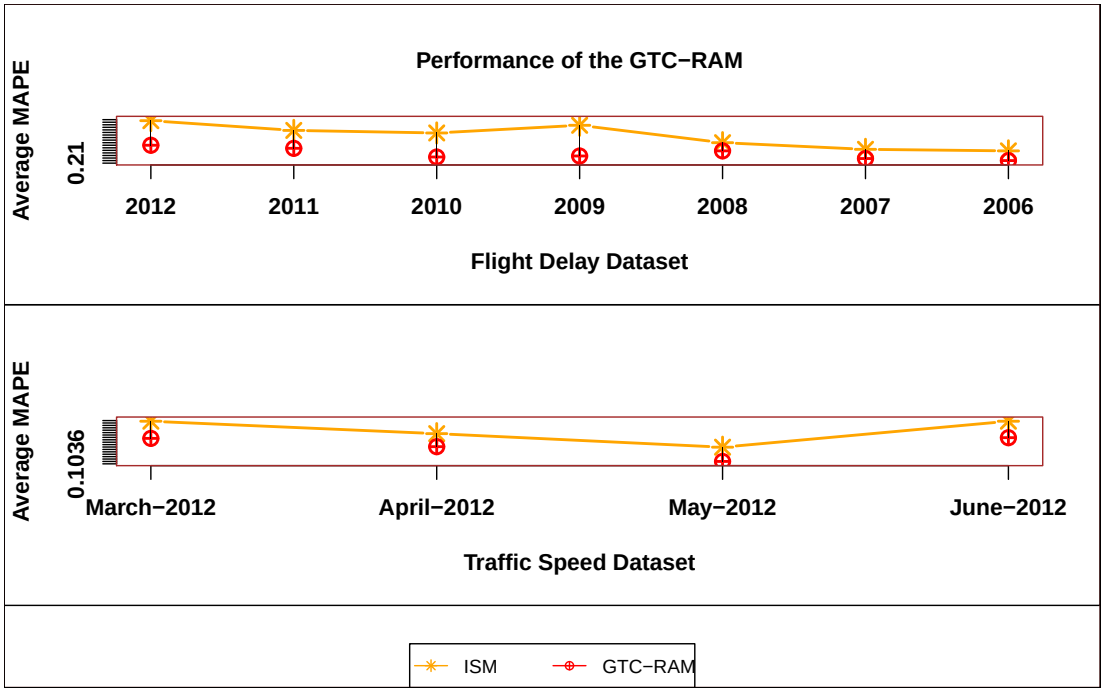

Figure 6.3: Comparison of GTC-RAM with individual baseline model

6.4.5 GP-RAM Results

Results of experiments done with GP-RAM are summarized in Figure 6.6. Although GP-RAM is better than GP-SM, there is no certain insight that GP-RAM beats ISM.

6.4.6 GP-RLSTMM Results

Comparison of GP-RLSTMM with ISM is illustrated in Figure 6.7. Effect of LSTM makes improvement compared to the GP-RAM, it is still not possible to say that GP-RLSTMM is reasonably better than ISM.

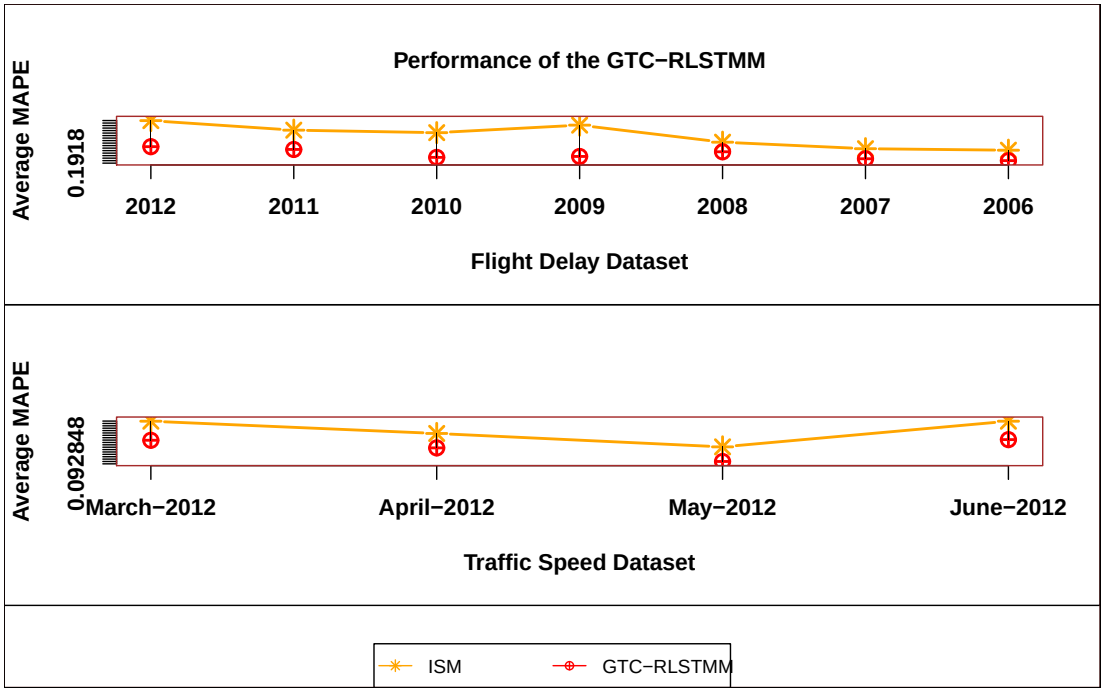

Figure 6.4: Comparison of GTC-RLSTMM with individual baseline model

6.4.7 TSC-SM-DFT Results

We compare TSC-SM-DFT with individual baseline SARIMA model Figure 6.8. We can say that performances of the TSC-SM-DFT and ISM are similar.

6.4.8 TSC-RAM-DFT Results

Figure 6.9 shows the results for comparison of TSC-RAM-DFT and individual baseline SARIMA model. TSC-RAM-DFT is another variant of ECFM that reasonably beats ISM.

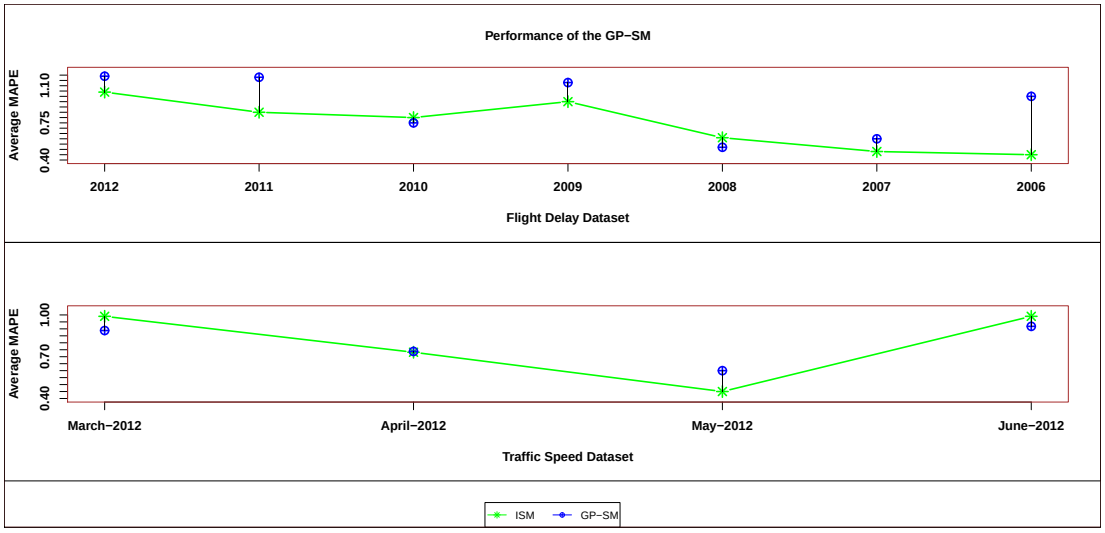

Figure 6.5: Comparison of GP-SM with individual baseline model

6.4.9 TSC-RLSTMM-DFT Results

Using LSTM instead of ARIMA in TSC variant of ECFM makes slightly improvement that is illustrated in Figure 6.10. Thus, we can conclude that TSC-RLSTMM-DFT is better than ISM also.

6.4.10 TSC-SM-DWT Results

We compare TSC-SM-DWT with individual baseline SARIMA model Figure 6.8. We can say that performances of the TSC-SM-DWT and ISM are similar.

6.4.11 TSC-RAM-DWT Results

Figure 6.12 shows the results for comparison of TSC-RAM-DWT and individual baseline SARIMA model. TSC-RAM-DWT is another variant of ECFM that reasonably beats ISM and it is slightly better than TSC-RAM-DFT.

Figure 6.6: Comparison of GP-RAM with individual baseline model

6.4.12 TSC-RLSTMM-DWT Results

Using LSTM instead of ARIMA in variant of ECFM makes slightly improvement that is illustrated in Figure 6.13 compared to TSC-RAM-DWT. Thus, we can conclude that TSC-RLSTMM-DWT is better than ISM also.

6.4.13 N2VC-SM Results

We compare N2VC-SM with individual baseline SARIMA model Figure 6.14. We obtain that N2VC-SM is certainly better than ISM.

6.4.14 N2VC-RAM-GTR Results

N2VC-RAM-GTR performs certainly better than individual baseline model ISM, illustrated in Figure 6.15. We obtain form experimental results that it also has improvement compared to variants such as GTC-RAM, TSC-RAM.

Figure 6.7: Comparison of GP-RLSTMM with individual baseline model

6.4.15 N2VC-RLSTMM-GTR Results

N2VC-RLSTMM-GTR is the best variant of the ECFM according to experimental results. Its performance compared to the ISM is showed in Figure 6.16.

6.4.16 N2VC-RAM-DRR Results

We compare N2VC-RAM-DRR with individual baseline SARIMA model Figure 6.17. We can conclude that N2VC-RAM-DRR is certainly better than ISM and it is the among outstanding variants of ECFM.

6.4.17 N2VC-RLSTMM-DRR Results

N2VC-RLSTMM-DRR is the second variant of ECFM based on accuracy performance. Its comparison with baseline individual SARIMA model is illustrated in Figure 6.18.

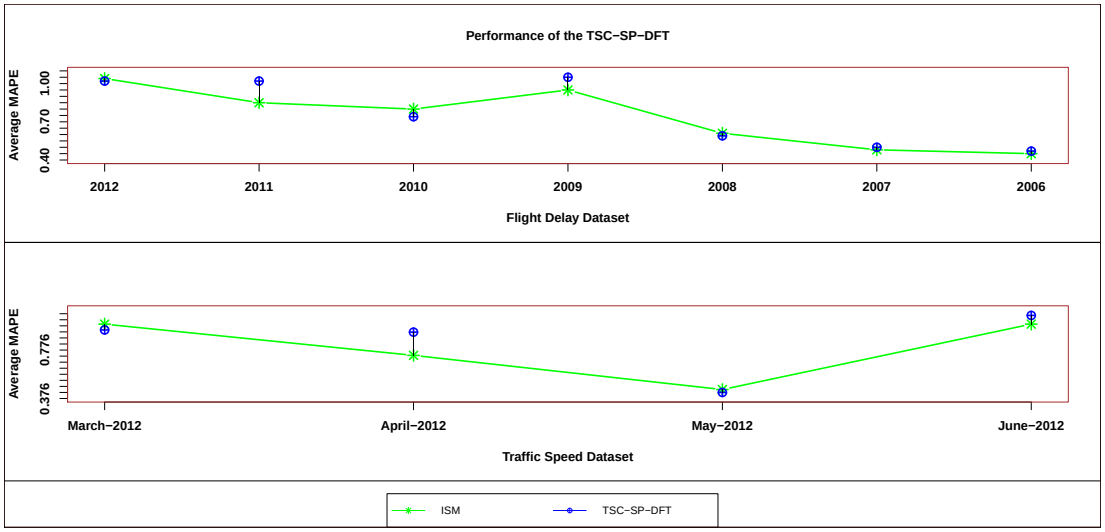

Figure 6.8: Comparison of TSC-SM-DFT with individual baseline model

6.5 Methodology Validation Summary on Flight Delay Dataset

We compare the quality of forecasts of the proposed approaches with those of the baseline ISM. We examine both the performance improvement via different grouping methods (graph-theoretic clustering, graph partitioning, node2vec clustering, time series clustering) and via different time series modeling approaches (SARIMA modeling, REG-ARIMA modeling). We test the performance on maximum and median time series of seven different years' data sets. Maximum time series are composed of maximum delays of each 3-hour slots of days and median time series are created by using median delay values of each 3-hour slots of days. Individual modeling and prediction are done on local times, combining is done according to UTC time. We note that the graph features are utilized both to cluster airports and used as regression variables in REG-M.

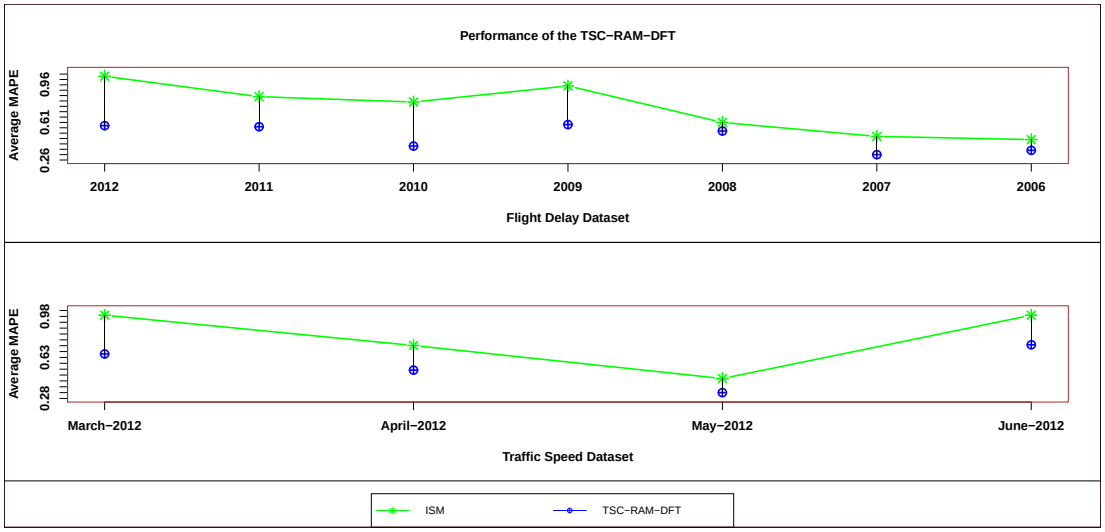

Figure 6.9: Comparison of TSC-RAM-DFT with individual baseline model

6.6 Forecasting Models using all Graph-theoretic Features

Figure 6.19 and 6.20 show the MAPE results where all graph-theoretic features are included in both grouping stage and REG-ARIMA. The yellow star-shaped line represents the baseline ISM. More successful models compared to ISM exist below this yellow star-shaped line.

Experimental results show that the proposed approaches have significant improvements compared to the baseline model ISM. In particular, GTC-RAM, TSDFT-RAM and TSDWT-RAM result in outstanding improvements over ISM. We can summarize improvements of the proposed approaches as follows.

On maximum time series: TSDWT-RAM shows an average of 55% improvement in terms of average MAPE of forecasts over the baseline. The improvements range from 43% to 62%, for the years of 2008 and 2011, respectively. GTC-RAM shows from 5% to 41% improvements, for 2008 and 2010. On the yearly average, GTC-RAM makes a 25% improvement. TSDFT-RAM provides an average of 18% improvement over the baseline in terms of the forecast accuracy.

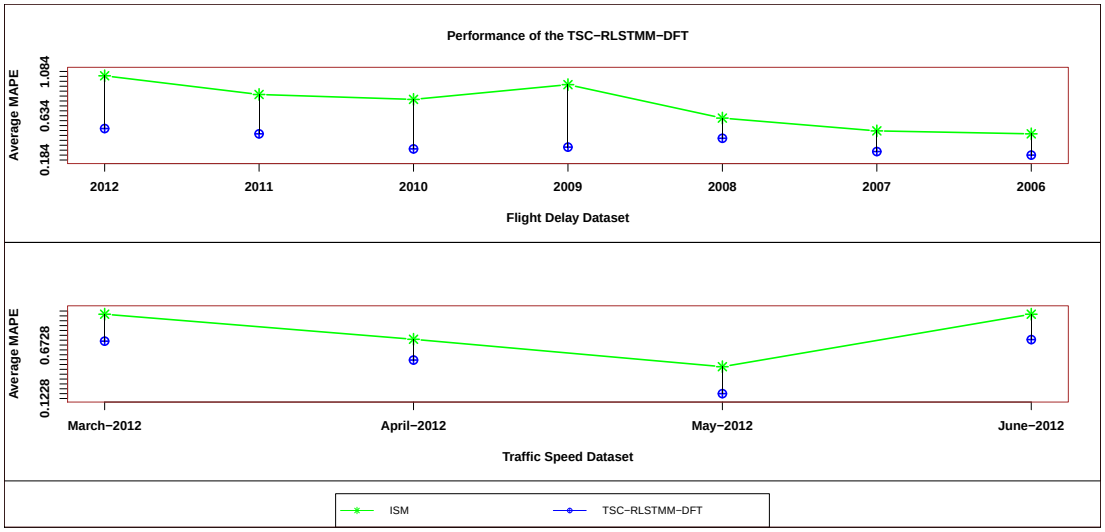

Figure 6.10: Comparison of TSC-RLSTMM-DFT with individual baseline model

On median time series: TSDWT-RAM shows an average of 18% improvement in terms of average MAPE of forecasts over the baseline. The improvements range from 13% to 45%, for the years of 2007 and 2008, respectively. GTC-RAM shows from 14% to 45% improvements, for 2008 and 2010. On the yearly average, GTC-RAM makes a 27% improvement.

Clustering makes the forecasting models more robust to the outliers in the time series. A further improvement is achieved by using a REG-ARIMA model where the graph-based features are used as the regressor variables. For many airports, ISM has a high MAPE that is significantly more than 1. The clustered model reduces the MAPE to values significantly smaller than 1. We have also checked the specific cases where the individual model performs better than the clustered model for an airport. According to clustering, some of the big airports may not belong to any cluster (e.g. ORD) or some small airports (e.g. VLD, CLT) may belong to a cluster on which prediction model performs worse compared to individual modeling. In all of these cases, the MAPE is significantly less than 1 for both types of models.

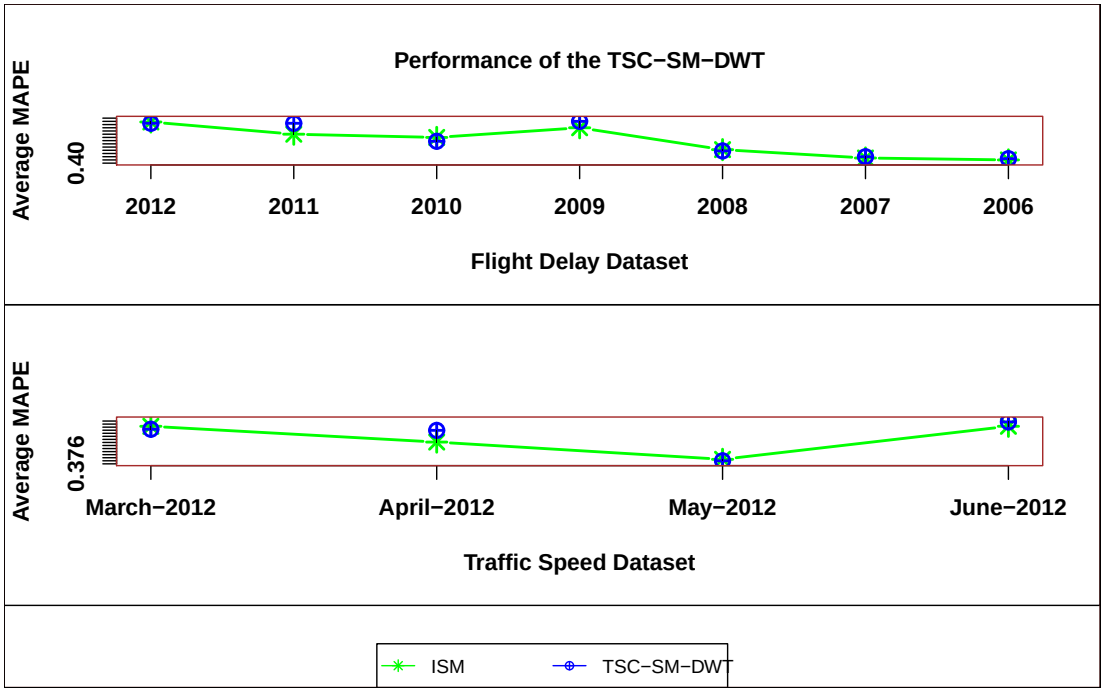

Figure 6.11: Comparison of TSC-SM-DWT with individual baseline model

6.7 Identifying Important Features for Forecasting

We repeat the experiments using a subset of the features, as opposed to using all. This helps us understand which features are the most important for accuracy improvements. We find out that "betweenness centrality" gives the best accuracy result among all cases on this setup. Note that betweenness centrality of a node in the graph measures degree of being the center for shortest paths. A node with higher betweenness centrality may correspond to a transfer center or a hub in the airport network.

Results of the accuracy improvements when the feature subset containing only "betweenness centrality" is used on maximum and median time series are presented in Figure 6.21 and 6.22. Accuracy improvements are illustrated in Figure 6.25 and summarized below.

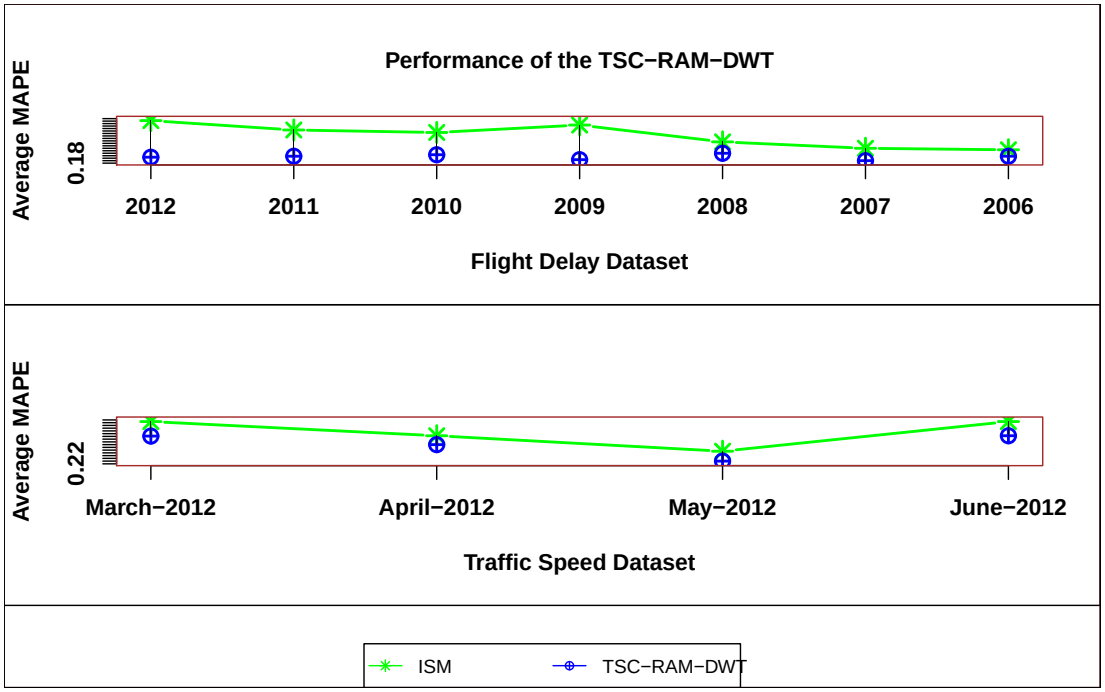

Figure 6.12: Comparison of TSC-RAM-DWT with individual baseline model

On maximum time series: On yearly average, GTC-RAM makes 45% improvement in terms of average MAPE of forecasts over the baseline. The improvements range from 26% to 63%, for the years of 2008 and 2009, respectively. TSDFT-RAM shows from 13% to 51% improvements, for 2008 and 2010. On the yearly average, TSDFT-RAM shows a 33% improvement. TSDWT-RAM provides same level accuracy when only betweenness centrality topological feature is used compared to case where all topological features are used. MAPE of this model ranges from 28% to 73%, and yearly average is 54%.

On median time series: On yearly average, GTC-RAM shows a 28% improvement in terms of average MAPE of forecasts over the baseline. The improvements range from 17% to 57%, for the years of 2008 and 2010, respectively. TSDFT-RAM does not have improvement for years 2006 and 2007, so its yearly average keeping out these years is 25%. Yearly average of TSDWT-RAM is also 25%.

We also evaluated the methods using MAE measure. We present MAE results only for the three top performing methods using MAPE. Accuracy results of these

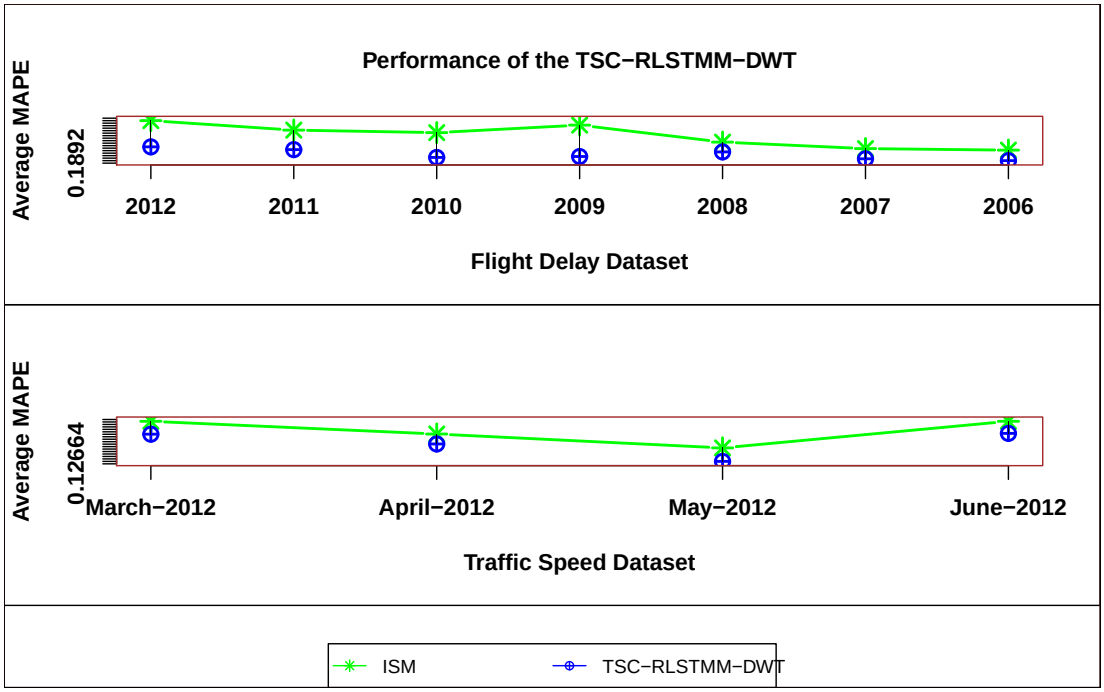

Figure 6.13: Comparison of TSC-RLSTMM-DWT with individual baseline model

methods measured by MAE are shown in Figures 6.23 and Figure 6.24 for the maximum and median time series respectively. The summary of improvements are illustrated in Figure 6.25 The performance behavior of the methods is similar with the evaluation by the MAPE.

Betweenness Centrality (BC) score of an airport is found to be a factor in understanding the delays associated with the airport. The BC does not always have a high correlation with the number of flights. The airports that are central in the paths of potential travel itineraries are vulnerable to further delay. Similarly, most of the *articulation points* of the airport network are found to be among the highest delayed airports. BC and articulation have less correlation with other measures such as the hub score, and with each other. Several airports have highly similar graph based features in the airport network. For example, ATL and ORD are consistently in the same clusters based on the graph centrality measures. This may help to gather more information about their delay patterns using additional data from each other.

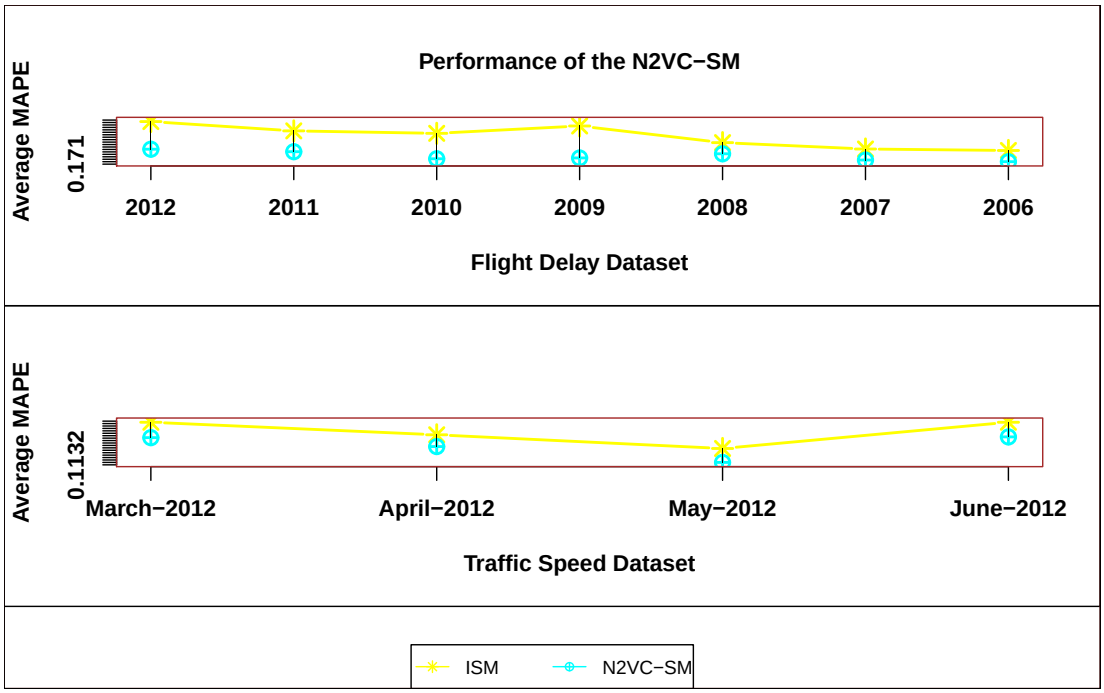

Figure 6.14: Comparison of N2VC-SM with individual baseline model

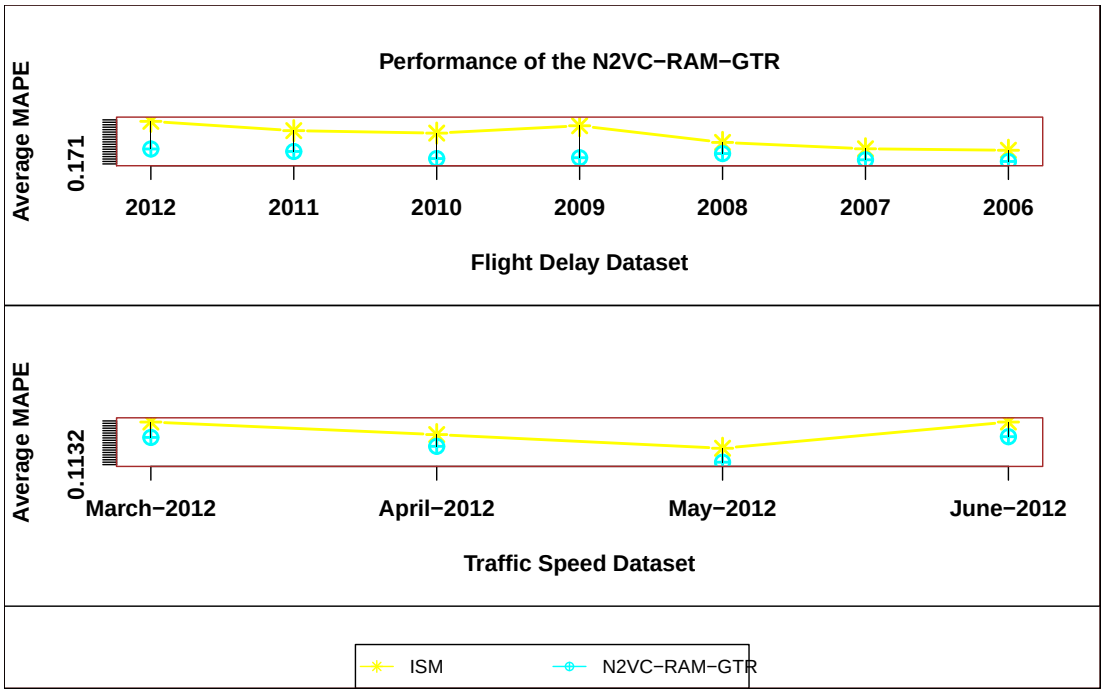

Figure 6.15: Comparison of N2VC-RAM-GTR with individual baseline model

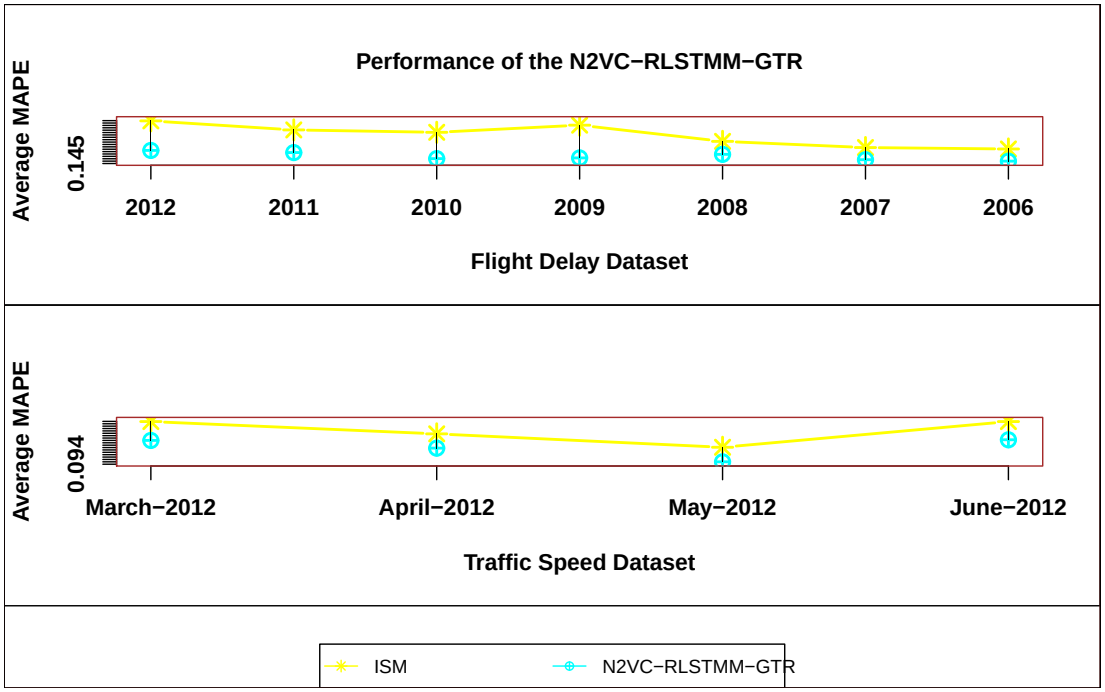

Figure 6.16: Comparison of N2VC-RLSTMM-GTR with individual baseline model

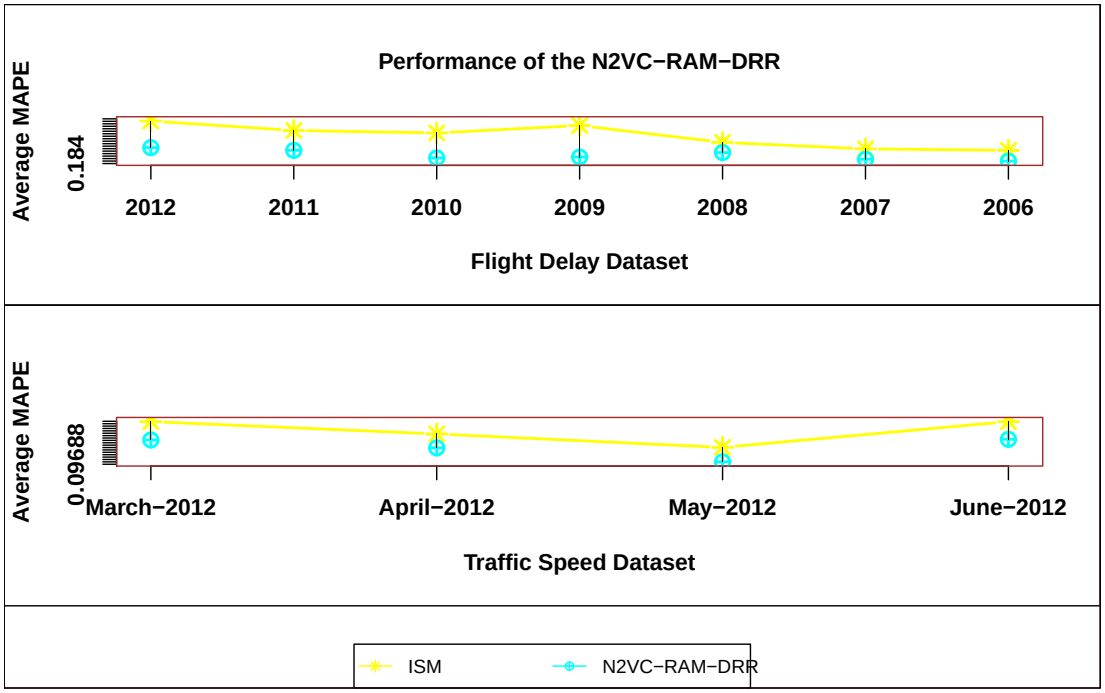

Figure 6.17: Comparison of N2VC-RAM-DRR with individual baseline model

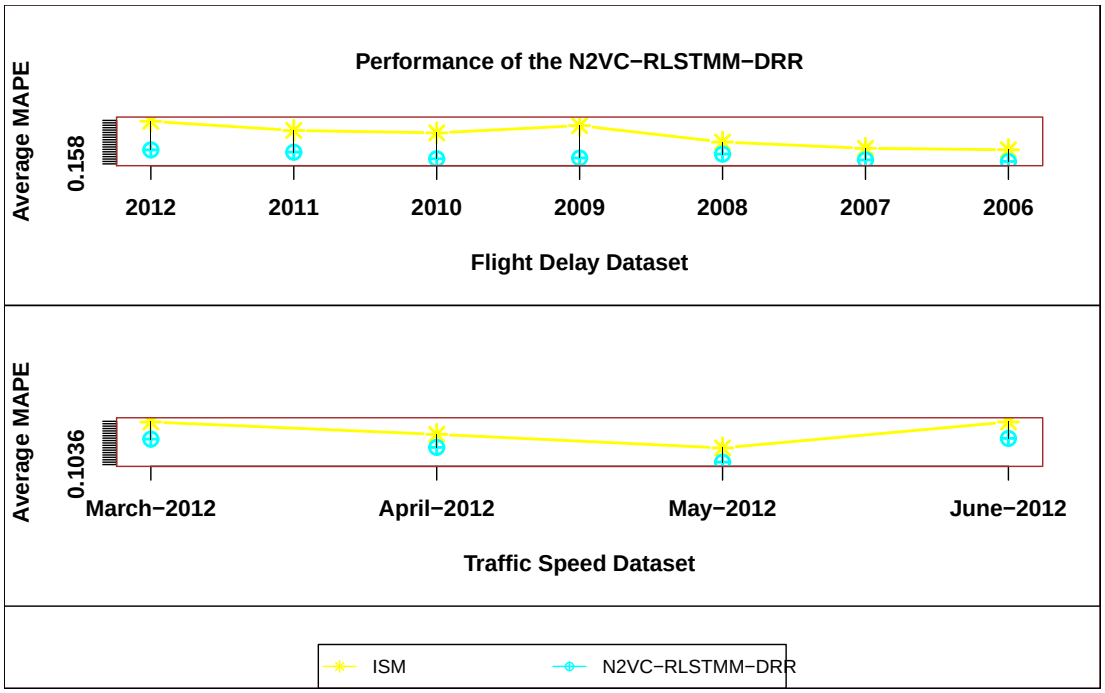

Figure 6.18: Comparison of N2VC-RLSTMM-DRR with individual baseline model

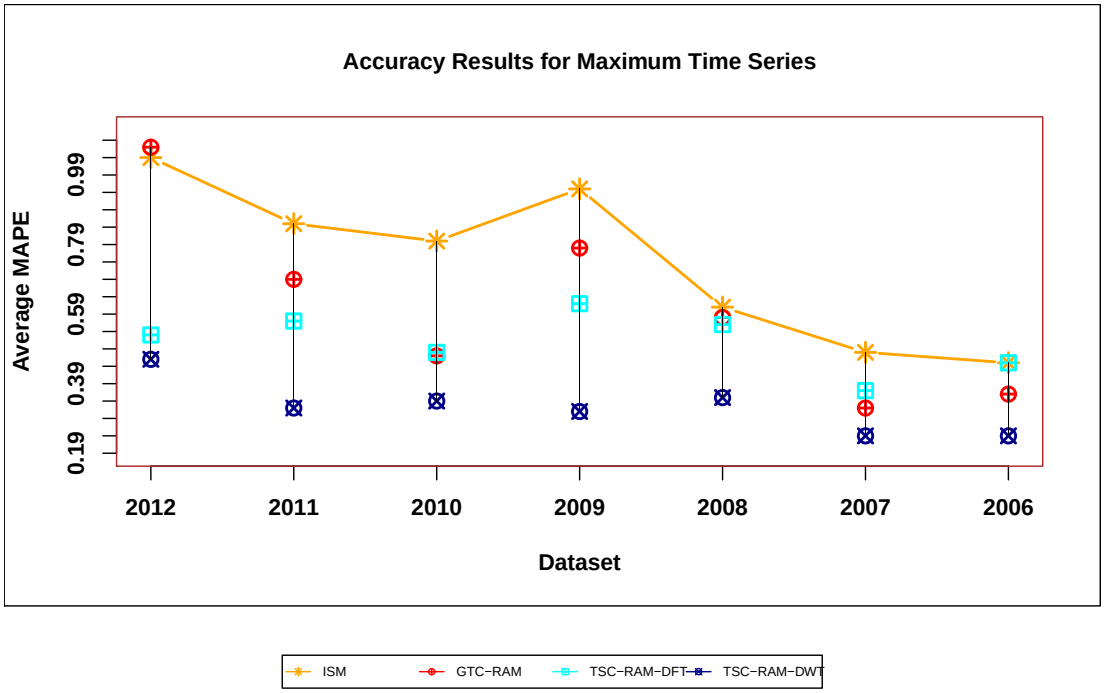

Figure 6.19: Accuracy comparison of proposed approaches using all features for maximum time series

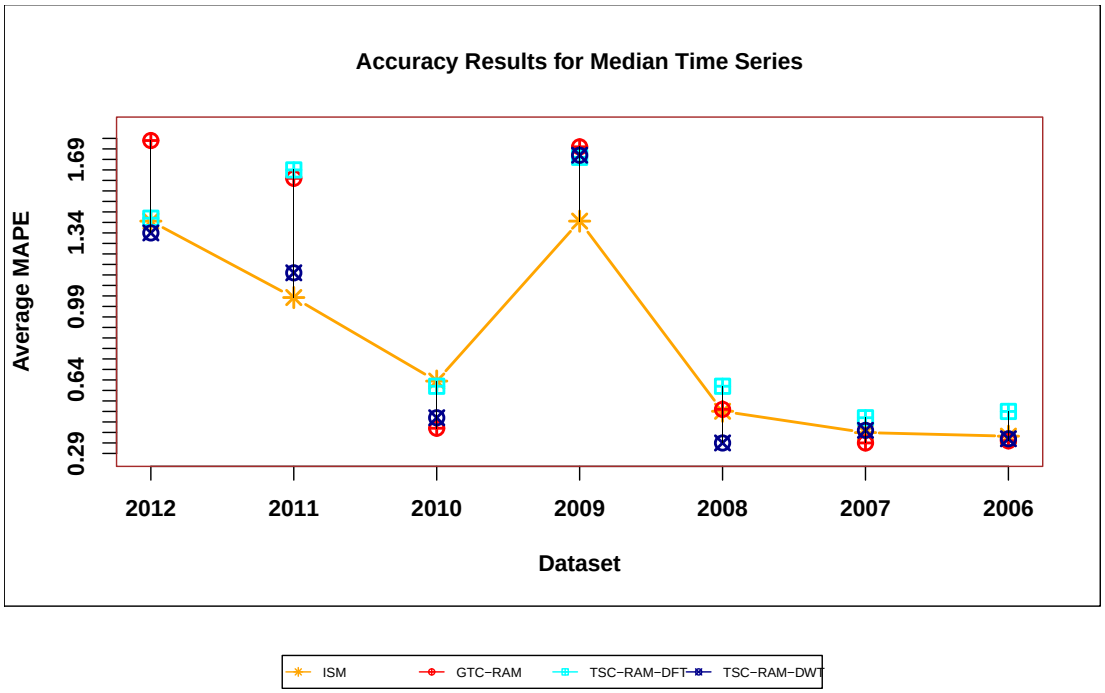

Figure 6.20: Accuracy comparison of proposed approaches using all features for median time series

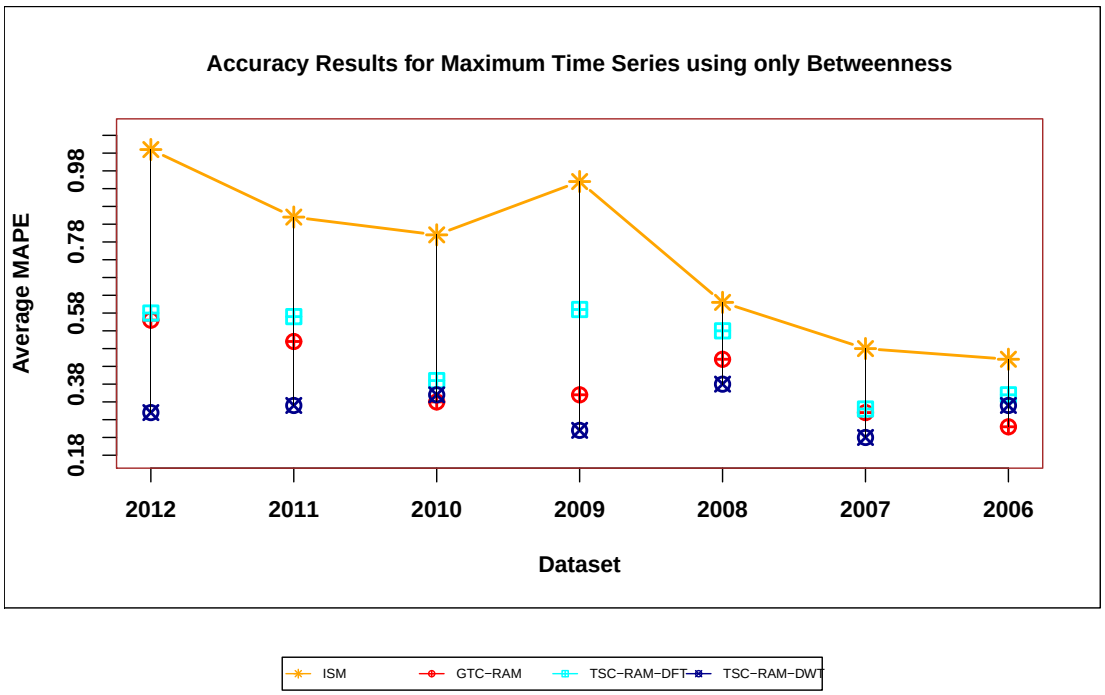

Figure 6.21: Effect of using only betweenness centrality feature on accuracy for maximum time series

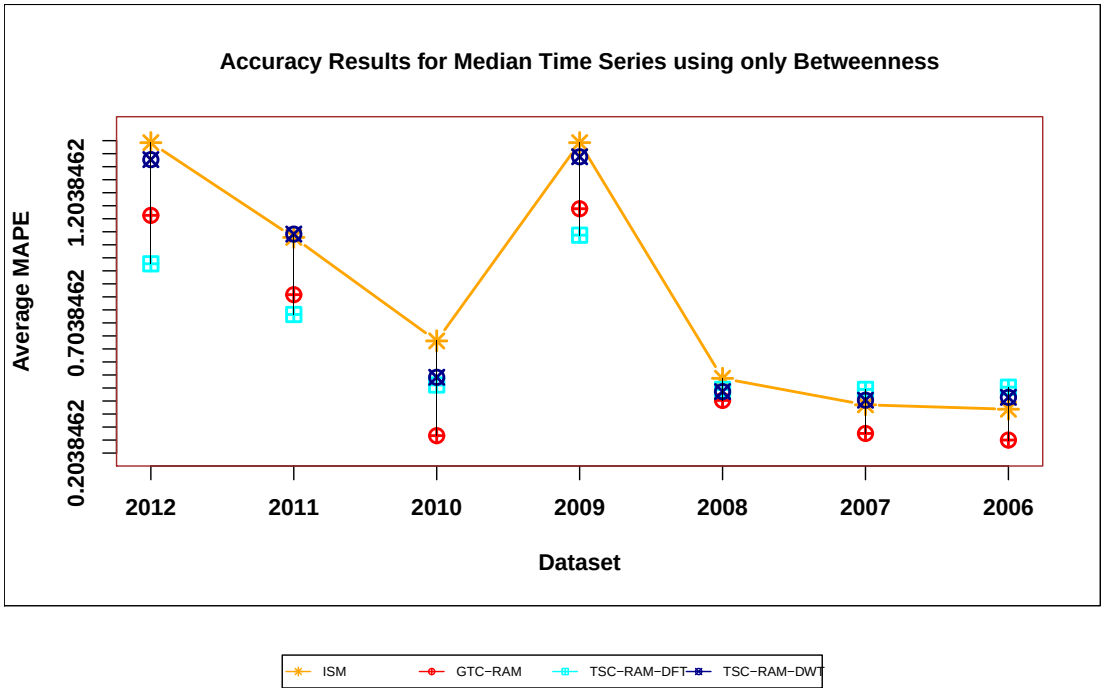

Figure 6.22: Effect of using only betweenness centrality feature on accuracy for median time series

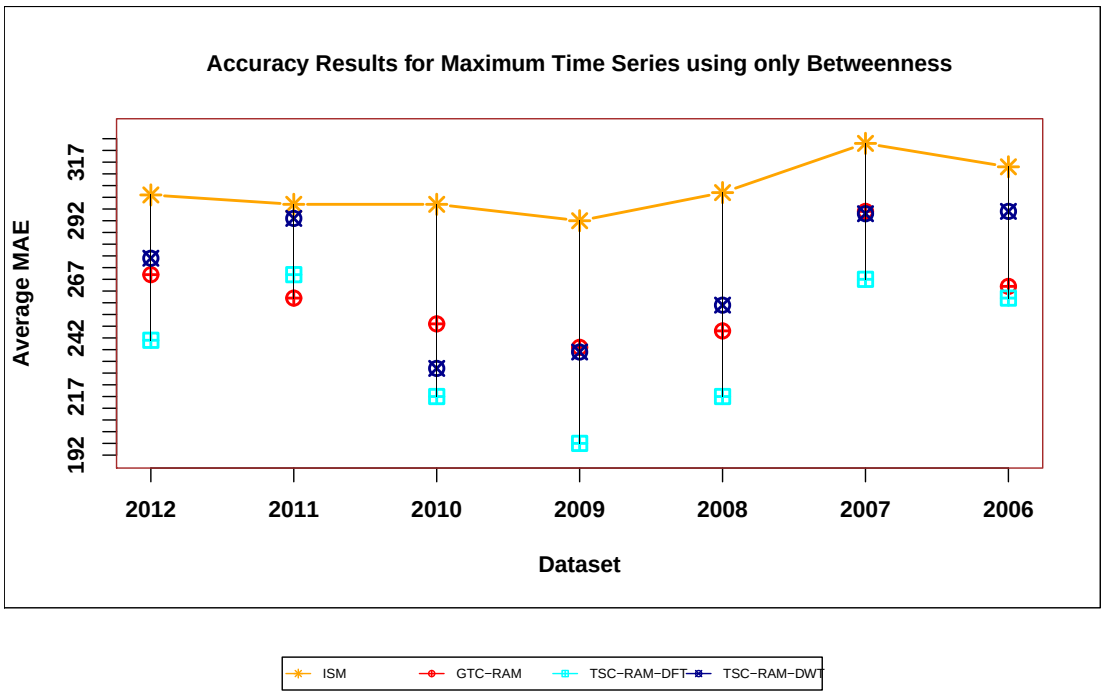

Figure 6.23: Performance of methods for maximum time series measured by MAE

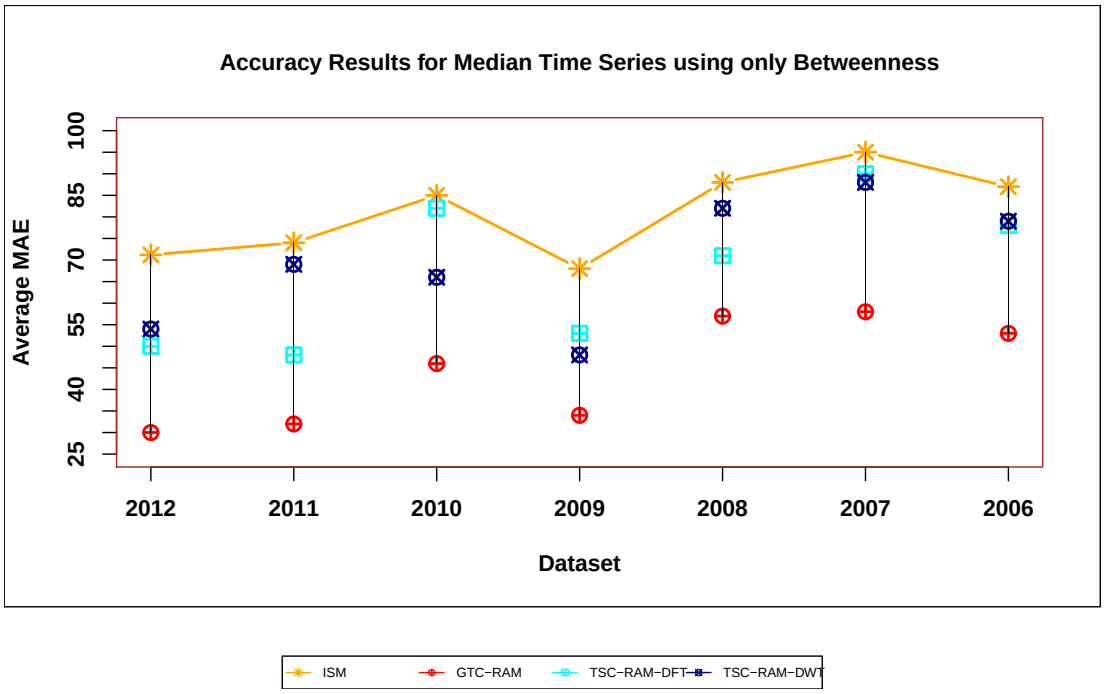

Figure 6.24: Performance of methods for median time series measured by MAE

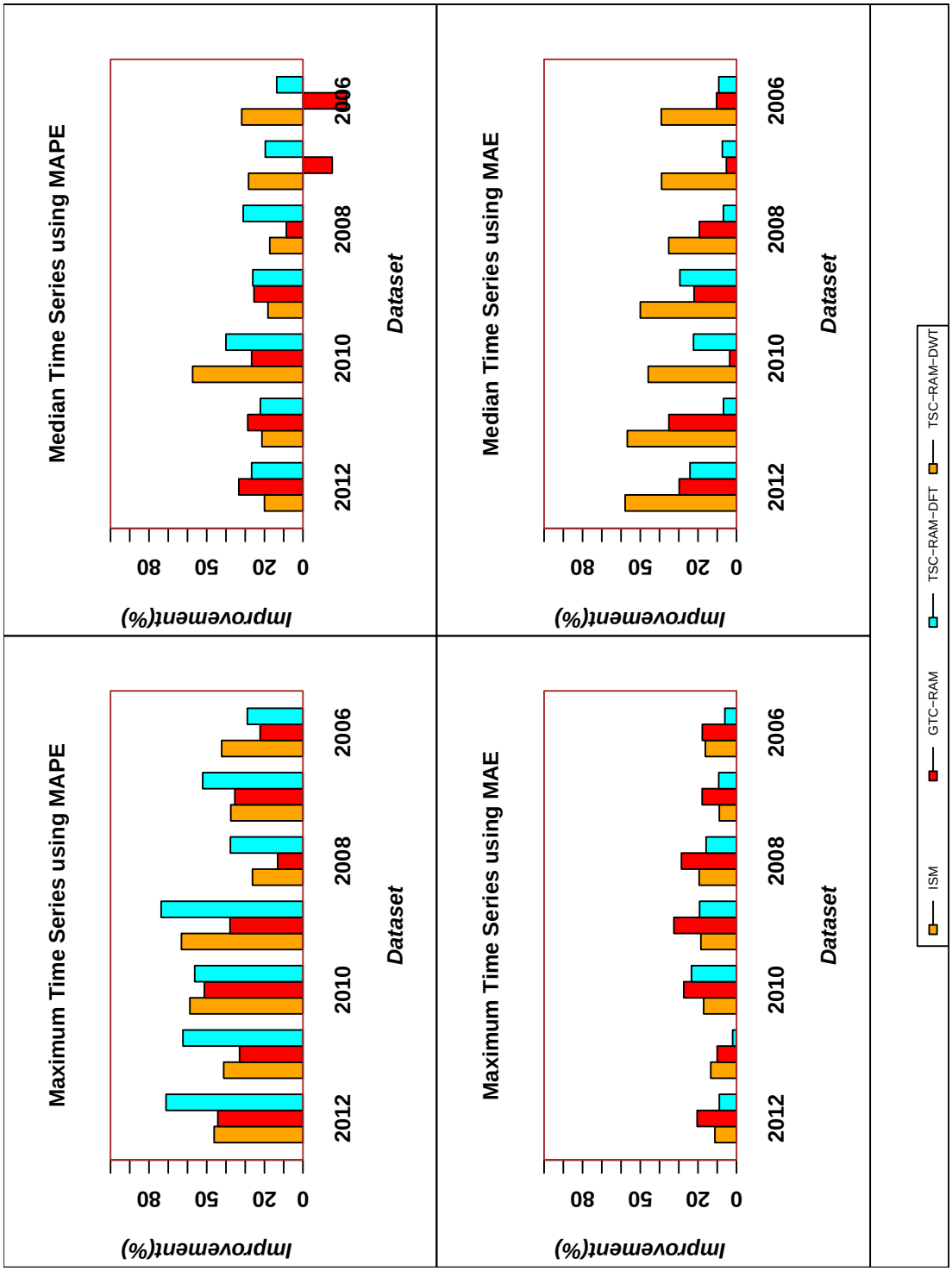

Figure 6.25: Accuracy improvements using only betweenness

Chapter 7

Conclusions

While transportation networks contain rich information, they have not been explored enough for some essential tasks in transportation, such as forecasting flight delays and forecasting traffic speeds. In this thesis, we incorporated transport information and utilized graph based scores, such as betweenness centrality (BC) and articulation points in forecasting of arrival delays and traffic speeds. The position of the nodes in the network and the nodes' delay/traffic time-series similarities are investigated as potential parameters to augment the models for forecasting.

We introduced the Exploratory Clustered Forecasting Modeling (ECFM) that uses a REG-M model enhanced with the results of clustering. The ECFM approach includes grouping and modeling steps that make use of the transport network. The network is used for both graph-based clustering of nodes and as an exploratory variable for the prediction model. Our experiments show that ECFM provides more accurate results than a baseline (e.g. SARIMA, LSTM) model applied individually for each airport. BC score is found to be an effective regressor in the clustered REG-M. When we compare ISM and ECFM with LSTM (Long-Short Term Memory), which is usually considered as the state of the art in forecasting, ECFM is found to be as good as LSTM, which are both more successful than ISM. If LSTM models are used as part of ECFM (e.g. REG-LSTM),

then this returns more successful results compared to the baseline models. This observation suggests that using network structure in similar forecasting problems is a promising direction to pursue.

To the best of our knowledge, this is among the first to utilize the transport network for forecasting flight delays or traffic speeds. Our work may inspire other types of analysis based on transportation networks. The trajectory of the delays/traffics can be analyzed by differentiating the airports/sensors that cause the delay/traffic propagation and those that are the victims of the propagation. This line of work can help policy makers to analyze transportation networks and improve traffic flow management.

Bibliography

[1] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

[2] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[3] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," *arXiv preprint arXiv:1703.06118*, 2017.

[4] V. Martinez, *Flight Delay Prediction*. ETH Zürich, Department of Computer Science, 2012.

[5] N. G. Rupp, "Further investigations into the causes of flight delays," 2007.

[6] C. Barnhart, D. Fearing, and V. Vaze, "Modeling passenger travel and delays in the national air transportation system," *Submitted to Operations Research*, 2010.

[7] N. G. Rupp and G. M. Holmes, "An investigation into the determinants of flight cancellations," *Economica*, vol. 73, no. 292, pp. 749–783, 2006.

[8] X. Fu, W. Luo, C. Xu, and X. Zhao, "Short-term traffic speed prediction method for urban road sections based on wavelet transform and gated recurrent unit," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[9] J. Zhu, Y. Song, L. Zhao, and H. Li, "A3t-gcn: Attention temporal graph convolutional network for traffic forecasting," *arXiv preprint arXiv:2006.11583*, 2020.

[10] S. Jeon and B. Hong, "Monte carlo simulation-based traffic speed forecasting using historical big data," *Future Generation Computer Systems*, vol. 65, pp. 182–195, 2016. Special Issue on Big Data in the Cloud.

[11] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis and future directions," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[12] T. Epelbaum, F. Gamboa, J.-M. Loubes, and J. Martin, "Deep learning applied to road traffic speed forecasting," 2017.

[13] R. Ghodsi, M. Zakerinia, and M. Jokar, "Neural network and fuzzy regression model for forecasting short term price in ontario electricity market," in *Proceedings of the 41st International Conference on Computers & Industrial Engineering, Last accessed: 20th April*, 2016.

[14] F. Van den Bossche, G. Wets, and T. Brijs, "A regression model with arima errors to investigate the frequency and severity of road traffic accidents," 2004.

[15] C. Dritsaki, "Forecast of sarima models: n application to unemployment rates of greece," vol. 4, pp. 136–148, 01 2016.

[16] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[17] M. Kumar and N. R. Patel, "Using clustering to improve sales forecasts in retail merchandising," *Annals of Operations Research*, vol. 174, no. 1, pp. 33–46, 2010.

[18] İ. Gür, M. Güvercin, and H. Ferhatosmanoglu, "Scaling forecasting algorithms using clustered modeling," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 24, no. 1, pp. 51–65, 2015.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[20] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[21] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, (New York, NY, USA), pp. 3:1–3:8, ACM, 2012.

[22] B. Hoppe and C. Reinelt, "Social network analysis and the evaluation of leadership networks," *The Leadership Quarterly*, vol. 21, no. 4, pp. 600 – 619, 2010. Leadership Development Evaluation.

[23] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1979.

[24] D. R. White and S. P. Borgatti, "Betweenness centrality measures for directed graphs," *Social Networks*, vol. 16, no. 4, pp. 335–346, 1994.

[25] F. Fouss, M. Saerens, and J.-M. Renders, "Links between kleinberg's hubs and authorities, correspondence analysis, and markov chains.," in *ICDM*, pp. 521–524, IEEE Computer Society, 2003.

[26] M. Zanin and F. Lillo, "Modelling the air transport with complex networks: a short review," *European Physical Journal Special Topics*, vol. 215, no. 292, pp. 5–21, 2013.

[27] G. Santos and M. Robin, "Determinants of delays at european airports," *Transportation Research Part B: Methodological*, vol. 44, no. 3, pp. 392 – 403, 2010. Economic Analysis of Airport Congestion.

[28] A. Kim and M. Hansen, "Deconstructing delay: A non-parametric approach to analyzing delay changes in single server queuing systems," *Transportation Research Part B: Methodological*, vol. 58, no. 0, pp. 119 – 133, 2013.

[29] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 60–75, 2013.

[30] Y. Cheng, "Solving push-out conflicts in apron taxiways of airports by a network-based simulation," *Computers and Industrial Engineering*, vol. 34, no. 2, pp. 351 – 369, 1998.

[31] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation research part C: Emerging technologies*, vol. 44, pp. 231–241, 2014.

[32] "FAA Strategic Plan, FY 2019-2022."

[33] "Federal Aviation Administration 2016 National Aviation Research Plan (NARP), Report of the FAA to the US States Congress."

[34] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[36] M. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 36104, 2006.

[37] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review*, vol. E 69, no. 026113, 2004.

[38] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[39] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Arxiv preprint cond-mat/0603718*, 2006.

[40] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, p. 066111, 2004.

[41] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*, vol. 23. ACM, 1994.

[42] K.-P. Chan and A.-C. Fu, "Efficient time series matching by wavelets," in *Data Engineering, 1999. Proceedings., 15th International Conference on*, pp. 126–133, IEEE, 1999.

[43] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep lstm recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[44] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[45] A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith, "Clustering rules: a comparison of partitioning and hierarchical clustering algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 475–504, 2006.