Network Tomography - Compressed Sensing Techniques for Traffic Matrix Estimation

By

Rashida Ali Memon

(Registration No: 00000099785)

A thesis submitted to the National University of Sciences and Technology, Islamabad in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in

Electrical Engineering

Thesis Supervisor: Dr. Sameer Hashmat Qazi

Co-Supervisor: Dr. Capt Syed Sajjad Haider Zaidi

Department of Electrical and Power Electronics

Pakistan Navy Engineering College (PNEC)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2021)

Network Tomography - Compressed Sensing Techniques for Traffic Matrix Estimation

By

Rashida Ali Memon

Supervisor

Dr. Sameer H. Qazi

Co-Supervisor

Capt Dr. Syed Sajjad H. Zaidi

G.E.C Committee

Dr. Bilal Muhammad Khan

Dr. Lubna Moin

Dr. Bhawani Shankar Chowdry (External)

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

Doctor of Philosophy (PhD) in **Electrical Engineering**

Department of Electronics and Power Engineering

Pakistan Navy Engineering College(PNEC), Karachi

National University of Sciences and Technology (NUST), Islamabad

July, 2021

Author's Declaration

I, **Rashida Ali Memon** hereby state that my PhD thesis titled **"Network Tomography - Compressed Sensing Technique for Traffic Matrix Estimation"** is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology (NUST), Islamabad, or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after my Graduation, the university has the right to withdraw my PhD degree.

Signature: ____________________

Dated: Name of Student: (Rashida Ali Memon)

Plagiarism Undertaking

I solemnly declare that research work presented in the thesis titled **"Network Tomography - Compressed Sensing Technique for Traffic Matrix Estimation"** is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and that of National University of Sciences and Technology (NUST), Islamabad, towards plagiarism. Therefore I, as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD degree, the University reserves the rights to withdraw / revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC / University Website on which names of students are placed who submitted plagiarized thesis.

Student/ Author Signature: ____________________

Dated:

Name of Student: (Rashida Ali Memon)

Dedicated to *my family and teachers.*

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations

TME	Traffic Matrix Estimation
TM	Traffic Matrix
OD	Origin-Destination
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GLM	Generalized Linear Model
RBM	Restricted Boltzmann Machine
DBN	Deep Belief Network
SP	Single Prior
SP-BV	Single Prior-Bounded Value
TRE	Temporal Relative Error
SRE	Spatial Relative Error
FCN	Fully Connected Network
CNTME	Single Prior
CDF	Cumulative Distribution Function
R-CNTME	Single Prior
GPU	Graphics Processing Unit
MSE	Mean Square Error
SD	Standard Deviation

AMD	Advanced Micro Devices
RMSE	Root Mean Square Error
NN	Neural Network
OD	Origin Destination
TG	Tomogravity
CS	Compressed Sensing
SF	Sparsity Factor
IP	Internet Protocol
ISP	Internet Service Providers
PaaS	Platform as a Service
POP	Point of Presence
IRIS	Iteratively Reweighted Least Square
IPM	Interior Point Method
ITM	Information Theory Method

Abstract

Traffic Matrix Estimation (TME) techniques address the problem of determination of a network's traffic demand matrix from its link load measurements and is considered critical for capacity planning, anomaly detection and many other network management related tasks. With the advent of cloud services such as IaaS (Infrastructure as a Service), Paas (Platform as a Service) and Saas (Software as a Service), the traffic patterns are difficult to model since they do not follow a single probability distribution such as Poisson, Gaussian, Negative Binomial etc., thus decreasing the estimation accuracy using the available methods.

Traffic Matrix Estimation for a large network with accuracy is of utmost importance and is considered a challenging problem. Many approaches use statistical inference distribution on traffic matrix elements that rely on initial or available measurements of the traffic flow (mean and variance). This thesis asserts and proposes a solution for the estimation of traffic matrix that possibly exhibits over-dispersion, which is a more severe problem with mice flows (i.e. small flows) than the elephant flows (i.e. large flows). Moreover, this thesis presents a traffic matrix estimator which shows optimal performance while minimizing errors when there are sparse and limited measurements(training datasets) availability. Furthermore, this thesis inves-

tigates the effects of sparsity and measurement errors (training data errors) for a large network.

The main contribution of this thesis are 1) investigation for the traffic matrix that may experience over-dispersion and formulation of a two-step optimization approach with appropriate accuracy and additional constraint. 2) Investigate and development of a novel architecture that demonstrates superior outcomes for simulations for real datasets and 3) review the case of traffic matrix estimation in which the measurements (training datasets) may be limited in size and may have missing information or incomplete data with errors

Chapter 1

Introduction

As with the rapid growth telecommunications traffic and network needs network management and planning to establish equipment requirements and routing information for subdivisions of the Internet is the significant responsibility of service providers. Network traffic analysis and management plays is the crucial task for the management of the large networks.

To manage these IP backbone networks traffic matrix information or measurements are required to design , plan and monitor a network. Network managements tasks includes load balancing , routing protocols configuration etc .Traffic matrix represents the amount of traffic flows among all possible origin-destination traffic pairs .The traffic volume between OD nodes is give by traffic matrix elements. A router , link or point-of-presence (POP) can be represented as node in a traffic network and traffic volume is expressed as number of packets and a router or link defines volume of traffic for a certain time is denoted as counts.

For a certain link measurements its difficult to estimate OD flows information in Traffic matrix since in a network number of the links are usually less the number of OD flows or simply, more unknown quantities are require to be estimated from a limited number of known data (link measurement). A ill-posed network problem is observed here also called an under-constrained network problem and solution for the ill-posed problem is provided by several researchers using multiple techniques.

Network tomography method is a modern estimation and prediction technique for traffic matrix prediction with accuracy. This techniques/ model also depicts a major ill-posed problem in estimation and prediction [4]. Numerous TM estimation techniques [5–7] exploit the requirement of the prior traffic matrix element information and end-to-end measurements, for the distribution and optimization assumptions for reliable modelling for the prediction of traffic measurements [8, 9].

Two type of network applications need traffic matrix estimation: 1) Online application and 2) Offline application. First one is related to network traffic management as it needs to get Traffic matrix within few seconds during a short real time period. Second one is related to network planning as it requires a Traffic matrix representing the mean and the static change of OD flows. Therefore, traffic matrix estimation techniques are directly propositional to the utilization of traffic matrix in different network applications.

Various literatures emphasized on supplementary data requirement for the estimation more accurately [10].Popular traditional methods includes tomogravity method [11], the principal component analysis (PCA) method, [10], etc. However with the increasing large networks the ill posed and under-determined system is becoming a challenging for the prediction and estimation [4, 6, 7]. Current network architectures and applications have new network traffic statistical features but they are still modeled by Poisson and Gaussian models, that create anomalies and inaccuracies in the estimation of network parameters [10] .

1.1 Focus of this Dissertation

Traffic matrix (TM) estimation techniques for multiple network applications are proposed in this thesis which will allow us to estimate accurate traffic matrix along with the other network parameters. Considering the link count measurements are provided by some real datasets. Accurate prediction of the traffic for all source-destination nodes connected through traffic routes is the main object of traffic matrix estimation.

Initially the prediction or estimations techniques for the networks are classified in two categories 1) the deterministic and 2) the statistical . For the deterministic approach the characteristics of the prior traffic matrix are required to recognize a accurate traffic matrix due to under-constrained or ill posed feature traffic matrix estimation. However the statistical approach required more information about traffic flows between Origin-Destination(OD) nodes for the true prediction and estimation of traffic matrix.

For the first contribution we refer the over dispersion problem where statistical methods usually fail when accuracy is considered. Statistical prediction approaches do not provide a reasonably accurate solution when faced with the problem of excessive dispersion. This work shows through real world datasets that dispersion (over-dispersion in our case) causes serious issues with smaller flows. As a result, a two-stage optimization strategy is proposed where larger flows are predicted with reasonable accuracy in the first step with more conservative estimates for dispersed smaller flows. A second step for optimization with more restriction refines the solution for dispersed small flows. Experimental results demonstrate that for ill-estimated flows, prediction can be increased up to 4 orders of predicted values.

For second contribution , a traffic matrix estimation framework is developed with guaranteed superior performance with availability of limited training data or outlier measurements. Moreover we investigated the limiting training data challenges and develop a new algorithms with architecture that can give solution for these difficulties and guaranteed better performance. This approach provides superior outcomes for estimation of traffic matrix using convolutional neural network based technique with limited training data availability and outlier end-to-end measurements.

The thesis flow of chapters is as follows: Chapter 2 consist of detailed literature review, Chapter 3 outlines real world datasets used for our proposed techniques. Chapter 4 discusses the first contribution estimation of traffic matrix elements that exhibits less dispersion, which are then step-wise applied for the estimation of the traffic matrix elements suffering from more over dispersion results. Chapter 5 discusses the second contribution traffic matrix estimation framework based on neural network and Chapter 6 finally summarizes and concludes the thesis.

Chapter 2

Traffic Matrix Estimation

Traffic matrix plays a vital role in several network applications. These network applications need traffic matrix for the network planning and network management. Traffic matrix mainly provides over all information about the data exchange among different nodes. For network applications, Traffic matrix is very important input to perform various network related task therefore, its accuracy is considered very crucial. To estimate these traffic matrix, Traffic Matrix Estimation techniques must be efficient and accurate for the effective performance of network applications.

In this chapter numerous Traffic matrix estimation techniques and approaches which are required for multiple applications for large networks such as network designing, dimensions and planning, network management and congestion control are discussed in detail.

2.1 Traffic Matrix Estimation Techniques

TME techniques which are produced in large-extent networks are distributed into two main categories: statistical and deterministic This distribution is based on the consumption of the techniques in a variety of network applications. In below, we detailed review about these categories are discuss.

2.1.1 Deterministic approach

Several Traffic matrix estimation techniques that fall into the category of this approach are the methods that apply prior traffic matrix features to optimize or find the unique solution in that region. Deterministic techniques use link count measurements, thus traffic data between all OD traffic pairs are also represented as constant values in traffic matrix. Usually, deterministic approach provide us the Traffic Matrix estimation problem as optimization of ill posed problem. Traffic Matrix estimation approaches that are related to this approach use link count measurements as constraints having different objective functions. One of the approach [12, 13] use objective function as max of sum of weighted flows and another approach [11, 13] set the objective function for the minimum distance between predefined Traffic matrix that was supposed to be combined with real traffic matrix and created ill-posed problem.

Usually all deterministic techniques express traffic matrix estimation problem as optimization problem. Table 2.1 shows few deterministic approaches considering traffic matrix problem as optimizations problem. These techniques using determin-

	Objective Function	Constraint
Linear Program	maximize $\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}$	$Ax \leq Y$
Least Square	minimize $\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij} - m_{ij})^2$	$AX = Y$
Information Theory	minimize $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{x_{ij}}{T_X}\{log\frac{x_{ij}}{m_{ij}}\}$	$Ax = Y$
Generalized Kruithof	minimize $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{x_{ij}}{T_X}\{log\frac{x_{ij}}{m_{ij}} - log\frac{T_x}{T_M}\}$	$Bx = \beta$

Table 2.1: A variety of deterministic approaches for TME

istic approach are Least Square approach [11], Linear Programming (LP) approach [11, 14], Generalized Kruithof approach [15],[11] Information coding and theory [11], [11] . Apply minimize of $|AX - Y|$ rather than $AX = Y$ due to the uncertainty of the accuracy of real network link counts.

2.1.2 Statistical Approach

This approach is based on assumption on traffic model. The origin-destination (OD) pair traffic follow a certain distribution such as the Poisson [16, 17] or Gaussian [18]. However, such assumption can be applied on limited number of elements of the Traffic matrix , eventually it can be applied on specific time period , thus the hypothesis of these distributions is not valid in general scenario [19].

The statistical approach use the random variable for link count measurement specified over continuous time periods therefore in traffic matrix, all origin-destination (OD) pairs estimated represents model with statistical characteristics of the traffic measurements.The Statistical methods include one-step and 2-step Expectation minimization methods, [17], [17], it considers the Bayesian approach method [16] under its optimization and it also includes Iterative method [17] and moments method [16]. Traditionally, traffic models such as Poisson [16], and Gaussian [18] were used in the development of modelling OD measurements. Recent work shows the implementation of artificial neural networks used as statistical approach as well using numerous ANN artificial neural network techniques. Statistical approach provides traffic matrix that combines mean, variance as well as statistical characteristics required for the different network applications.

2.2 Traffic Matrix Estimation Challenges

Prediction and accurate estimation of traffic matrix is mathematically calculated using equation (2.2.1).

$$Y = AX \tag{2.2.1}$$

In the equation three network parameters are used for a smooth prediction and estimation. These parameters are measured link loads Y , routing matrix A and traffic flows X. Routing protocols such as Intermediate System-Intermediate System (IS-IS) [20] and Open Shortest Path First (OSPF) [21],can be helpful to find the routes. For large networks, these routes can be obtained using famous algorithms such as Dijkstra [22] or Bellman-Ford algorithms [23] ;and SNMP protocol.

For the first step in the traffic matrix estimation, X (traffic flows), Y (link loads) and A (routing matrix) are required for estimation of certain traffic. Equation (2.1.1) depicts ill-posed condition for under-constrained system , where count of OD flows is more then the provided number of measured links. This will lead to a infinite set of solutions for equation (2.2.1) and there will be no unique solution. This is called ill-posed network problem.

2.3 Problem in the Deterministic Approach

Consider a 3-node network with 2 links and 3 origin-destination flows as shown in figure 2.1. Links are measured as 12 and 16 and these links can be used for estimation other link flows.

Measured link loads as shown in figure give us the summation results of 12 and 16 as measure link loads. These constrains are shown in figure 2.2 in the form of equations. This shows the ill-posed network problem where number of knowns and number of unknowns are under-determined and they have no unique solutuon.

The line AB (hyperplane AB) as shown in figure 2.2 denotes the solutions. Point P denotes the initial available solution which is required for the prediction of new solutions. The hyperplane solution is satisfying all spatio-temporal constraints.The outcome is initially available traffic matrix or calculated using an alternative method using link measurements such as the gravity model.

Similarly, we consider another toy network in figure 2.3 which have 4 directed links and six OD paths. Routing matrix is as shown in figure in matrix form, routing matrix rows represent the connected links and routing matrix and it represents the OD pairs for linear solution.

To estimate of the traffic matrix with accuracy, it is assumed that an prior information of the Origin-Destination traffic is accessible for the connection between the traffic flows and connected links. Once again network shows ill-posed problem and system of equations for solution is under-constrained , more count of unobservable then observable are present. This is again an ill-posed or under-constrained problem as the number of unobservable is greater than the number of constraints (observables)

. Consequently, this is ill-posed problem which will not produce unique solution. However, we can apply mathematical model rather than deterministic solution. Some researchers used historic data to calculate prior, then applied some algorithm to estimate the matrix.

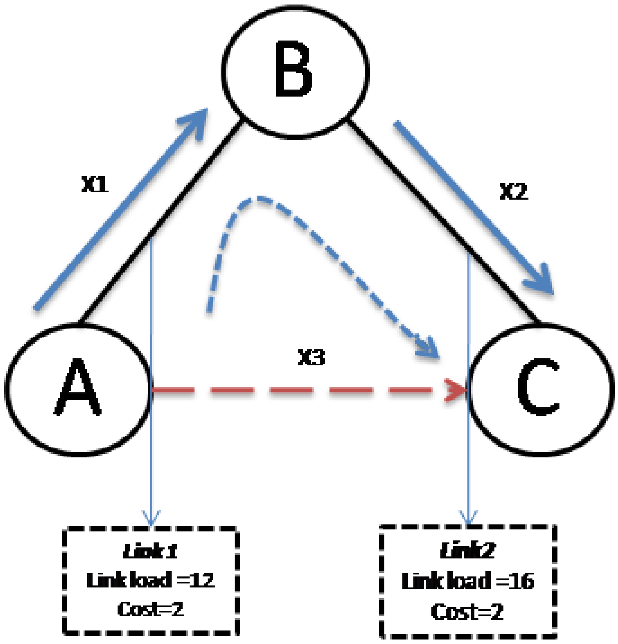

Figure 2.1: Toy network

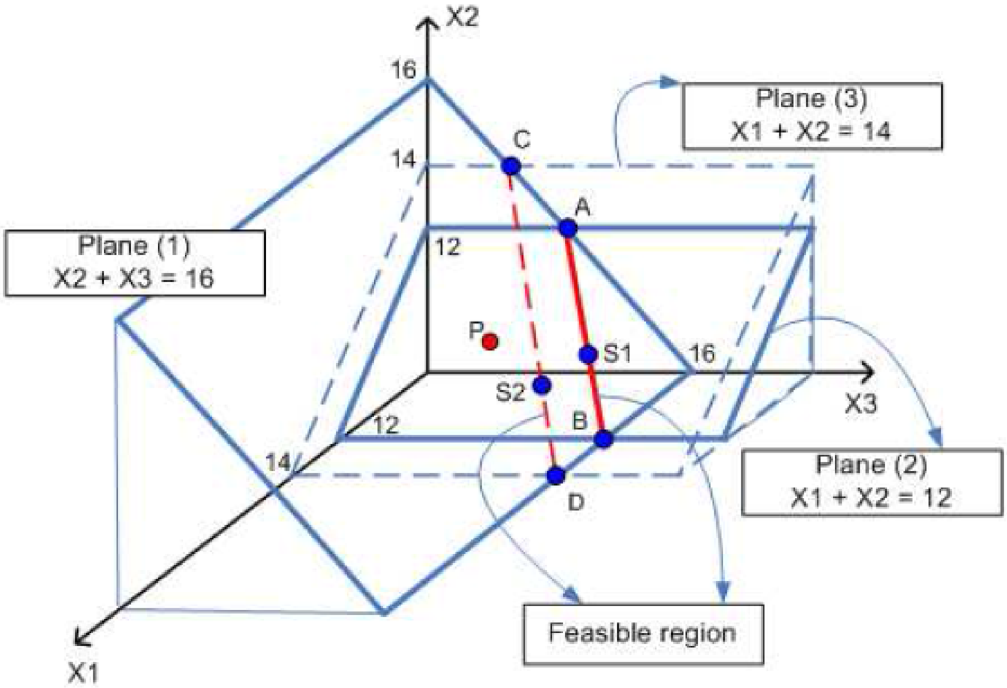

Figure 2.2: Solution Hyperplane

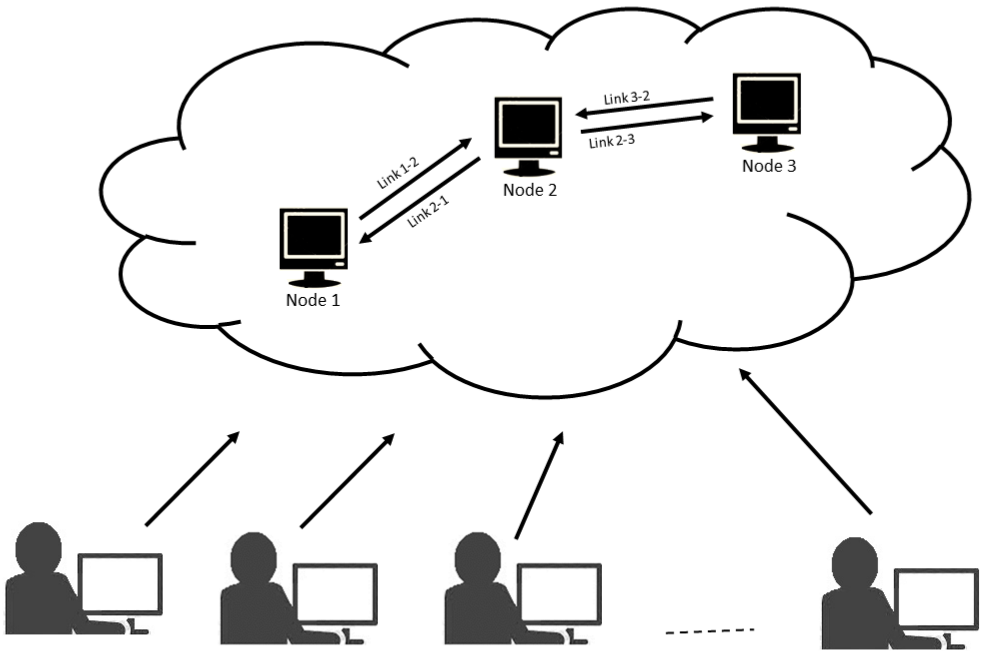

Figure 2.3: A toy cloud network

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{2.3.1}$$

2.4 Traffic Modeling Problem in the Statistical Approach

Generally four inputs are applied to the statistical approaches as shown in flowchart in Figure 2.4.

- Assumption on the elements of the traffic matrix: These assumptions influence particular statistical approach that will be useful and applied.
- Prior Traffic matrix measurements: Statistical methods required a previous es-

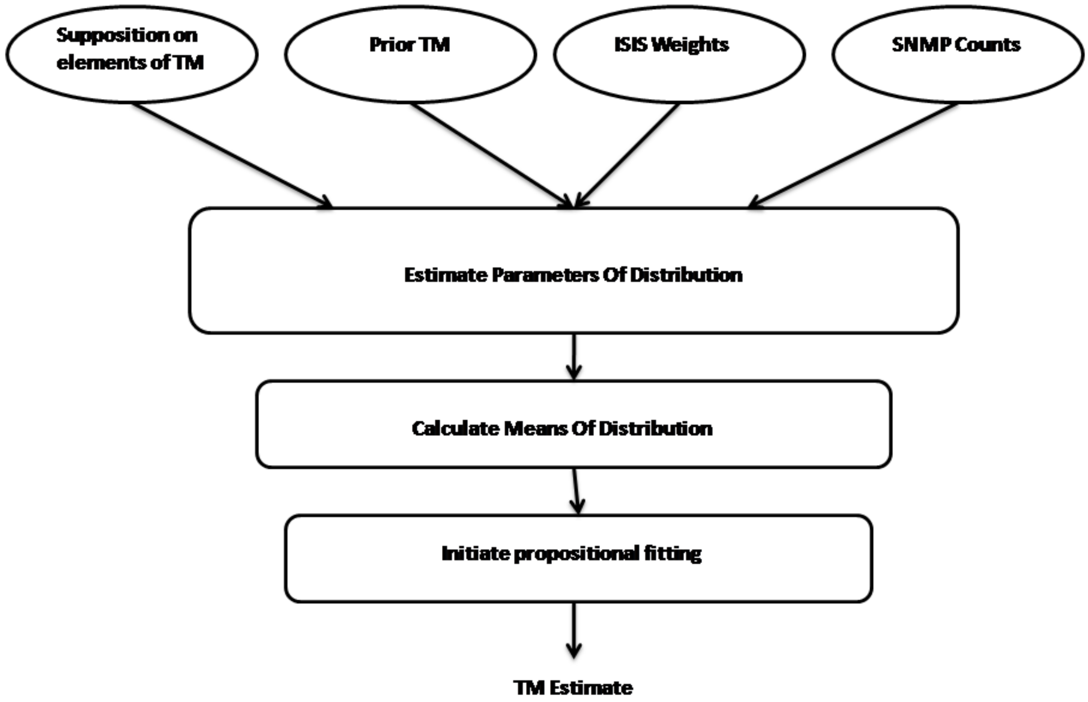

Figure 2.4: General diagram for statistical approach

timate of Traffic matrix achieved from prior data and measurements.

- ISIS weights: This input helps to calculate shortest paths to generate the A matrix.
- SNMP data: SNMP is required to enforce constrains on predicted traffic matrix. It provides observed link counts.

Statistical approaches vary with assumptions applied on the traffic matrix components, and on the particular method, to predict and estimate the corresponding parameters. However, the Expectation Minimization method start with a random matrix to estimate the Traffic matrix , but it may get trapped in the optimization because of some wrong assumptions.

2.5 Various Techniques for TM Estimation

Many techniques and algorithms are introduced previously for Traffic Matrix estimation for large networks. We analyze and review explaining each technique proposed earlier.

2.5.1 Generalized Linear methods (GLMs)

In s statistical model, there are two components of linear methods namely systematic and random. Firstly we consider systematic components as shown in equation (2.5.1).

$$Y = \sum_{a=1}^{m} \gamma_a x_a \tag{2.5.1}$$

Where, xa represents the known independent variable and γ_a are constant parameter. In our model we are collaborating both random and systematic components. Now a GLM is examined which is indicated as,

- Let z be a dependent random variable whose parameter θ_1 with distribution is discussed in section 1.1.
- $X_1 ... X_m$ are the set of independent variables whose predicted result Y.
- A mapping function $\theta = f(y)$ linking the parameter θ with the predicted value Y of the GLM model.

Where z represent normal distribution with θ as mean For optimization of the traffic matrix the GLM make an important role. NBM and Binomial with optimal properties are under consideration for the purpose of optimization of traffic matrix A.

2.5.1.1 Poisson Variation

In poison distribution the variance of the data must be equal to the mean.

$$var(Y) = E(Y) \tag{2.5.2}$$

Although, we have a data that perform over-dispersion where the mean must be smaller than variance. In [24] Nelder and Wedderburn presented the relation between variance and mean by applying iteratively-reweighted least squares (IRLS) algorithm that only depends on the variance and mean of the data. The final estimates are known as quasi-likelihood estimates (MQLE), In maximum likelihood estimation (MLE) and there are many properties of optimality which are shown below. Let us assume that the variance is directly proportional to mean.

$$var(Y) = \phi\, E(Y) \tag{2.5.3}$$

$$\left\{\begin{matrix} If\phi = 1 & \text{then the variance equal to mean} \\ If\phi > 1 & \text{over-dispersion occurs} \\ If\phi < 1 & \text{under-dispersion occurs} \end{matrix}\right\} \tag{2.5.4}$$

In IRLS algorithm the weight $w^* = \phi\, E(Y)$, poisson weight $w = \mu$ divided by ϕ, but cancel the factor of of ϕ during the computation of estimator which optimize the poisson MLE. Therefore poisson optimizations are MQLE when the variance is directly proportional to the mean. The general form of optimizer variance with parameter ϕ is given by,

$$Var(\gamma) = \phi X^{'} T X^{-1} \tag{2.5.5}$$

Where $T = diag(\omega_1, \ldots, \omega_n)$, When $\phi = 1$ it will reduce the Poisson variance. This shows that the Poisson distribution error is fundamental in the existence of over-dispersion.

2.5.1.2 Negative Binomial

Negative binomial is an alternative method for over-dispersion modeling. It is start with count data from a PRM and substitute a multiplicative random effect ϕ to characterize unobserved heterogeneity.

Let us assume we have conditional distribution of Y given that unobserved random variable ϕ with variance and mean,

$$Y|\phi \sim P(\mu_{\phi}) \tag{2.5.6}$$

In this negative poission mode the the data must be poisson if we only find ϕ. By taking integrate it out of the likelihood function as assumption related to distribution, efficiently solving the unconditional distribution of the consequence. Moreover, for mathematically convenient assume that ϕ has γ distribution with parameters a and b. This distribution with mean a/b and variance a/b^2, consider $a = b = 1/\sigma^2$ which create the mean of the unobserved value equal to 1 . With this data we can find the unconditional distribution of the event which is called negative binomial distribution. The representation of density in term of parameter a,b and mean calculated below. Even though in our case $a = b = 1/\sigma^2$.

$$P(Y = y) = \frac{\Gamma(a + y)}{y!\Gamma(a)} \frac{b^a \mu^y}{(\mu + b)^{a+y}} \tag{2.5.7}$$

This is the best distribution in term of number of failure a prior to k-th sequence of Bernoulli distribution with probability success of π. The π density related to analysis applying substitution from the illustration above. The NBD with parameters $a = b = 1/\sigma^2$ with expected value and variance is given by,

$$\begin{aligned} E(Y) &= \mu \\ Var(Y) &= \mu(1 + \sigma^2 \mu) \end{aligned} \tag{2.5.8}$$

- If $\sigma^2 = 0$ there is no unobserved heterogeneity.
- If $\sigma^2 > 0$ then the variance is greater than the mean. So the NBD is ober-dispersed.

More generally, these types of events can be derived with the help of iterated law of expectations with taking under consideration of gamma distribution. We need conditional probability events $E(Y|\phi) = Var(Y|\phi) = \phi\mu$ with assumption of mean equal to 1 and variance. The unconditional mean is generally the expected value of the conditional mean.

$$E(Y) = E_\phi[E_{Y|\phi}(Y|\phi)] = E_{\phi\mu} = \mu E_\phi(\phi) = \mu \tag{2.5.9}$$

Where, subscripts shows clarification over distribution when considering in expectations.

$$\begin{aligned} Var(Y) &= E_{\phi}[Var_{Y|\phi}(Y|\phi)] + Var_{\phi}[E_{\phi}(\phi\mu)] \\ Var(Y) &= E_{\phi}(\phi\mu) + Var_{phi}(\phi\mu) \\ Var(Y) &= \mu E_{\phi}(\phi) + \mu^2 Var_{phi}(\phi) \\ Var(Y) &= \mu + \mu^2\sigma^2 \\ Var(Y) &= \mu(1 + \mu\sigma^2) \end{aligned} \tag{2.5.10}$$

Subscript is used to clarify over the distribution we are taking the expected values or variance.

2.5.1.3 Simplex Method (SM)

SM [25] is the most general method for numerical solution of the LPM. SM is based on optimal LPM that are associated with extreme value of the under-observed region. Thus, SM circulate around the boundary layer of the observed region which are able to enhance the performance of objective function. As a consequence, some solution are ignored because they are not related to extreme value.

2.5.2 Linear Programming technique

Goldschmidt [12] intended traffic matrix prediction and optimization problem by means of a Linear Programming (LP) method. Enhancing OD flows summation and relating it to limitations that showing the OD flows on a link that equal or lesser compared to link capacity. Suggestion by authors are that it would be valuable if path length as a cost function is added weight to obtain more accurate results. Several different methods have been implemented to tackle the linear programming formulation such as Simplex method and Interior Point Technique.

2.5.2.1 Interior Point Method (IPM)

SM is associated with relative extreme value along the edge of the observed region. In IP method the interior of the observed region cuts to reach the optimal solution. This concept was implementing for a long time [26, 27]. The LBM of Frisch [28–30] and the CM of Huard [31, 32]. The IPM is divided into 3 types, Projective-Scaling (PS) technique [33], and Path-following (PF) technique [34, 35] and affine-scaling (AS) technique. The affine scaling technique is implemented in [14] for the solution of optimization problem TM estimation for nonlinear systems.

2.5.3 Gravity Approach Method

For traffic matrix estimation, the gravity based models are used in wide range of fields in social science [36–38], [39–41], [42, 43], in telephony network [42, 43] and in transportation networks[39–41]. Initially gravity model was applied in [44] and was more elaborated and evaluated as Choice Model [19].

Generally, the gravity model is applied on a network to get a pre-defined Traffic Matrix known as prior TM. This pre-defined traffic matrix include characteristics of the real TM and estimation techniques associate with prior TM to determine the solution for ill-posed or under constrained traffic estimation problem.

2.5.4 Gravity Model (GM)

It is sometime called trip distribution models which are implemented in transportation application for optimizing traffic matrix [45]. In GM there is trip exchange between urban area and zones which is directly related to attraction of zones and inversely applied to the separate zones functions. For the traffic links optimization issues, we gather data packets transferred to and from nodes.

A generalize equation is shown below.

$$X_{ji} = \frac{g(S_j, W_i)}{u_j i} \tag{2.5.11}$$

Where $g(.)$ is a varying function, X_{ji} is the traffic going from j to i. S_j is a parameter related to leaving j point. W_j is a parameter shows factors related towards j and $g_i j$ shows the friction factors between j and i.

A tomogravity model proposed by Zhang et al [11]. This technique follows two stages. In first stage the generation of prior TM using SGM that creates usage of intra link measurement data. In second stage the optimization of prior TM attained from the GM, using the LSM that find the best set of fitting variable. For larger networks this implementation needs 7 seconds or less by researchers [11]. Generally, the GM is used to get the rough set of TM called apriori TM that provides the specifications of the real TM. Optimization methods such as statistical and deterministic methods cooperate with the prior traffic matrix information to get a accurate estimate of solution formed by the limited TM optimization problem.

2.5.5 Generalized GM

This section presents the straightforward gravity model for optimization and estimation for ill-posed structure of network. As a matter of first importance, we represent the IP network by Fig 2.5. Figure shows three networks labelled B, C and

D and they are associated by large netwrok A . The Large Network have dedicated customer connection and client connection as access and edge points. Generalized models counts the edge and access joins and in the event that the network has no large network connection and access interface, we get the straightforward GM as given below:

$$\begin{aligned} T(e_i, e_j) &= T^{in}(e_i) \frac{T^{out}(e_j}{\sum e_{k\epsilon E} T^{out}(e_k)} \\ T(e_i, e_j) &= \frac{T^{in}(e_i}{\sum e_{k\epsilon E} T^{in}(e_k)} T^{out}(e_j) \end{aligned} \tag{2.5.12}$$

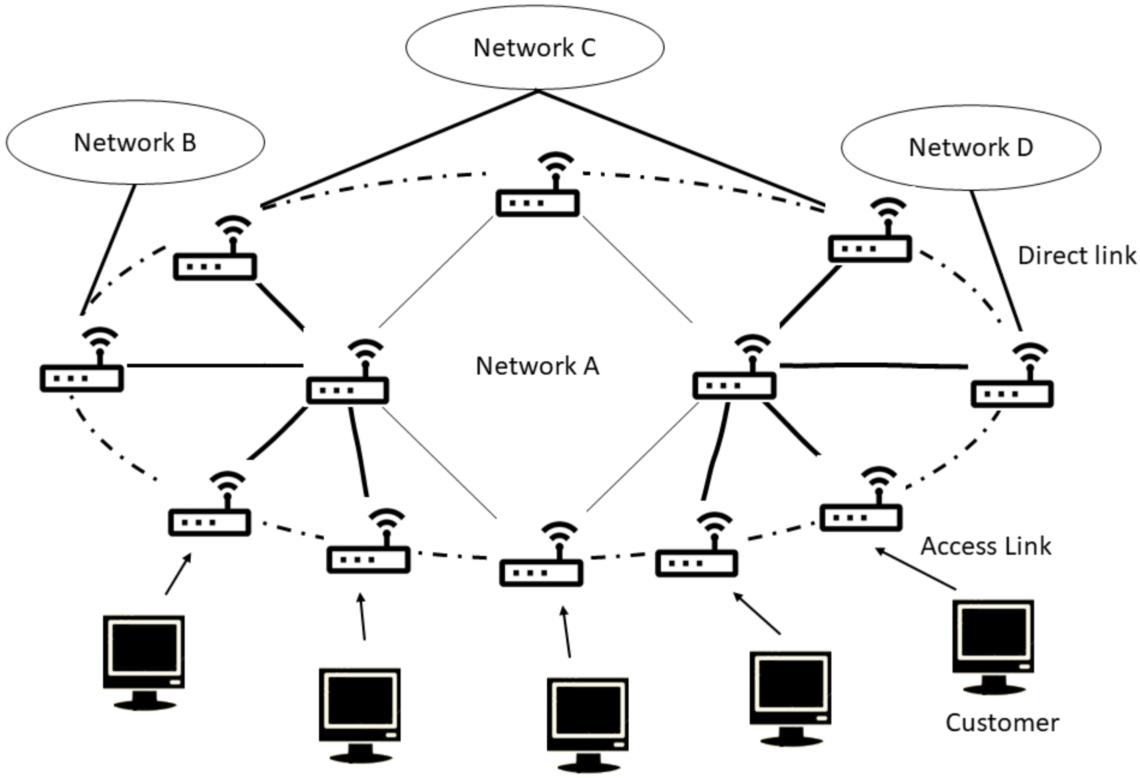

Figure 2.5: The IP network components.

Nonetheless, this GM is unreasonable because of the way that the unmistakable treatment to peer connection and access interface by genuine network is generally existed. Coming up next is the summed up GM which makes the traffic from access connect to large network interface named as edge traffic. The traffic from large network node connect to edge interface in genuine network, p_i the ith edge connection of the network, P_i the arrangement of friend joins conveying traffic to the network, and the connection of all companion joins. Therefore the summed up GM is given below. The output traffic from the entrance interface to the friend connect is provided below :

$$T_{out}(c_i, p_j) = \begin{bmatrix} \frac{T^{in}(c_i)}{\sum_{c_k} L^{in}(c_k)} \sum_{p_k \in P_m} T^{out}(p_k), & if p_j \in P_m \\ 0, & otherwise \end{bmatrix} \tag{2.5.13}$$

This section deal with simple gravity model and its generalized version. Initially, an IP network factors are shown in figure 2.5. In the figure, we observe three peer networks, i.e., network B, network C and network D connected to a large network A by the appropriate node links and access point network . Both the access points and node links are denoted as edges, and the interior link in the large network. Both links (interior) and edge links are commanded and thus denoted as directed links. In GM only edge links are involve.

The simple GM for this scenario is defined below: [11],

$$\begin{aligned} L(e_i, e_j) = L^{in}(e_j)\frac{L^{out}(e_j)}{\sum_{e_k} T^{out}(e_k)} \\ L(e_i, e_j) = L^{in}(e_j)\frac{L^{out}(e_j)}{\sum_{e_k} T^{in}(e_k)} T^{out}(e_k) \end{aligned} \tag{2.5.14}$$

E shows the network links. In first stage the generation of prior TM using SGM that creates usage of intra link measurement data. In second stage the optimization of prior TM attained from the GM, using the LSM that find the best set of fitting variable.

Moreover, this GM is unfeasible due to that the fact that the distinct action to access and peer link by real network is extensively existed. The following is simplified GM optimize the traffic matrix for multiple situations. First situation is inbound traffic (peer link to access link), second situation is outbound traffic (access link to peer link) and third situation is internal traffic (access to access link).

The generalized GM is represented as follows.

$$L_{out}(c_i, p_j) = \begin{bmatrix} \frac{L^{in}(c_i)}{\sum_{c_k} L^{in}(c_k)} \sum_{p_k \in P_m} T^{out}(p_k), & \text{i}f p_j \in P_m \\ 0, & otherwise \end{bmatrix} \tag{2.5.15}$$

The inbound traffic from peer to access link is represented as,

$$L_{out}(c_i, p_j) = \frac{L^{out}(c_j)}{\sum_{c_k} L^{out}(c_k)} T^{out}(p_i) \tag{2.5.16}$$

The internal traffic from access to access link c_j is represented by.

$$L_{internal}(c_i, p_j) = \frac{L^{out}(c_j)}{\sum_{c_k} L^{out}(c_k)} T^{out}_{internal}(p_k, c_j) \tag{2.5.17}$$

Where,

$$L^{out}_{internal}(c_j) = T^{out}(c_j) - \sum_{p_k \in P} T_{inbound}(p_k, c_j) \tag{2.5.18}$$

2.5.6 Tomography Model

The traditional statistic approach used for the solution of traffic matrix estimation problem was proposed by Vardi as in equation 2.2.1. .In this section we will talk about equation 2.2.1 and the routing matrix A. Equation 2.2.1 shows the link load vector disclose the load for individual direct link. The binary routing matrix is a matrix, directed links are its rows,the OD pairs are its columns and zero entry shows no directed link.

	1 EF	2 EG	3 EH	4 FE	5 EG	6 FH	7 GE	8 GF	9 GH	10 HE	11 HF	12 HG
E→F			1									
E→G	1	1										
F→E				1			1			1		
F→H					1							
F→H			1			1						
G→F	1							1		1		
G→H							1		1			
H→F							1				1	
H→G										1		1

Table 2.2: The network routing matrix

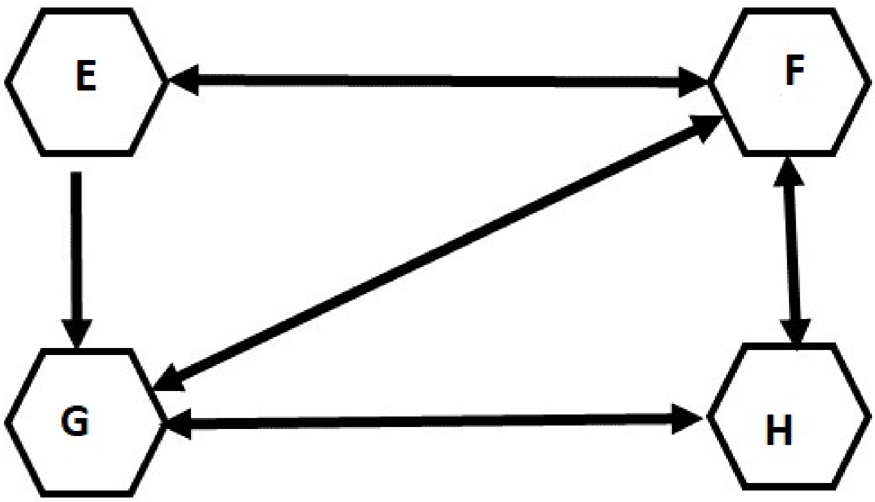

Figure 2.6: The Network Topology

Consider the toy network example in figure 2.5 OD pairs (4x3) and 9 directed links are shown in figure 2.5. Table shows the routing matrix A , the blank entries are zero. Every two node (router) routing is pre-specified. As in example both of the paths G→F→E and G→H→F→E are connecting the OD pair G→E.

2.5.7 Moore-Penrose Inverse and Least Square method

The routing matrix in Moore-Penrose inverse method is given by A^{+} and following four equations reflect the Moore-Penrose Inverse:

$$\begin{aligned} AA^{+}A &= A \\ A^{+}AA^{+} &= A^{+} \\ (AA^{+})^{H} &= AA^{A} \\ (A^{+}A)^{H} &= A^{+}A. \end{aligned} \tag{2.5.19}$$

The general solution of Equation 2.6.2 is as follows

$$X = A^{+}Y + (I_{NxN} - A^{+}A)T, \tag{2.5.20}$$

A^{+} shows the inverse of the routing matrix A (Moore-Penrose), I_{NXN} is representing identity matrix. N-element random vector is given by T, consisting a random value.

2.5.8 Information Theory Method

Information theory and coding is a customary means in communication networks [46]. We start with fundamental probability specifications as PX(x). The ill-posed linear inverse problem is observed in multiple fields as in Seismology, Astronomy, and Medical Imaging [46, 46–49]. Therefore it is concluded that some prior information is required for the reconstruction of the estimation or prediction. To solve the ill-posed problem minimization equation with regularization parameter and penalization function is proposed as in equation below

$$p(x, y) = p(x)p(y) \tag{2.5.21}$$

Equation can be defined as product of the marginal probability is equal to the joint distribution. It can also be related to conditional probability

$$p(x|y) = p(x) \tag{2.5.22}$$

Entropy is also one of the crucial element in the prediction of probabilities. Entropy rule assist us in finding the probability distribution in different situations. Technically, if initial history information about a random variable is unknown then uncertainty regarding that random variable is very elevated, hence to minimize this uncertainty certain distribution should be selected.

And if initial history information about a random variable is known using few constraints, applying rule of thumb of entropy states to maximize the entropy H(X|C) of X conditional. Thus we select a solution to maintain the uncertainty provided the constraints conditions. [47].

The ill-posed inverse problem of a network is observed in multiple fields as in Seismology, Astronomy, and Medical Imaging [46, 46–49]. Therefore it is concluded that some prior information is required for the reconstruction of the estimation or prediction. To solve the ill-posed problem minimization with regularization parameter and penalization function is proposed as in equation (2.5.22)

$$min_x \quad \|y - Ax\|_2^2 + 2.\lambda^2 log\pi(x), \tag{2.5.23}$$

In equation 2.5.23, $\|\|_2$ represents L2 norm, $\lambda > 0$ specifies regularization parameter and penalization function is given by J(x). Equation 10 leads to the proposals used in several fields with pragmatic and theoretical success. These methods and strategies are referred as regularization of ill-posed problems [17]. Bayesian approach [17]is usually considered for regularization when x is estimated from random variable using initially available probability distribution function with density π(x) and Gaussian White noise with variance σ^2.

$$\begin{pmatrix} y_r \\ y_{r-1} \\ \vdots \\ y_1 \\ S_{(r,r)} = cov(y_r, y_r) \\ S_{(r,r-1)} = cov(y_r, y_{r-1}) \\ \vdots \\ S_{(r,1)} = cov(y_r, y_1) \\ S_{(r-1,r-1)} = cov(y_{r-1}, y_{r-1}) \\ S_{(r-1,r-2)} = cov(y_{r-1}, y_{r-2}) \\ \vdots \\ S_{(r-1,1)} = cov(y_{r-1}, y_1) \\ \vdots \\ \vdots \\ S_{(1,1)} = cov(y_1, y_1) \end{pmatrix} = \begin{pmatrix} a_{r,c} \\ a_{r-1,c} \\ \vdots \\ a_{1,c} \\ a_{r,c} \times a_{r,c} \\ a_{r,c} \times a_{r-1,c} \\ \vdots \\ a_{r,c} \times a_{r,c} \\ a_{r-1,c} \times a_{r-1,c} \\ a_{r-1,c} \times a_{r-2,c} \\ \vdots \\ a_{r-1,c} \times a_{1,c} \\ \vdots \\ \vdots \\ a_{1,c} \times a_{1,c} \end{pmatrix} \lambda$$

2.5.9 Expectation Minimization Method

In authors in [18] considered an OD request as Gaussian conveyance with a expected value and a fluctuation σ, which are connected via $\sum i = \phi(\lambda_i)^c$ when λ and ϕ are assessed for c = 2 in [18]). The assumed method is drived from the expectation minimization and the contingent assumption work Q is given below :

$$Q(\gamma^{n+1}, \gamma^k) = E(L(\gamma^{n+1}|X^T)|Y^T, \gamma^n \qquad (2.5.24)$$

Where $\gamma = (\lambda, \phi)$ denoting the boundaries to be determined.

The strategy comprises of 2 stages, "Assumption step" and the "Boost step".

In the assumption step, γ and arrangement of interface tallies estimations, and afterward a continuous arrangement OD requests prediction is conducted. The deterministic traffic matrix assessment methodology is applied when an earlier TM and a connection provides the estimation. In [18], the conditions (2.6.6) are compared with the condition given in (2.6.5) for traffic estimation problem.

$$\begin{aligned} 0 &= c\phi\lambda_i^c + (2-c)\lambda_i^2 - 2(1-c)\lambda_i\beta i^k - c\alpha_i^k \\ 0 &= \sum i = 1L\lambda_i^{-c+1}(\lambda_i - \beta_i^k) \\ &\text{Where} \\ \alpha_i^k &= R_{ii}^k + \frac{1}{T}\sum t = 1T(m_{t,i})^2 \\ \beta_i^k &= \frac{1}{T}\sum t = 1Tm_{t,i} \end{aligned} \qquad (2.5.25)$$

Since there are $I+1$ obscure boundaries and $I+1$ non-straight conditions, the issue can be settled utilizing Newton-Raphson calculation as follows:

$$\gamma_{n+1} = \gamma^n - [F(\gamma^(n))]^{-1} f(\gamma^n) \qquad (2.5.26)$$

In above equation Jacobian of $f(\theta)$ wrt θ is given by F.

2.5.10 Moment Matching Method

The traffic matrix estimation method for an IP network was first developed by Vardi. In this, the population of expected values has been carried out by certain rule and properties which estimated the mean values of different sample data. This estimation further utilizes to determined the unknown parameters values. The following equation is used for estimation of unknown parameter.

$$\begin{aligned} E(A_i) = \hat{A}_i &= \sum_c j b_{ij} \zeta j \\ S(A_i, \hat{A}_i) = S_{i\hat{i}} &= \frac{1}{k} \sum_k A_i^k A_i^{\hat{k}} - \hat{A}_i^k \hat{A}_i^{\hat{k}} \\ &= \sum_c j a_{ij} a_{\hat{i}j} \zeta_j \\ &(1 < i < \hat{i} < r) \end{aligned} \tag{2.5.27}$$

In above equation , a_{ij} shows the routing matrix A. Equation can be written in simplified form as,

$$\begin{bmatrix} \hat{A} \\ S \end{bmatrix} = \begin{bmatrix} C \\ B \end{bmatrix} \zeta \tag{2.5.28}$$

S denotes the co-variance matrix of A, and B and C are obtained from the routing matrix.

2.5.11 Bayesian Method

BM is widely used in TM optimization problem. Tebaldi et al [17] implemented BM. In this approach the model problem's algebric equations with Markov chain Monte Carlo Algorithm. In this method the routing matrix can be divided into

$[B_1, B_2]$, where B_1 and B_2 are the non-singular and singular matrices respectively. Generally, the vector X elements are rearranged in such a way that it can be rewritten as $X = [X_1, X_2]$ and it can be derive from,

$$X_1 = B_1^{-1}(Y - B_2X_2) \tag{2.5.29}$$

The rearrangement of B and X can be implemented through QR decomposition [17]. This analysis optimize the full vector X in to X_1 and X_2. The Gibbs sampling algorithm was used by Tebaldi. The most generic Markov chain Monte Carlo algorithm is given by this technique. In this method the element X_2 simulate iteratively from the conditional probability distribution $P(X|X_i, Y)$. Fopr simulations regarding random variable the conditional probability distribution need to be simulated. Moreover the conditional probability distribution is implemented to find random variable for a distribution. The following steps are involved in Metropolis within Gibbs sampling algorithm are,

- Sketch X_2 from the assumed TM with prior TM parameters.
- Claculate prior X_1 and posterior X_1 using

$$\begin{aligned} X_1 &= B^{-1}(Y - B_2X_2) \\ X_{1(n+1)} &= B^{-1}(Y - B_2X_{2(n+1)}) \end{aligned} \tag{2.5.30}$$

- $X_{2(n+)}$ are rejected or accepted with random probabilistic approach.
- Return to stage 1 and iterate.

$$min1, \frac{\prod_{j=1}^{r} P(X1_j^n)}{\prod_{j=1}^{r} P(X1_j^{n+1})} \tag{2.5.31}$$

2.5.12 Kruithof Method

The Kruithof approach [50] is one of the famous methods used extensively applied in telephony networks. This method is used with the availability of the initially available traffic matrix information along with measurement data for given links. The measurement data links provide the traffic flow entering and leaving in a node on a particular network , which are actually rows and columns of a traffic matrix. least square technique and The Kruithof approach were merged together so that estimated traffic matrix should be stable with traffic flows entering and leaving a network [51]. This approach was generalized with KL distance [15] and [52] (difference of two probabilistic distributions) as a a nonlinear optimization.

2.5.13 Neural Networks for TME

Recently Artificial intelligence (AI) and neural network techniques have been introduced in traffic matrix estimation. Several AI neural network (ANN) model, which recognize patterns among link counts and actual OD flows are currently researchers interest. The ANN pattern recognition model use data (link measurements) of network for training and testing to develop a model. The trained data is then used to predict and estimate traffic matrix. Several researchers have been working on this emerging field for estimation and prediction using deep learning mechanisms. Recurrent Neural Network, Deep Belief Network, Long Short-Term Memory Recurrent Neural Network (LSTM RNN) are few notable neural network techniques used for TME. Recent researchers have used the DNN (deep neural networks) to find the solutions for linear and nonlinear relations between the inputs. Accordingly, the neural networks provide solution to routing scheme as well as relationships between links and traffic flows.

2.5.13.1 Artificial Neural Networks

In Artificial Neural Networks (ANNs) is considered Machine Learning (ML) model, its motivation is derived by biological behavior of interconnected neurons. Every Neuron executes a unique task depending on the number of inputs, and interconnection of neurons form a network. We consider a neuron as a node or unit in computer network. Usually biological neural networks do not form a specific structure, where as ANN form a layer based network. Input layer is the first layer, then multiple number of layers hidden layers and finally the output layer is the last layer of the ANN. Deep learning concept arise here if numerous layers are between input and output layers. Figure 2.7 depicts the ANN network having 3 input layer (3 neurons), following 2 hidden layers having 4 neurons , ending with output layer with 2 neurons.

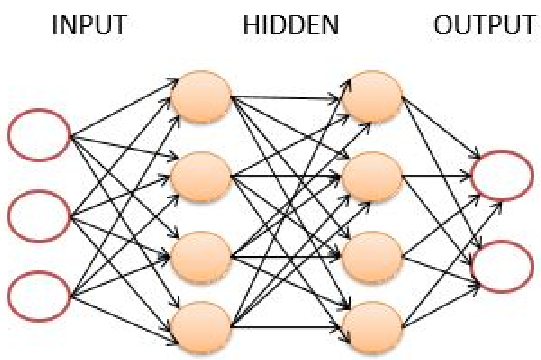

Figure 2.7: Neural networks

In the mid-20th century Artificial Neural Networks (ANNs) were suggested as a learning algorithm with neurons as drived from biological behaviours. The backpropagation algorithm with multiple layer network was launched in 1975 [53]. However, the field of ANN was finally emerged with successful results in 21st century with the advent of GPUs . It was not possible to train complex networks without high speed GPUs , as complex computations are required for training data. Now Artificial Neural Networks (ANNs) can be used for the solutions of non-linear systems which are in complex form. Nevertheless it is still difficult to manage large number of parameters to select and be able to learn efficiently, they can effect learning process and fitting competence.

Computational operations performed in an ANN are defined as:

$$y_i = f_i \sum_{\forall j} W_{i.j} x_j + b_i \tag{2.5.32}$$

In equation above f_i denotes the activation function, $W_(i, j)$ is the weight of this layer, b_i is the bias, output of the neuron is given by y_i and its input to the next layer and x_i is the input from previous layer. To find optimal solution depends upon the learning phase of $W_(i, j)$ and b_i parameters. SGD (Stochastic Gradient Descent) techniques are applied for optimization using multiple optimization techniques and functions. Optimization techniques actually minimize the cost function for the evaluation.

- **Activation function:** This function is applied on the input values of a neuron/node. Mostly used examples are: Linear, sigmoid, rectified (ReLu) and hyperbolic tangent.
- **Learning Rate:** This function is the stepwise iterations to minimize the cost function.It affects the learning rate in terms of training and testing time.
- **L2 regularization:** This function increases the cost of neural parameters and is dependent on the application and this function also avoids over-fitting.

- **ANN size:** To determine the training time and fitting capability the topology of the network has to known; this function determines the size and topology of the network..

- **Other parameters:** There are more parameters that effect the training time such as the optimization algorithm, epoch size (the count of samples performed every iteration),and the max count of iterations.

2.5.13.2 Neural Networks For Traffic Matrix Prediction

Recently Neural Networks (NN) are involved in modeling and predicting network traffic . Because of the efficient learning and adaptive capabilities of NNs they are widely used in many network management applications. Neural network algorithms are able to estimate and predict any function when data relationships are not known [54]. The Neural Network depends upon the observed data instead on the analytical model for nonlinear, non parametric, and adaptive modeling approach. A Neural network is fully determined by the data set that can provide its architecture and parameters.

The interconnected nodes are termed as neurons and all connections are specified by a weight . Neural network consist of number of neurons:

- An input layer,
- One or more hidden layers and
- An output layer.

The Feed-Forward Neural network architecture is widely famous in various fields . In this architecture the data pass through the network in forward direction only shown in Figure 2.8.

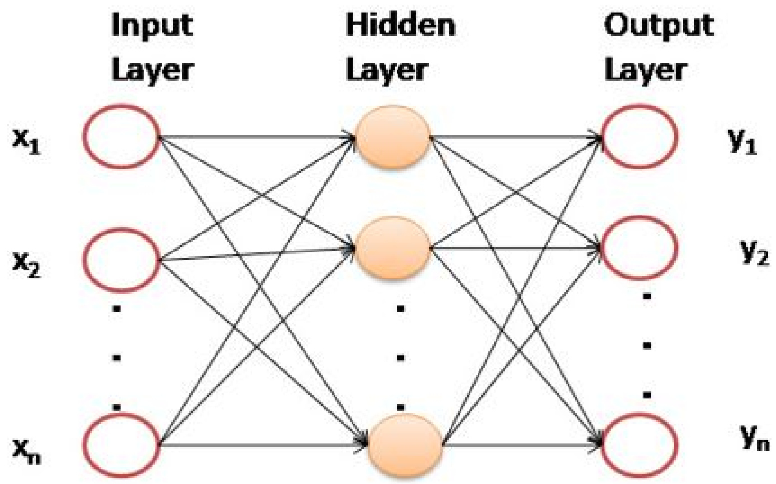

Figure 2.8: Neural Network

A Neural Network when applied as a predictor includes two phases: a) the training phase b) the prediction phase.

For the first phase the training set is applied to the first layer which is input layer and the parameters of the neural network with be regulate according to the parameter setting to develop the output for the specific inputs .

The backpropagation algorithm is mostly used as learning tool which consider backward propagation of the error when weights are constantly changed until output error reaches the desired value. That is why neural networks are easily correlate between input and output corresponding sets.

The testing of Neural network is represented in this phase . A new input which is not the part of training is applied to neural network to estimate the output . This new predicted output is based on the findings of the training phase.

Neural Networks should have at least one hidden layer to predict nonlinear values. more number of hidden layers will increase the complexity of the network , eventually slowing down the training process. The hidden layers and number of nodes (neurons) in each layer are chosen experimentally. For efficient t neural networks the activation function should be sigmoid function.

2.5.13.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a exceptional type of Neural Networks, which has exceptional performance computations and optimization associated with Computer Vision and Image Processing. CNN successful performance is observed in several areas which includes Image Classification , Data feature extraction and Speech analysis and recognition. Recently, Emami at el applied CNN and graph theory to estimate the network parameters and promising results were observed.

The powerful feature of deep CNN is the usage of multiple feature extraction stages that enhances the learning ability of data representation. Big data clouds and hardware technology has coined in interesting research ideas in CNN architectures. Researcher have explored multiple parameters in CNN using various activation and loss functions, optimizers and architectural advancements.

2.5.13.4 Brief History of CNN

CNN was introduced as new class of Neural Networks (NN) by [55] in 1989 in a task related to Machine Vision. Initially CNN were used as best learning algorithms for image classification and segmentation,and retrieval related tasks [56]. CNNs showed prominent winning results and caught the attention in industry as well. Today, high-tech enterprises like Microsoft, Facebook, Google, ATT and NEC have created research centers for new CNN architecture advancements (Deng et al. 2013). The most important feature of CNN is its ability to explore correlation in data in terms of spatial or temporal. CNN architecture is sub divided into various multiple learning layers to perform multiple transformations [57]. These layers are referred as convolutional layers and they extract important features from correlated data points. CNNs when compared with standard Neural Network , it replace the general matrix multiplication reducing the number of weights and consequently reducing the complexity of the network. CNNs are in fact successful deep learning architecture as it has hierarchical layers for successful training. The CNN topology (layers) controls the spatial and temporal correlation reducing the number of network parameters and increasing the performance using backpropagation algorithms. CNNs have become more efficient with advent of computation techniques and competent usage of GPUs.

2.5.13.5 CNN Structure/Architecture

CNN has been widely used in computer vision tasks , its multilayered hierarchical structure gives it the ability to extract features (low, mid, and high-level) which is main reason for the popularity of this algorithm . Figure 2.9 shows a representative CNN structure CaffeNet, which includes 5 convolutional layers and 2 max-pooling layers and 3 fully-connected layers:

- **Convolutional layer:** In Figure 2.9 the yellow block shows convolutional filters. These filters perform the transformation of input images data (feature) into the output data (feature) , shown as blue blocks in Figure 2.9. A detailed first layer convolutional process is shown in Figure 2.10. Consider an image as input for feature extraction using CNN , the convolution filter has three channels corresponding to three RGB color dimension of input image. Each filter channel performs dot product for a specific region to construct a feature channel using a activation function (usually Relu) [58]. Filter channel make certain iterations to cover the entire input image and construct rectified feature output. the rectified feature output will act as input to other convolutional layer. Mathematically , the filter process can be viewed as :

$$a_{i,l+1} = F(\sum w_{i,l} a_{a,l} + b_{i,l}) \tag{2.5.33}$$

Where, w, b represents the weight and bias parameters respectively.

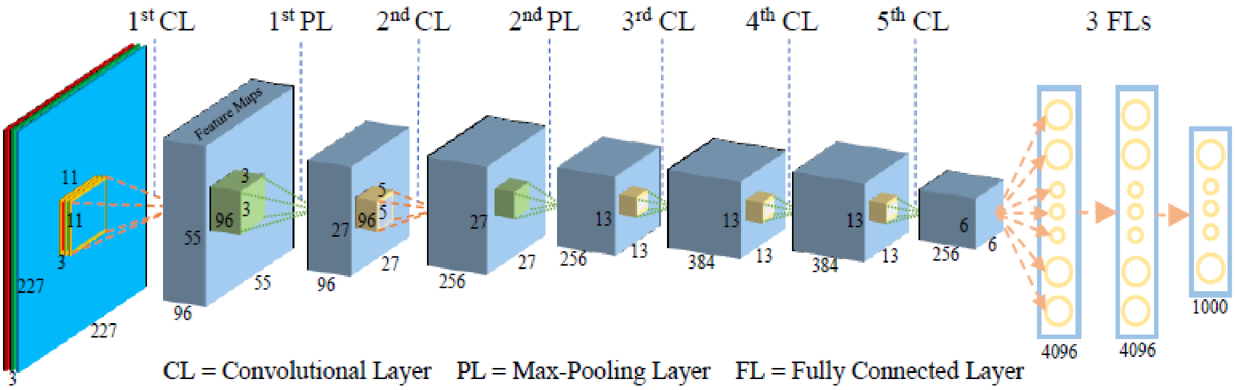

Figure 2.9: CaffeNet architecture

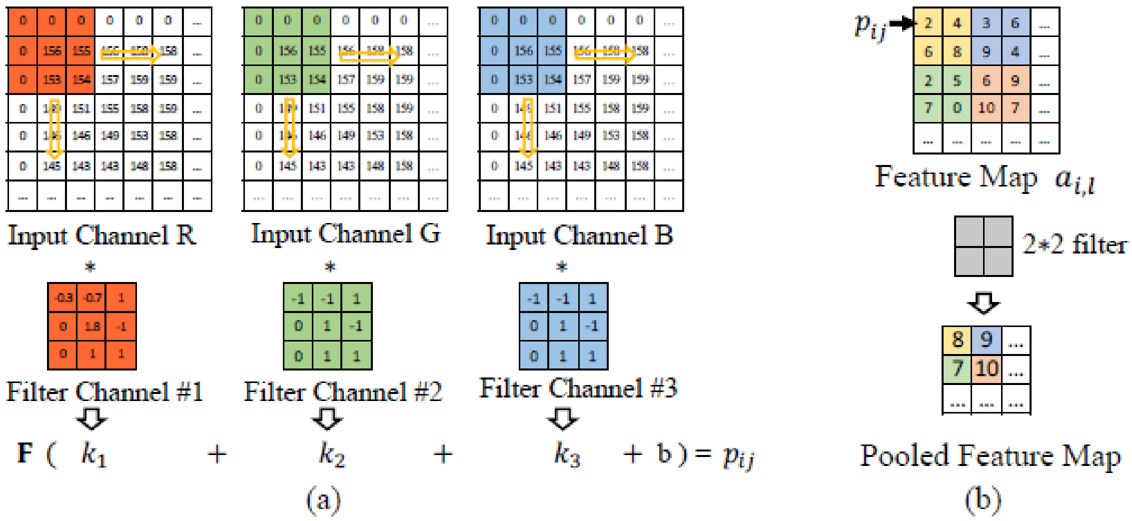

Figure 2.10: Convolution and max-pooling process

- **Pooling layer:** Each convolutional layer is followed by pooling layer as shown in Figure 2.9. Pooling layer performs down-sampling to reduce data dimension of the input data feature. Max-pooling is achieved by applying 2x2 pooling window to select 2x2 region input data feature. spatial size for the output data feature will be reduced by this method. This process will eliminate the redundant data , hence decreasing data processing load.

- **Fully-connected Layer:** This layer achieves a comprehensive feature evaluation based on extraction done by convolutional layers and generate N-dimensional vector , N is the number of the required target as in network parameters N will be number of unknowns. Later, optimizer function such as softmax can be applied for required output data using some prediction functions. Recently, many researchers have focused on the convolutional neural networks optimization and structural algorithms. VGG, GoogleNet and RestNet are example of deeper network structures. However, computational cost, slow training process, large dataset availability are few concerns which can affect the performance and efficiency.

2.6 Deep Belief Networks

Artificial Neural networks has more extended techniques with deep learning methods, in which machine learning model is used to learn deep hierarchical data models [59]. Deep Belief network is very significant deep learning model [60, 61], and its constructed by connecting number of Restricted Boltzmann Machines (RBMs) . RBMs structure and architecture is shown in figure 2.11.

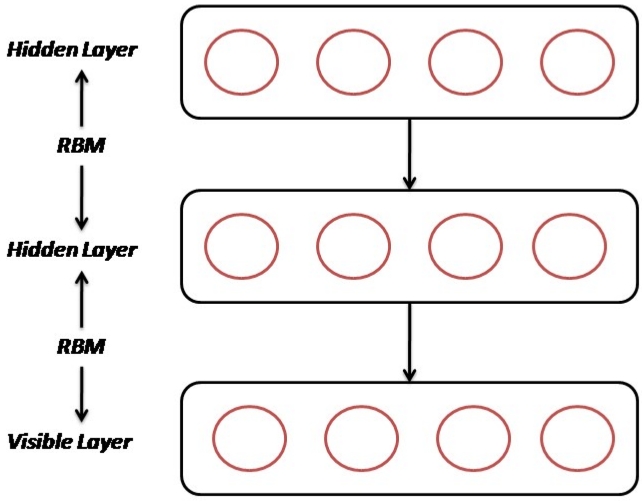

Figure 2.11: Stack of RBMs

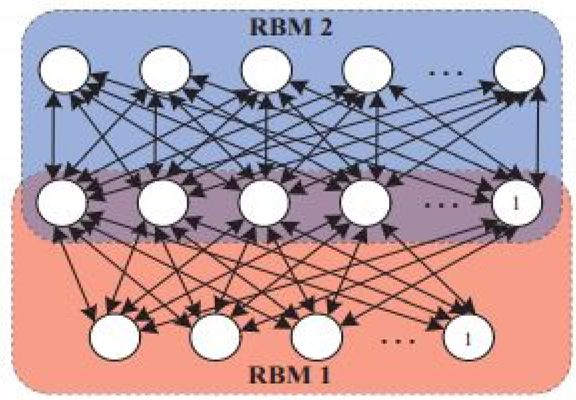

Figure 2.12: DBN architecture

A unidirectional graphical model of RBM is described in [60] having number of visible and hidden layers using v and h notations. In layer structure hidden and visible layers are connected in such a way that peer layers of same units are not directly connected where as all other layers are connected.In RBM archetecture all variables are stochastic using Gaussian and Bernoulli probability distribution [61].

A DBN is combination of numerous RBMs as shown in Figure 2.11 and 2.12), where lower and higher layers are visible to hidden layers only. A joint probability distribution function for all layers is characterized as,

$$p(v,h) = \frac{exp(-E(v,h))}{\sum_{v,h} exp(-E(v,h))} \tag{2.6.1}$$

RBMs is an unidirectional model consisting hidden and visible layers [61]. Each visible layer is connected to all hidden layer. All values of hidden and visible layers in RBMs are stochastic in nature usually, incorporated with Gaussian and Bernoulli distribution.

In above function E denotes the function of energy [61] and generally probability distributions for visible and hidden layers are Gaussian and Bernoulli and function is now defined as:

$$E(v,h) = -\frac{1}{2}\sum_{i=1}^{I}(b_i - v_i)^2 - \sum_{j=1}^{J} a_j h_j - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{i,j} v_i h_j, \tag{2.6.2}$$

I and J are both layers (visible and hidden) [61]. Where as b_i and a_j are terms as biases. However if probability distribution is Bernoulli then for both layers visible or hidden the energy function becomes,

$$E(v,h) = \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{i,j} v_i h_j, \tag{2.6.3}$$

Gaussian and Bernoulli distributions with conditional probabilities for hidden units remains constant, and are given as :

$$P(h_j = 1|v) = sigm(a_i + \sum_{i=1}^{I} w_{i,j} v_i), \tag{2.6.4}$$

Where sigm (z) represents the sigmoid function [61]. Similarly, for the visible units which are Gaussian, they are calculated by

$$P(v_i|h) = Nb_i + N\sum_{j=1}^{J} w_{ij}h_j, 1 \tag{2.6.5}$$

Where $Nb_i + N\sum_{j=1}^{J} w_{ij}h_j$ denotes the Gaussian probability distribution whose mean and variance are $b_i + \sum_{j=1}^{J} w_{ij}h_j, 1$ and 1 [61].

For Bernoulli units, the conditional probabilities for are :

$$P(h_j = 1|h) = sigm(b_i + \sum_{i=1}^{I} w_{i,j}h_j), \tag{2.6.6}$$

Nevertheless, high computational complexity still remain a challenge in deep learning models [60, 61]. A layer-wise greedy strategy for training model must be engaged.

2.7 Conclusion

In this chapter, TM estimation problems were classified into two categories : deterministic approach and statistical approach, few noticeable challenges were highlighted for traffic matrix estimation.

The motivation to find the solution for the challenges in the prediction and estimation of Traffic Matrix lead us to develop a TM estimation algorithm using compressed sensing which is discussed and elaborated through this thesis.

Eventually evaluation of the proposed TM estimation techniques, for both categories using for real back-bone network was developed. We concluded the chapter with a detailed literature review on relevant TM estimation problems.

Chapter 3

Description of Internet datasets

For the performance evaluation two different real world datasets are applied on proposed algorithms. Motivation for the application of real world datasets is to evaluate the capability of proposed method feasibly close to real world cloud networks. Accordingly, real world datasets Abilene [62] and GÉANT [63] are used for the assessment.

3.1 Abilene Dataset

Abilene network is developed by Internet2 community and is considered as high-speed IP backbone network which connects numerous educational institutes in the United States [34]. The network provide scalable and cost effective academic and research environment with hybrid packet and optical network. Abilene network data is accessible to the universities and educational institutes (Internet community) for the technical research work. Figure 3.1 shows the Abilene network topology across United States.

Datasets considered for this thesis are actual Origin-Destination (OD flows) traffic matrix. It has 144 OD traffic flows which were gathered at 5-minutes constant interval for a 24 weeks period. The data was assembled by a researcher at the University of Texas in Austin, Professor Y. Zhang [64] on 12 routing devices. The routing devices with names and locations are tabulated in Table 3.1.

In Abilene network dataset, traffic measurement and routing information is available. Traffic measurements provide end-to-end traffic measurement for the period of 24 weeks long. Each time slots is 5 minute long, consequently there are more than 2000 measurement points. These measurement points are considered as to exact 2016 (7 x 24 x 12)/week.

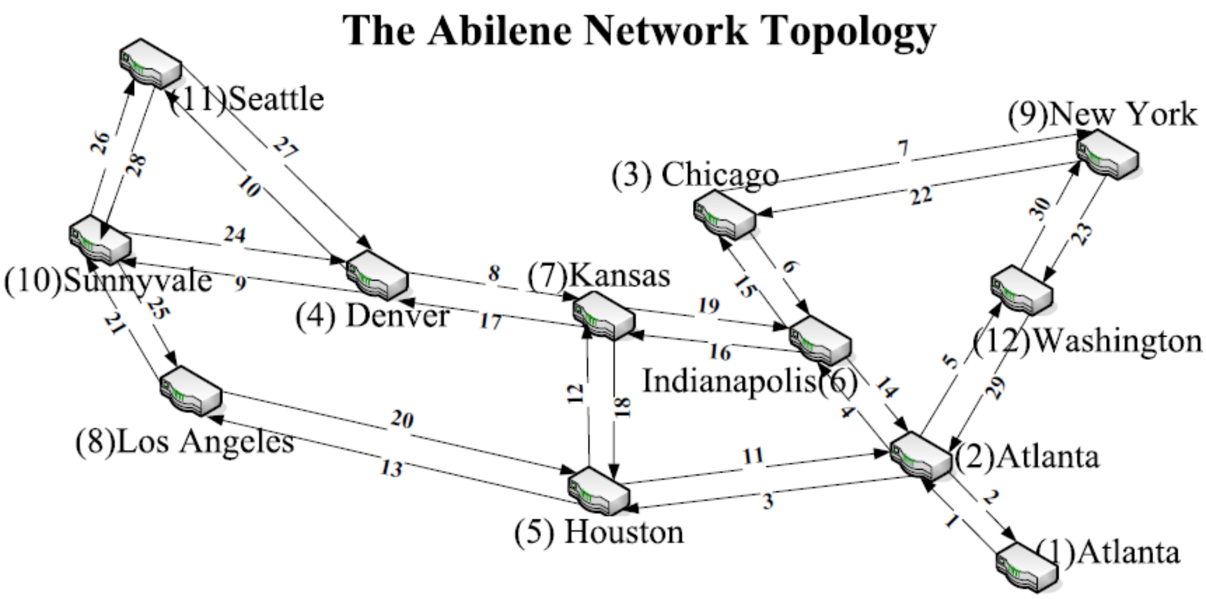

Figure 3.1: Topology of Abilene Network [1]

The Abilene (Internet2) dataset, (a large IP Internet backbone network in the United States developed primarily for research and teaching) was used to test our technique. This Dataset gives us traffic measurements statistics and information

$ATLAM5$	Atlanta GA
$ATLAng$	Atlanta GA
$CHINng$	Chicago IL
$DNVRng$	Denver CO
$HSTNng$	Houston TX
$IPLSng$	Indianapolis IN
$KSCYng$	Kansas City MO
$LOSAng$	Los Angeles CA
$NYCMng$	New York NY
$SNVAng$	Sunnyvale CA
$STTLng$	Seattle WA
$WASHng$	Washington DC

Table 3.1: Abilene Network with Routers and Locations information

regrading routing matrix for the network. The Abilene network, which includes 11 WAN nodes and 41 network lines, the dataset includes a RM (Routing Matrix) and OD flows (TM) for a three-week period. As a result, the Abilene network is responsible for transporting 121 total OD flows. Equation 2.2.1 describes a relationship that makes calculating link counts simple. The byte counts numbers are sampled every week at 5 minute time interval, yielding 2016 byte /samples. As a result, the same amount of sampled values are in network for Link Counts across all 41 linkages.

For our first contribution of this thesis Abilene datasets provide routing matrix information and Origin-Destination traffic flows for a period of three weeks. Abilene network with 11 nodes and 41 directed links is considered for the performance evaluation. With 11 node and 41 links network have total traffic of 121 Origin-Destination (OD) flows. Equation 2.2.1 can easily calculate the link counts.

For second contribution Abilene network is applied for the performance evaluation of our proposed CNN-based traffic matrix elstimation technique. We used two distinct platforms for this, Jupyter Notebook and Spyder python 3.7 on an AMD Radeon R7 240 PC. All of our simulations were thoroughly tested on Colab, which runs TensorFlow and Keras on GPU Tesla K80. The Abilene network for the assessment of this robust technique we consider total of 12 nodes. Since 12 nodes are considered, number of Origin-Destination flows (OD flows) will be squared of nodes i.e 144 OD flows. There are 54 directed links, where 30 links are used for the connection among the nodes, rest links provide the connection with all other nodes over the internet. For this contribution, external node is considered as an independent node. This node is connected with Abilene Network and load traffic provided to this independent node. The dataset for this contribution also consider traffic measurements that provide end-to-end traffic measurement for the period of 24 weeks long. Each time slots is 5 minute long, consequently there are more than 2000 measurement points. These measurement points are considered as to exact 2016 (7 x 24 x 12)/week.

3.2 GÉANT Dataset

GÉANT network dataset provides network traffic measurements to the educational institutes and research purposes. This network is based in Europe as shown in Figure 3.2. GÉANT defines the e-infrastructure for educational, investigational and scientific modernization. GÉANT network is renowned for its incorporated connections and collaborations with high end reliability and unrestricted access to applications. GÉANT network is a significantly large IP network connecting more than 40 European countries in one backbone network. This network provides circuit-oriented services with a high speeds up to 100 GBPS.

The GÉANT network is the major network connecting 50 Million users in the world. This network interconnects 39 national research and education network (NREN) with 10,000 scientific institutes across the Europe and all around the world. This network can reach at maximum speed of 500 GBPS connecting 100 other network all over the world.

The GÉANT dataset is provided in the form of anonymized topology. The data provides origin-destination traffic data for a period of 4 months. This network topology includes 23 nodes with 74 directed links with 529 possible Origin-Destination (OD flows). The datasets is provided in XML format with traffic matrices information. To extract the information from XML format, XML parser was developed on MATLAB to get the traffic matrix and routing matrix information.

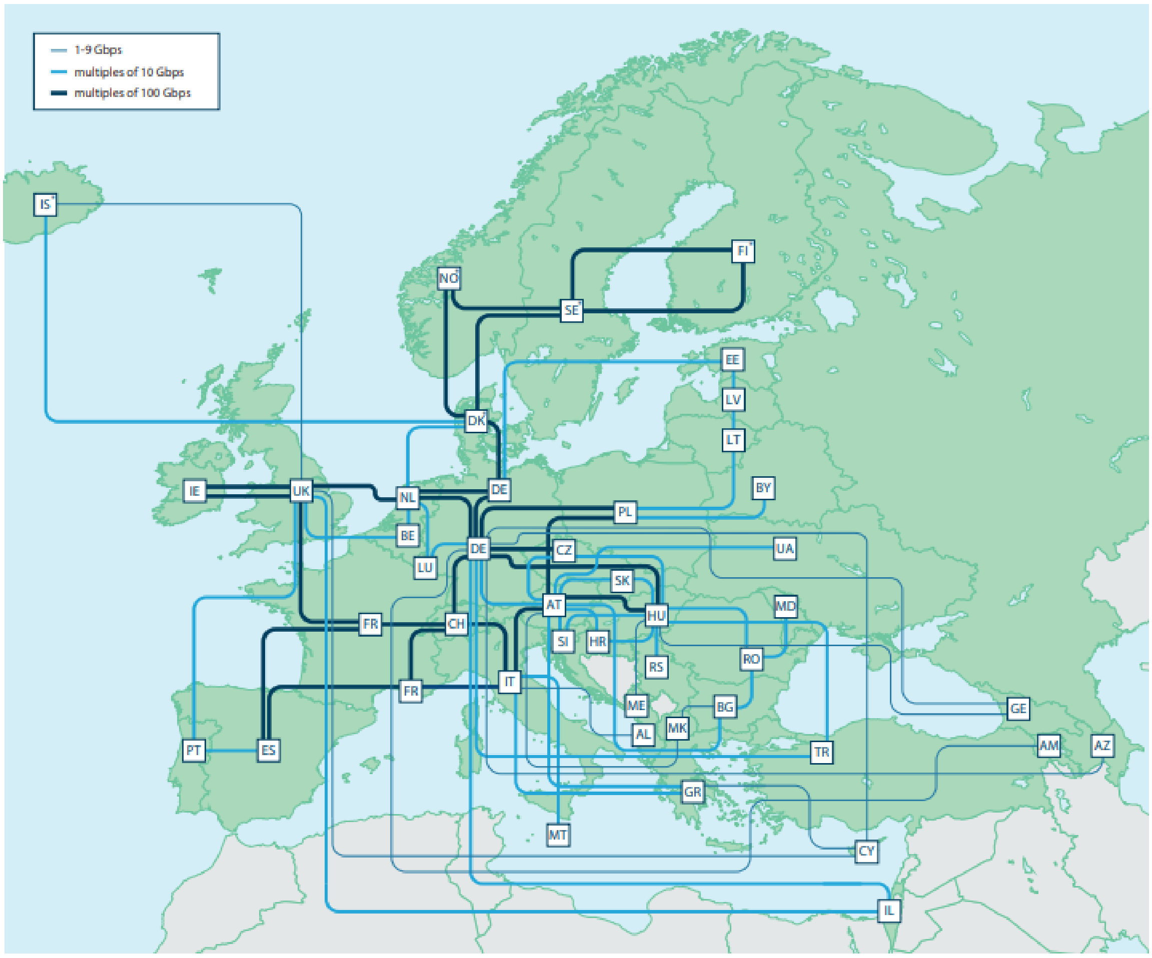

Figure 3.2: GÉANT Network [2]

The datasets (Abilene and GÉANT), 2000 consecutive time samples are used for simulations, the first 500 (for Abilene network) are used as training dataset to predict traffic demands. Similarly first 1500 time samples for GÉANT dataset are applied for training for the prediction of traffic demands.

The GÉANT network is the major network connecting 50 Million users in the world. This network interconnects 39 national research and education network (NREN) with 10,000 scientific institutes across the Europe and all around the world. This network can reach at maximum speed of 500 GBPS connecting 100 other network all over the world.

Chapter 4

TME for Over-dispersion

4.1 Introduction

Traffic Matrix estimation is required for the purpose of network management as well as future network planning and it is very crucial topic of research for researchers. Monitoring the parameters of the network all at once is often not possible because of the complexity of the network. Online cloud services such as Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS), the traffic patterns are diversified its difficult to model them. Numerous scalable techniques are used to estimate network parameters using statistical signal processing methods suitable for linear models for wide area networks. As the complexity of network increases, we observe over-dispersion problem in which the TM elements that show over dispersion, a feature when changes in traffic flows is greater than the mean. Consequently over dispersion becomes more intense problem with small flows (mice flows) and this makes Traffic Matrix Estimation very difficult and complicated.

Several new concepts have been developed for the estimate of network parameters, such as traffic volumes [65–67], and network delays , [68, 69]. Kriging [70–72], Compressed Sensing [63, 73], Cartography [62], and Tomography [65–67], are some of the approaches used to estimate network parameters. The research contribution addressed in this chapter is based on a statistical approach to the distribution of traffic matrix components in order to generate a traffic matrix element expectation given Link Counts data. These strategies depend on preliminary assumption about the traffic flows mean and variance. Many authors have utilized various distributions in their innovative work, such as Gaussian, Poisson, and others [74]. Even if the available prediction of the traffic flows' in terms of variances, in terms of mean or in terms of prior applied distribution is known, the strategies are still largely reliant on that knowledge [19], the 'goodness' of this earlier solution determines the quality of the solution obtained. Tebaldi [17]and Vardi [16] presented preliminary work on estimation of traffic in IP network backbones, demonstrating that few elements of the traffic matrix can be predicted for link load measurements, while the rest can be algebraically computed, limiting the range of the problem to the estimation of fewer source-destination (OD pair) traffic flows. If the routing matrix has a rapidly declining eigen values spectrum, such techniques may be useful [71]. Other significant techniques for traffic matrix prediction and estimation includes Bayesian Learning techniques for estimation matrix developed by Nie [75] and Xiaobo [76]. Some studies [77–80] have looked at the best placement of network monitors for scalable network monitoring. Qazi and Moor investigated the network parameters estimation using network tomography given the underlying network model was calculated with errors [81, 82]. The complexity of network tomography-based estimating techniques due to cloud-based applications, in the presence of heterogeneous traffic has increased the traffic matrix estimation [16], and genetic algorithms have also been applied to the network tomography problem [83]. We establish an optimum solution utilizing statistical method for the excessive dispersion problem in Traffic Matrix estimation. To estimate elephant and mice flows, we proposed a optimization method comprising of two steps.

- In the first stage, larger flows (elephant flows) close to Gaussian approximation are calculated with reliable accurate predictions, while over-dispersed mice flows have higher estimation error.
- The second step adds an extra constraint to the second bounded-value optimization to provide a solution for smaller flows (mice flows) that are over-dispersed.

4.2 Problem Formulation

We re-examine the topic of 'traffic matrix' estimate, which is stated as a linear relationship between:

$$Y = AX \tag{4.2.1}$$

Where Y $\in$R^{mxt} is the matrix of (known) Link Count Measurements, and m shows the number of network links, recorded at time period given by t, binary routing matrix of size m x n is represented by A $\in$R^{mxn}, where n shows the number of source-destination paths for the network, it is assumed over long intervals the routing matrix remains stationary, and that is valid for Internet due to the fact that majority of the wide area paths are stable on the for number of hours ; X is the unobservable (Traffic Matrix).

Researchers have previously modelled the traffic matrix related to various statistical distributions (Negative Binomial, Gaussian, Poisson etc). Statistical prediction approaches do not provide a reasonably accurate solution when faced with the problem of excessive dispersion. This thesis work shows the real world dataset that dispersion causes serious problem for small traffic flows.

As a result, a 2-stage optimization strategy is proposed in which larger flows are predicted with reasonable accuracy in the first step with more conservative estimates for small dispersed flows. A second optimization step with an extra restriction (bounded value) refines the solution for dispersed small flows. Experimental results demonstrate that for ill-estimated flows, prediction can be increased up to four orders of predicted values.

4.3 Traffic Matrix Estimation for Over-dispersion

The proposed two steps algorithm. Initially, the original model in (1) is visualized:

$$Y = AX + \epsilon \tag{4.3.1}$$

Consider a example of a simple network having 3 nodes as illustrated in

Figure 4.1. The Sample Routing Matrix is given as:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{4.3.2}$$

The directed links are represented by the rows of the routing matrix, and the OD pairs are represented by the columns of the routing matrix considered to be sorted in the same way (as mentioned above) for simplicity.

It is considered that an initial OD traffic flows history is present in order to find the direct links among the traffic flows and arrive at an accurate traffic matrix estimation. Calculate the value of a prior matrix using this prior data, as shown in [75]. For the previously present historical data, we employ a model (Generalized Linear Model) to generate the regression co-efficients linking the link flows (predictor) and response (predicted) variables. A Gaussian distribution is assumed for the Link counts and OD flows depending on our analysis of the available datasets. It is to note that this model only has to be run once.

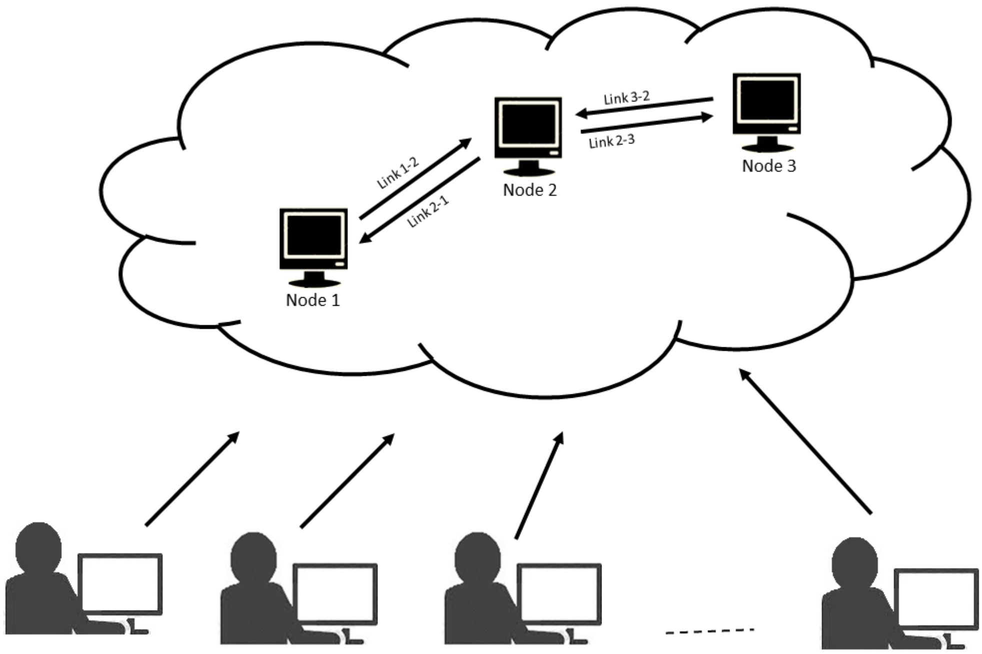

Figure 4.1: A Cloud Network Example

Then, utilizing the previously calculated regression coefficients, we calculate a before hand solution using the present known link count vector. By utilizing the time-based direct links calculated by the GLM without including the space-based constraints, we get the best projected value of the traffic matrix (X_{prior}). The linear model constraints may not be satisfied by this solution. We resolve an optimization issue comparable to the work in [65] after computing the prior.

$$Min_x \|Y - AX\|_2^2 + \|X - X_{prior}\|_1 \tag{4.3.3}$$

Subject to:

$$\begin{aligned} AX &\leq Y \\ X_i \geq 0, \quad i &= 1, 2, ..n \end{aligned} \tag{4.3.4}$$

As a result, the optimizing function seeks to minimize the following:

- Square of errors as in l_2 norm in the linear model
- The difference of the predicted traffic flows and the former GLM-calculated traffic matrix flows.

The mentioned approach results in a solution known as posterior solution ($X_{posterior}$) that integrates the constraints which are based upon space of the model. Afterwards, the recent $X_{posterior}$ using the previously available prior X_{prior} initiating time dependent prediction, suitable for the linear model constraints based on space as well. Considering purpose, it is attempted to pivot all the traffic flow matrix components which are nearer to the straight line $X_{posterior} = X_{prior}$ that is to be considered as new constraint in computing the final solution. The second optimization is given below:

$$Min_x \|Y - AX\|^2_2 + \frac{1}{\sigma^2}\|X - X_{posterior}\|_1 \tag{4.3.5}$$

Given that:

$$\begin{aligned} AX &\leq Y \\ X_{posterior} &\geq \frac{1}{\sigma^2}(mx + c) \\ X_i &\geq 0 \quad i = 1, 2, ..n \end{aligned} \tag{4.3.6}$$

Where σ^2 shows the variance of traffic flow matrix.

The result obtained using the first optimization is referred to as employing a single prior (SP), with alterations proposed in recent research [76], and the total solution after solving the second optimization is referred to as Single Prior-Bounded Value method, or SP-BV. The name for our proposed scheme as two-step optimization as, in the first stage, it ties those TM flows that are reliably estimated with little divergence in time-based prediction with the nearest space-based constraint complying solution. After establishing TM estimates, i.e. $X_{posterior} = X_{prior}$, it finds the left behind flows using regression co-efficients for the straight line ,where for ideal conditions m equals to unity and c equals to 0.

4.4 Performance Evaluation

The Abilene (Internet2) dataset, (a large IP Internet backbone network in the United States developed primarily for research and teaching) was used to test our technique. This Dataset gives us traffic measurements statistics and information regrading routing matrix for the network. The Abilene network, which includes 11 WAN nodes and 41 network lines, the dataset includes a RM (Routing Matrix) and OD flows (TM) for a three-week period.

As a result, the Abilene network is responsible for transporting 121 total OD flows. Equation 2.2.1 describes a relationship that makes calculating link counts simple. The byte counts numbers are sampled every week at 5 minute time interval, yielding 2016 byte /samples. As a result, the same amount of sampled values are in network for Link Counts across all 41 linkages.

The plot of variance vs. mean is shown in Figure 4.2. The variation values for mice and elephant flows are 4 orders of value bigger than the mean values in this plot, indicating that OD traffic volume is very variable. As a result, over-dispersion is a problem that affects both large (elephant) and small flows. It contradicts the typically Poisson distribution assumption, which holds that the mean and variance are equal. Figure 4.3 gives the graphical representation of hyper-parameters of shape and scale. These parameter were modeled using gamma distribution for the given random variables. (α) and (β) are notations used for the shape and scale parameters respectively.

Like gamma distributed random variable, the mean rate of the OD count flows can be supported to a random variable that has normal distribution having similar mean and variance. It can be seen that the shape parameter is five times higher in magnitude (on average) than the scale parameter for large flows, moreover it is seven times higher in magnitude than mouse flows scale parameter. This only adds to our conviction that:

- The Poisson assumption of mean = variance cannot be applied to dispersed flows.
- Larger flows are less deviant than smaller flows (when distribution is Guassian). enditemize

  Likewise, modelling the OD flows not practicable because of the mean values of the OD flows are high. As a result, in the remainder of this contribution's discussion, we regard OD flow methods to be regularly distributed.

  The Traffic matrix correlation is illustrated in Figure 4.4. This demonstrates that smaller OD flows have minimum correlation with one another, whereas larger i.e elephant OD flows have a lot maximum correlation, as would be seen in the actual world. As a result, there are significant causal correlations between some traffic matrix elements. This improves estimation possibilities for the traffic matrix's larger (elephant) flows, near to the genuine points even if are

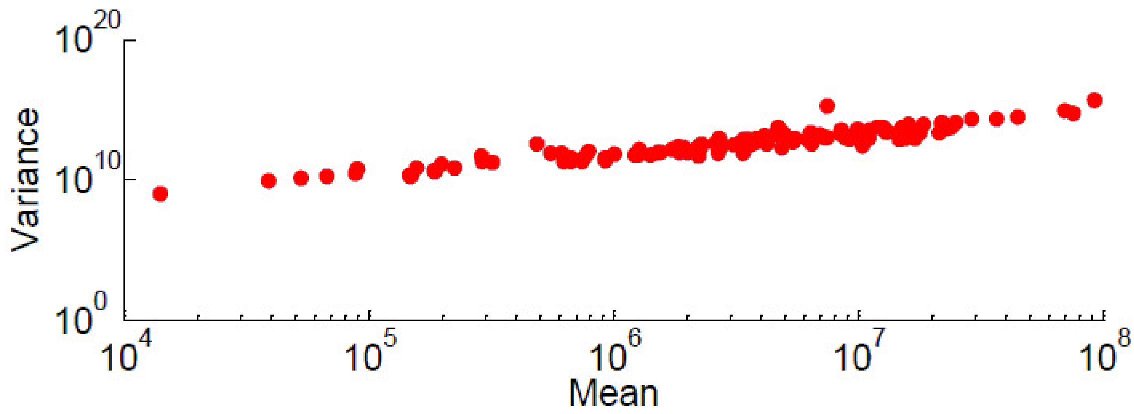

Figure 4.2: Over-dispersion observed in OD flows (OD flow volumes are in Kbytes)

over-dispersed. The over-distributed mice flows are a difficulty; nevertheless, predicting the maximum fluxes, our suggested technique can correctly calculate these.

Figure 4.5 presents the computed slope and y-intercept values for the new constraint in our algorithms used for TM estimation. Figure 4.6 and 4.7 depicts how, the optimization, our algorithm tighten the 'loosely' estimated over-dispersed flows not calculated in the intial stages, bringing them nearer to the ground truth value of the OD flow to satisfy both constraints. The authors of [74] came to similar results, stating that only 33 % of the major TM flows accounted for

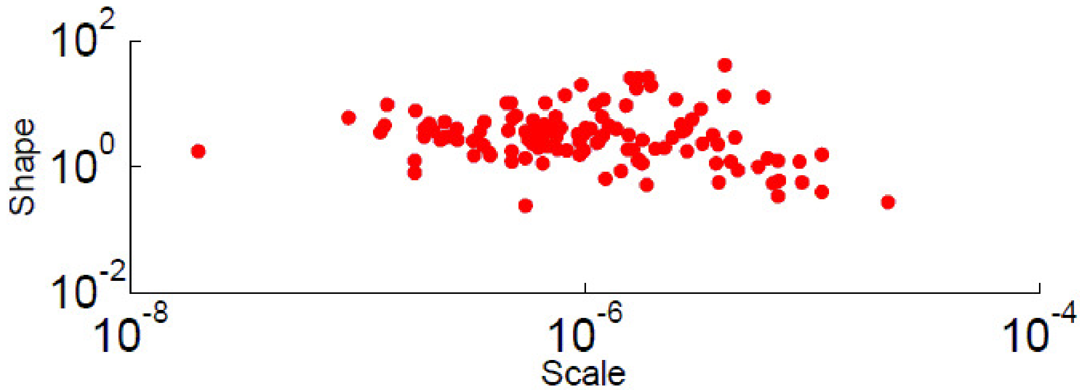

Figure 4.3: Shape and Scale parameters of the Distribution

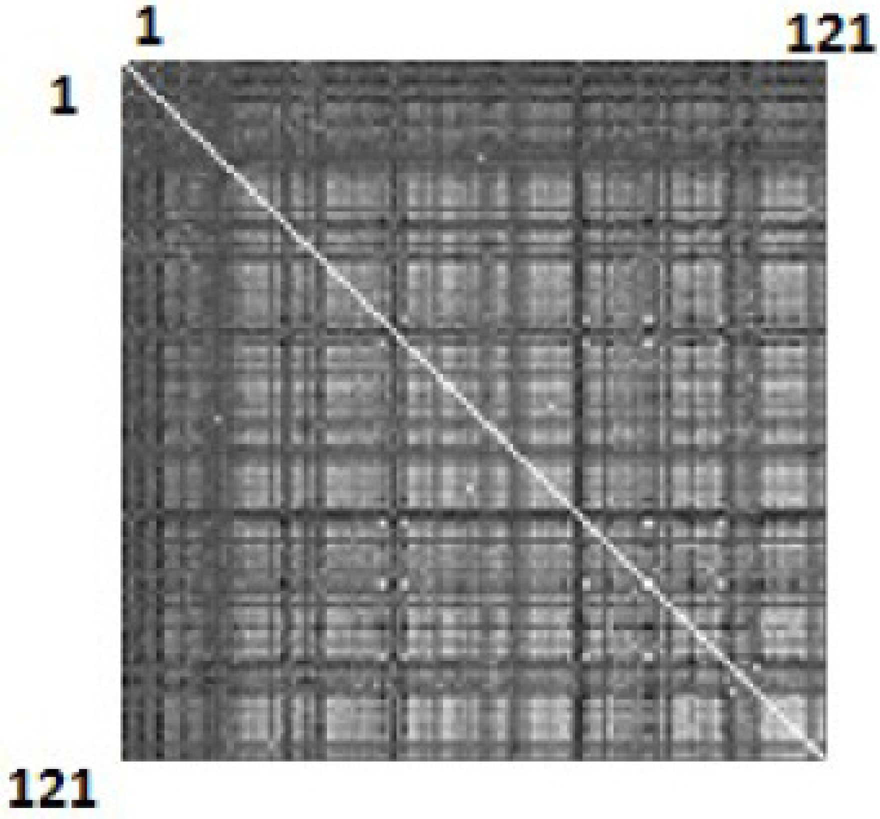

Figure 4.4: Origin-to-destination flow correlation matrix. (Flows are arranged in ascending order, top left showing mice flows correlations the bottom right showing elephant flows)

over 95.0 % of the total traffic volume. As a result, estimating the big OD fluxes is sufficient. In addition, network providers are more concerned with determining the size of larger OD flows. Depending on the assumptions, the authors adjusted the model to make it little under-constrained for smaller flows in order to improve predict larger flows, lowering the count of observable in the traffic estimation. Proposed method is same in that it depends on the notion that elephant flows are more reliable to estimate, which we use to improve our smaller flows estimation.

Figure 4.7 shows the performance gain of our two-stage strategy over a single previous technique for an over-dispersed flows with a change in prediction (time) and constraint (space) specification. The thesis work assumes for training data initial 100 time slots are considered for construction of the initial prior using defined models (GLM), and that link end-to-end measurements at random interval are used to estimate OD flows for 2 specific selected time periods ranging from 100 and 2016. Bigger flows are predicted with good accuracy in the existence of a healthy prior solution, which is available for larger flows only, and this may be utilized to reduce the error prediction of the OD flows which are smaller of the traffic matrix; these numbers demonstrate up to 4 orders of estimation performance improvement.

The plot of real-time tracking of two mice OD flows is shown in Figure 4.8. For the purpose of this study, it is assumed that the first 500 time intervals are utilized as training data over generating the initial solution using linear models, and using Link Count measurements to estimate all OD flows for the next 100 time intervals. Our system estimates smaller, OD flows that are skewed more from normal distribution (than large flows) with higher accuracy, making

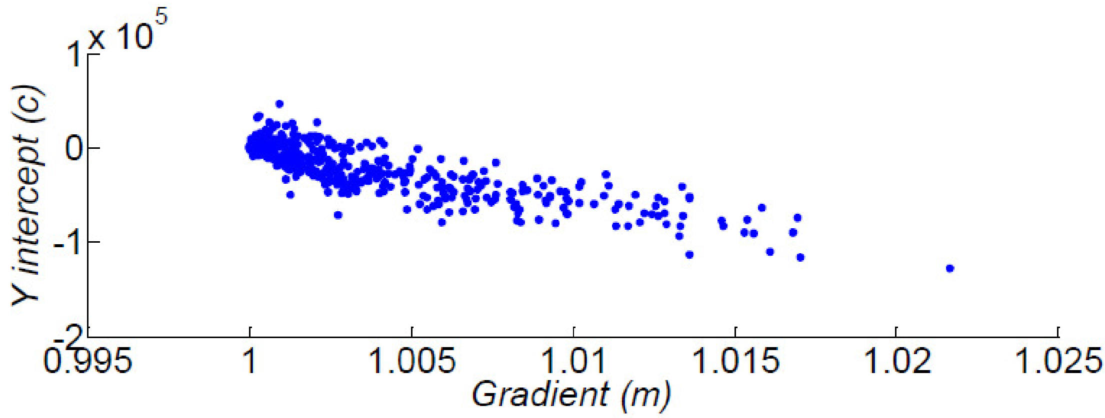

Figure 4.5: Computed slope and y-intercept values

anomaly prediction achievable even for tiny OD flows, as shown in the image. Furthermore, telecom carriers are more interested in getting an estimate to within a certain tolerance level, such as 10 % accuracy for network planning and management purposes. [74].

4.5 Conclusion

A thorough investigation of over-dispersion of traffic flows in traffic matrix is conducted. We present a two-stage optimal approach that focuses on finding the reasonable estimation and prediction of the traffic matrix elements for traffic flows while satisfying constraints for improved estimation and prediction of flows nearer to distributions, and subject to some constraint for small traffic flows estimation nearer to their ground truth values. Experimentation in the real scenario the Abilene Network demonstrates, our method produces accurate estimation results.

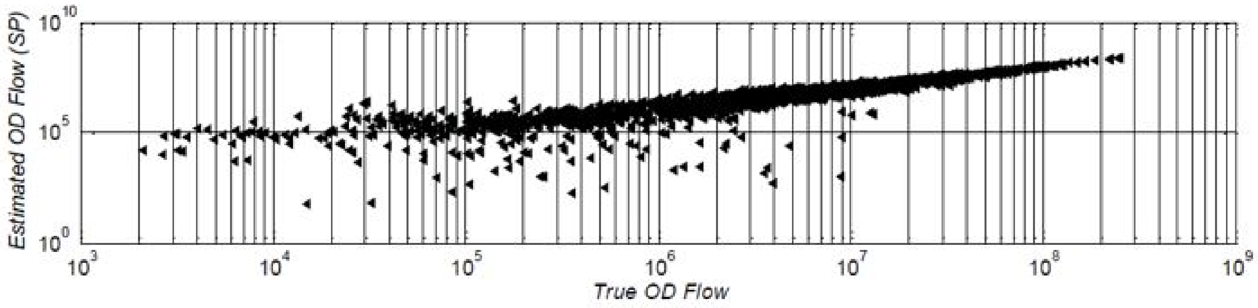

Figure 4.6: The Single Prior (SP) approach to TM estimation

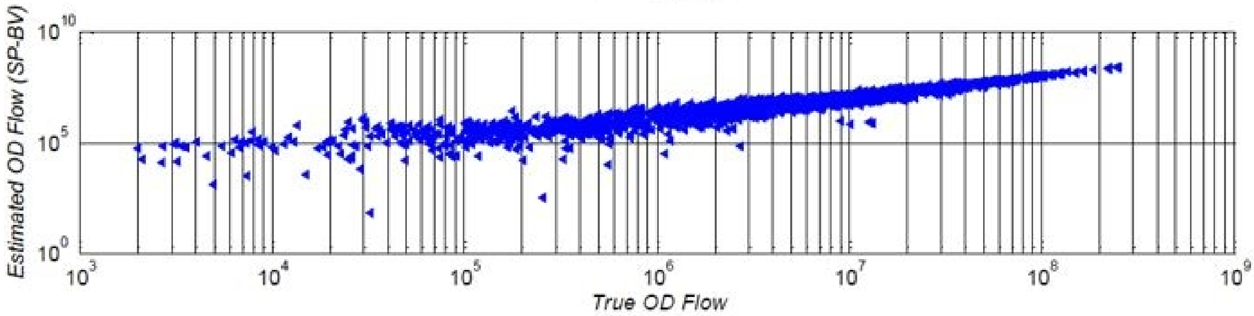

Figure 4.7: SP and SP-BV approach to TM estimation

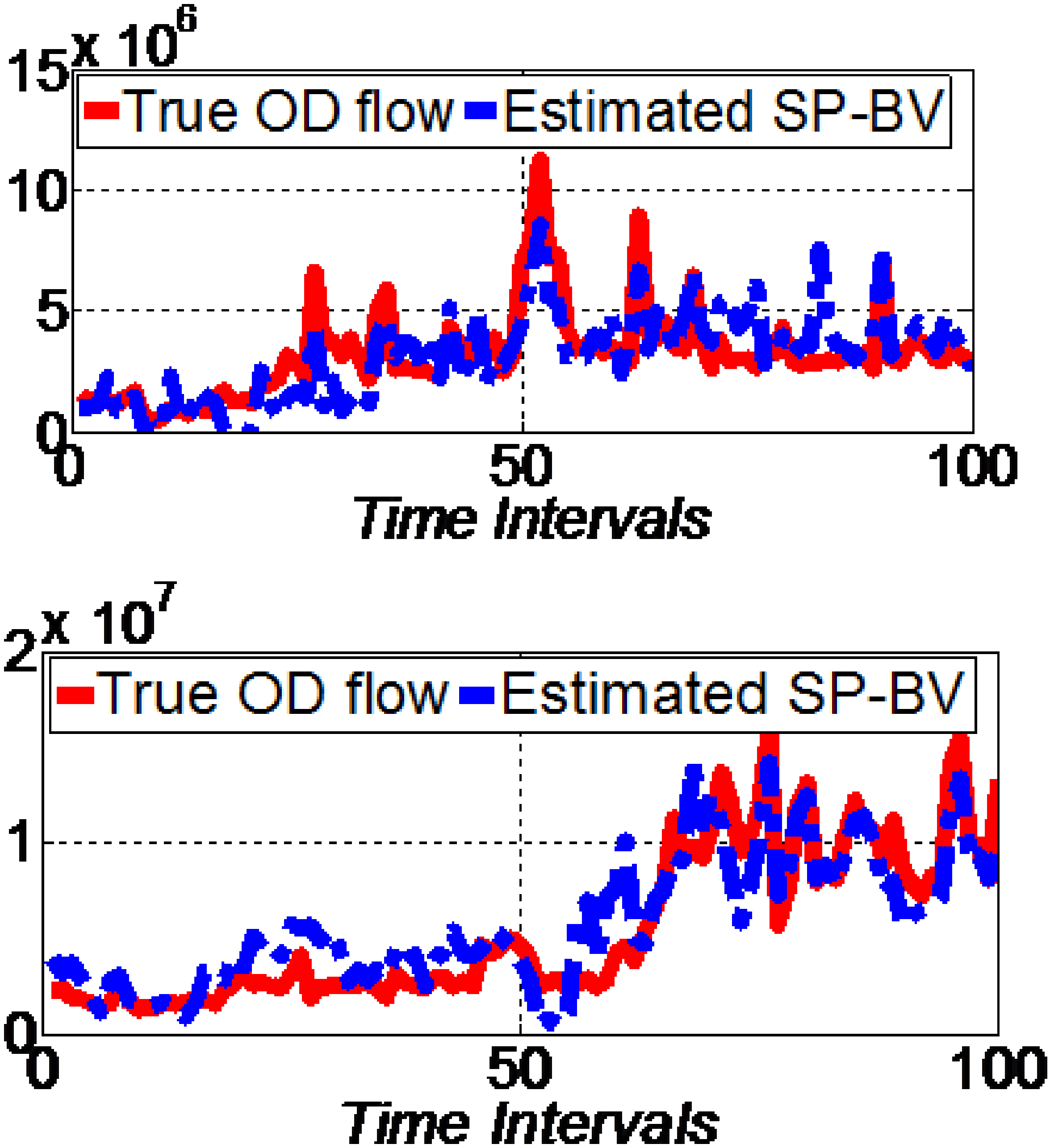

Figure 4.8: A comparison of the SP-BV algorithm's anomaly prediction performance on two mice OD flows showing that the method's performance is acceptable for smaller, over-dispersed OD flows, whose prediction is enhanced by our proposed algorithm.

Chapter 5

Traffic Matrix Estimator

5.1 Introduction

Traffic Matrix estimation plays a significant role in network capacity planning and dimensioning, as it provides traffic flow information between all pair of node in a certain network. Poisson [19], and Gaussian [19] are traffic models which are analytically simple and tractable, however the models provide inadequate accuracy and efficiency decline when applied to large IP networks. Thus, a new Traffic matrix estimation technique is required for the accurate measurements with efficient algorithms for network planning and management function.

Business internet traffic volume in the United States has risen significantly from 45 billion GBs to 112 billion GBs for years 2016 to 2020 respectively, and is predicted to increase to 224.80 billion GBs by 2023 [84]. The biggest contributors to this rise in traffic load are,

- Cloud Computing services, which provide all users with an all-in-one Internet solution in the form of Infrastructure as a Services, Platform as a

Service, and Software as a Service.

- SDNs develop clever technology to ensure Internet users' Quality of Service even in the face of increasing Internet traffic.

Conventional traffic engineering techniques are not applied for Software Defined Networks, which puts a significant amount of strain on the network. Akyildis et al. presented a detailed study of the obstacles [85] of implementing Software Defined Networks. In the absence of a Network Monitoring System, the network is exposed to security threats such as DoS/DDoS [86] and Sybil assaults [75]. Network Monitoring System may detect and stop network traffic irregularities in real time. In fact, one of the primary motivators for the development of network traffic estimating methodologies and algorithms is quick anomaly identification.

With traditional Traffic Engineering solutions circumvented in SDNs, after some time interval before the network's maximum QoS is stretched out, demanding further capacity planning by network administrators. As a result, estimating network parameters is becoming increasingly critical for better and more reliable network capacity planning and management. Many academics have looked at the estimation of dynamic live network properties [87]. Network Delays and [19], Traffic Volumes with packet losses [88] and Network Capacity planning are the significant elements that affect network QoS. Voice Over IP (VoIP) is a network application that require end-to-end measurements for delays to be under a certain threshold. Another network application like capacity planning make use of appropriate traffic shaping techniques, as adaptive flows may be harmed by non-adaptive flows if this is not done [89].

Researchers have recently showed a lot of interest in using neural networks to estimate traffic matrixes. Artificial Neural Networks (ANNs) [90] and Recurrent Neural Networks (RNNs) [3] are two architectures that have been used in the literature to estimate Traffic Matrix. Potential outcomes are noticed employing a new approach for traffic matrix estimate called Conventional Neural Network (CNN); although the findings for traffic matrix prediction were impressive, the forecast will depreciate when real-world data is taken into account. Here are some perfect conditions:

– An assumption that a large amount of training data is available, which is rarely the case in reality.

– An assumption that the training data is complete. It doesn't have any measurement noise, missing measurements, or outliers, for example.

If any of the preceding considerations are violated, CNN utilizing a typical design may deliver average performance, for example. We will get poor results if availability of training dataset is limited and if adaptive learning-rate optimizer like Adagrad as used in [66], because it will timely lower the learning-rate when training progresses over time due to the false or variable availability of huge datasets.

5.2 Related work

The methodologies for measuring network parameters have been extensively discussed in the research literature. Kriging [63], Cartography [91], Tomography [92], and Compressed Sensing [74], are a few notable technique-specific terminology coined by academics. The network parameter estimate approaches are classified as space, time, or spatio-temporal [17] techniques. The temporal component of the m network's parameter is determined by the network training dataset that are available, whilst the spatial part is determined by the information of the network architecture and its affect on the measured values. In these spatio-temporal techniques, the network parameter estimate or prediction algorithm belongs to methodologies like Linear Optimization, Bayesian Estimation, and Expectation Maximization [88].

It is important to note that the correctness of training data is critical for the subject under study in this thesis, referred as traffic matrix estimation. The Expectation Maximum based Estimation technique, for example, employs statistical inference methods such as traffic matrix elements distribution. These later utilized for the generation of an expected value of the elements of traffic matrix based on information from Link Counts. Nevertheless, such methods largely rely on prior or assumed information of the traffic flow mean and variation. Numerous scholars have used various distributions for this purpose, such as Gaussian, Poisson, and others [16]. Even when preliminary estimates of traffic flows of a available distribution are observed, such techniques depends heavily on information of the initial prior [88], that serves as a initial note for algorithms optimize to get a unique solution that satisfies maximum of the problem's space-based constraints. As a result, these strategies rely heavily on the 'goodness' of a available and earlier solution. The idea of implementing a mixed (hybrid) approach, parts of the traffic matrix elements are predicted or computed algebraically/mathematically based on link load measurements, was proposed by Tebaldi et al in his initial research work[17] and Vardi [16] for estimation in large cloud networks. This approach considerably decreases the problem's complexity because it only requires prediction of a less number of Origin/Destination flows (OD traffic flows). Many techniques have been demonstrated to be useful in circumstances where the routing matrix's Eigen spectrum is rapidly fading [71]. Bayesian Learning approaches developed by Nie and Fan [93] are two notable contributions linked to the problem of TME.

Recent research literature reveals that Neural Network techniques for estimating Traffic Matrix utilizing various algorithms produce promising results. For Network Traffic Matrix estimate, Jiang et al. [90] combine the Neural Network technique with time and frequency. Emami et al. [3] proposed a Traffic Matrix Estimation architecture based on Convolutional Neural Network (CNN) one portion of the training data is translated into one larger dimension by taking network's topology into account. It allows for the use of Convolutional Neural Network-based approaches, which are more smart than various kinds of Neural Networks and are designed for picture (2D) datasets.

Qian et al. [72] propose a new method for estimating Traffic Matrix by using just current (partial) or incomplete Origin/Destination flow data without any training data. Recurrent Neural Networks (RNNs) are used in this technique to estimate unobserved OD flows from known OD flow observations.

Based on a dynamic network measurement paradigm, a compressed sensing model Qazi et al [91] for Network Traffic Matrix Estimation is developed by author. Instead of a stationary routing matrix, the model is based on networks traffic demands . This technique demonstrates how Traffic Matrix estimation with tolerable mistakes can lower Link Count measurements even more.

Src/Dst	a	b	c	d
a	0	1.2	0	0
b	4.3	0	0	5.6
c	6.3	0	0	3.2
d	0	7.9	0	0

Table 5.1: Link Adjacency Matrix using Graph Embedding technique

– Demonstration of the TM estimation problem.

– Robust CNN Based Traffic Estimation is developed using a statistical technique for TME

– Performance Evaluation using real world datasets.

– Association of R-CNTME and CNTME performance.

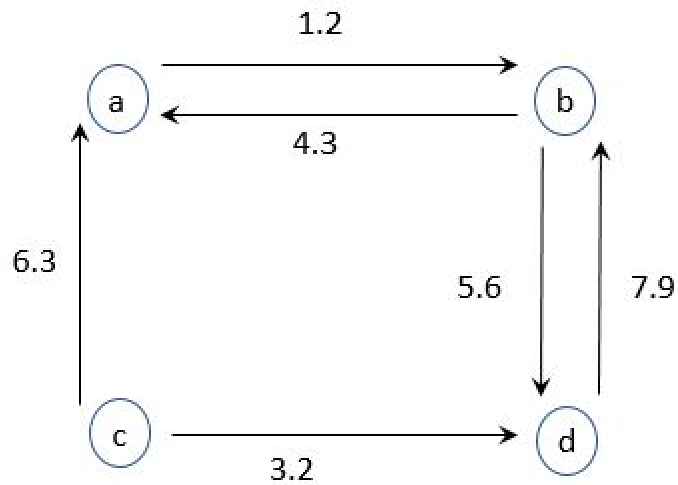

Figure 5.1: Link Adjacency Matrix (traffic volume over directed links is shown)

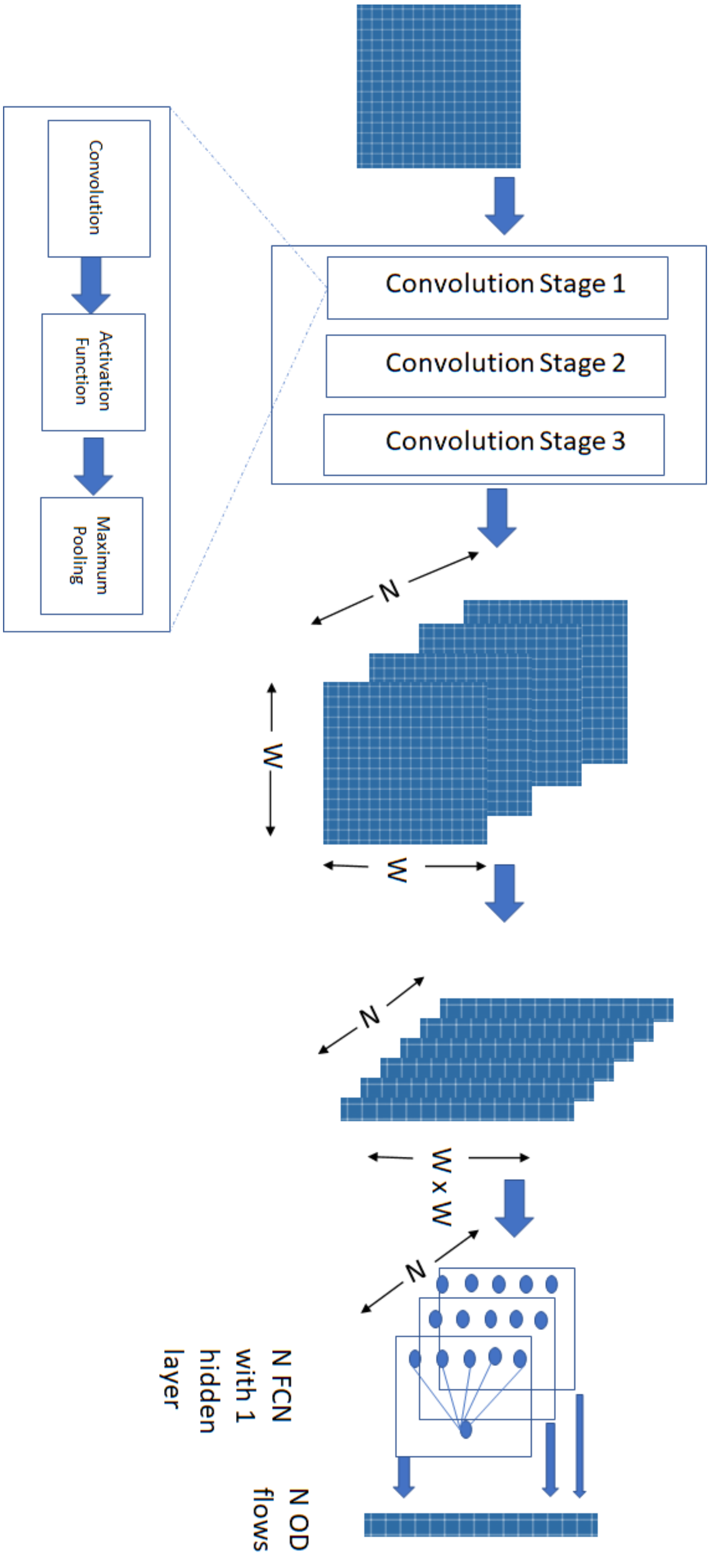

Figure 5.2: CNN-based technique (CNTME) [3]

5.3 Traffic Matrix Estimation Challenge

The Traffic Matrix estimation challenge is generally given by the relationship below,

$$Y = AX \tag{5.3.1}$$

Where, Y $\in$R^{mxn} shows observable Link Count Measurements matrix, and m present the network links, recorded at specified time intervals. A $\in$R^{mxn} shows Routing Matrix, where m presents number of wide area network links and n presents number of nodes. These result in n^2 possible paths between nodes. Before proceeding further, an assumption is made on R (Routing Matrix) that is not changing or remain constant over long duration, it remains stable (unlike usual wireless networks). This is true specifically for the case of Internet, as maximum of the network paths can remain unchanged (stable) for long hours or may be days. $X \epsilon R^{n^2 xt}$ is unknown Traffic Matrix.

The traffic matrix issue, as shown in Equation 5.3.1, is very demanding to solve due to many reasons including; (a) The Routing Matrix A is a 'Big' matrix; and since $m << n$, the system of equations shows ill-posed problem, under-determined and under constrained. It's important to note that a unique solution may not exist for an under-determined system of equations. If we have a reasonable, not ill-posed, or over-determined linear system , on the other hand, then we might have a one-of-a-kind answer. (b) In today's world of dynamic software defined, the premise that A is steady is no longer valid. Furthermore, in today's world, deterministic algorithms are no longer used to determine routing. In Software Defined Cloud Computing Platforms, several user-centric decisions are still treated as network-oriented decisions in order to achieve good load balancing on network servers or the network as a whole. The assumption of a stationary routing matrix is used in first-generation research. As mentioned in the previous section, A has looked into a variety of spatio-temporal approaches. Recent work, such as [3], has explored ways for integrating topology information into the model employing flexibility, such as using link measurements and transforming them into the (L2AMs) link matrix. It loosens the space-based constraints of the issue while allowing exact prediction using a neural network's efficient learning process.

5.4 Robust CNN-Traffic Estimator

Emami et al. [3], in a recent paper incorporating convolutional neural networks and graph embedding, presented an idea to use graph embedding to include the topology of a network into a neural network-based estimate architecture. This allows one dimension to be added to the training data. Traditionally, the training data consists of two 1 D vectors representing L2AMs (link loads) and source-destination fluxes. The graph technique can be utilized to transform the observed data of L2AMs link loads(1D to 2D measurements) by introducing the topology implicitly into the measurement framework. This is accomplished using the L2AM [3], a 2D link adjacency matrix. Figure 5.2 displays a prototype example of this graph embedding technique and an architecture (CNN-based) for learning source-destination flows properties using L2AMs matrices training dataset.

The output of the convolution part, as shown in Figure 5.2, consists of N matrices that are factorized for the generation of N unique feature vectors of size W x W representing to each OD flow, which are applied to N Fully Connected Networks (FCN) with a single layer that is hidden and single layer corresponding to each OD flow which is output layer. We consider L as 13 and W as 9. The proposed architecture (CNTME) outperforms other contemporary approaches; but, it is observed that performance deteriorates if it considered that 2-D training/learning data obtained from topology information of network via L2AMs (link matrix) is sparse (as it has more the number of OD pairs than the number of links) and also contain noise in it.

We propose rewiring the proposed architecture that the ultimate feature set learned is the system as a whole instead each OD flow is considered individually. This approach will dilutes any learning faults that result in inaccurate features being learned for individual OD flows. R-CNTME is the name given to this new suggested architecture. In the following paragraph, we go through this in further depth.

CNTME architecture magnifies mistakes in the estimation of every individual OD flow via N parallel FCN if the 2-D training data has measurements that are limited or have numerous outliers. The features matrices formed after the down-sampled output of the convolution layers are flattened ahead of (rather than after) the fully connected layer, resulting in a single feature vector for the whole system (N OD flows) rather than separate OD flows. This dilutes the learning errors in the CNN stage that are produced by the sparsity of 2-D training/learning data, restricted training data, and probable outliers. Figure

5.3 shows a detailed schematic of the proposed architecture.

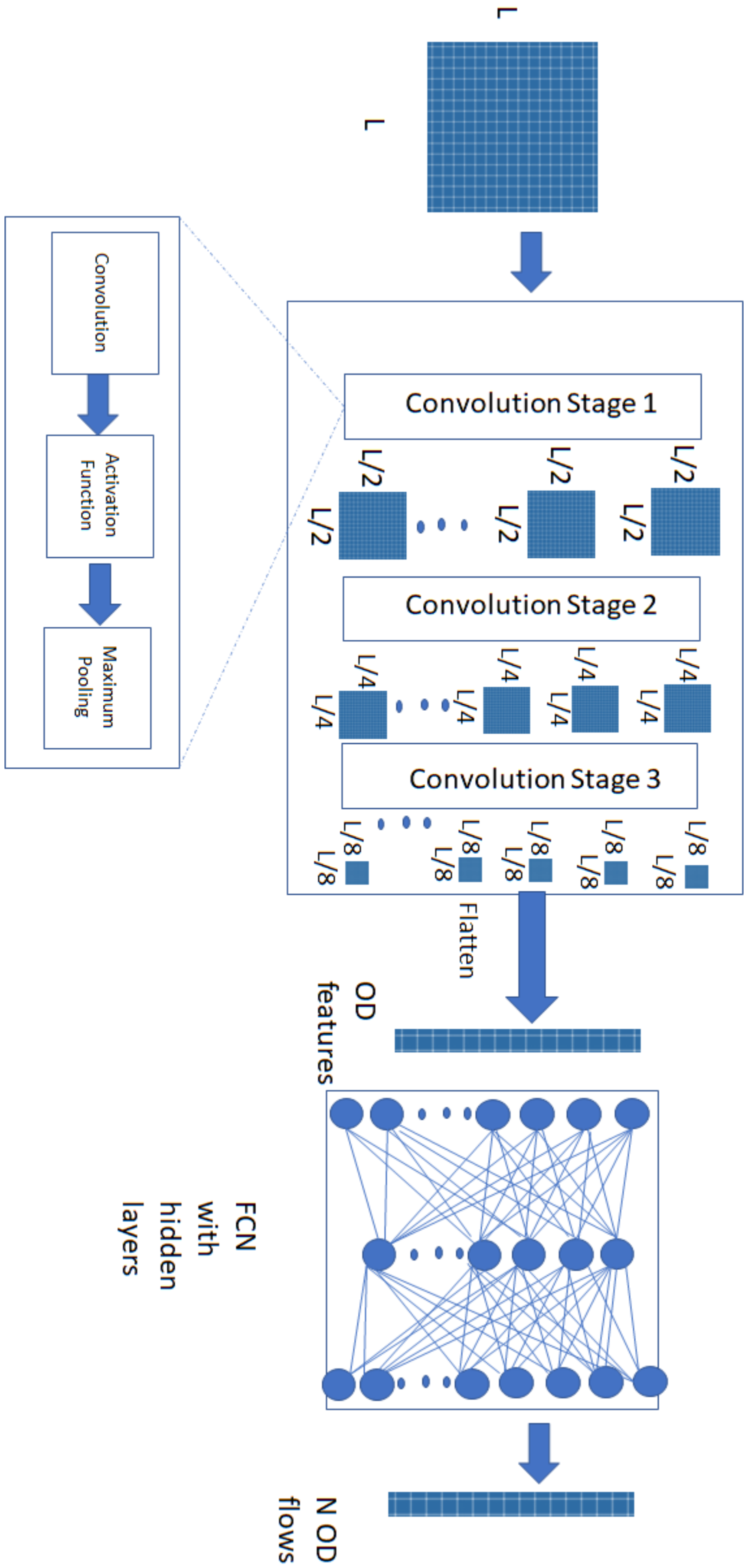

Figure 5.3: Schematic for R-CNTME Estimator

5.5 Performance Evaluation

The evaluation of our proposed approach ROBUST CNN-based can be carried out using two real datasets.

Abilene Network is the first network-backbone in the United States available for researchers. It has total of 12 nodes with 54 links. As a result, there are 144 OD flows (because n = 12 and N =122=144). As a total of 54 links (i.e. m = 54), with 30 of them providing connectivity between the near-neighbor nodes. The remaining nodes form a network that connects all other Internet-connected nodes. The Abilene Network dataset includes end-to-end traffic measurements for 24 weeks. 2016 measurement points are available per week (i.e. 7 x 24 x 12 = 2016) because the measurement time windows are considered 5-minute intervals.

The GÉANT dataset is the second dataset we utilize to performance evaluation of our proposed ROBUST CNN-based approach. This network consist of 23 WAN nodes and 74 directed WAN links. As a result, the GÉANT network transports a total of 529 potential Origin-Destination (OD) transactions. The GÉANT dataset contains anonymized topology information and OD Traffic Data (TM) for a total of 23 WAN nodes and 74 directed WAN links over a four-month period. This dataset contains traffic matrices (TM) in XML format that are measured every 15 minutes. A MATLAB-based XML parser is developed to get the traffic matrix and routing matrix; then Link Counts are calculated using Equation 5.3.1 in the same way as the Abilene Dataset.

We treat the external network as a separate node in our study. We combine it with the Abilene Network's 12 nodes and the GÉANT Network's 23 nodes to handle the load balance between the network (Abilene or GÉANT) and the external network. As a result, the L2AMs for the Abilene network become 13x13 matrices, while the GÉANT network becomes 23x23 matrices.

We utilize Adam optimizer instead of Adagrad for evaluation of of proposed architecture to make a reasonable comparison with Emami et al. [3] Since noise is generated in the training data and Adagrad would be a poor choice for evaluation as it decreases its learning-rates with training simulation epochs. Mean Squared Error (MSE) is applied in algorithms as loss function.

We employ the Softplus activation functions, just like Emami et al. [3], because they don't have the negative values problem when estimated, hence no more steps are required for negative values solutions. For input values x, Softplus generates output values using the function $f(x)$.

$$f(x) = log(1 + \exp^{x}) \tag{5.5.1}$$

As indicated in the preceding section, the suggested approaches' performance was assessed using Abilene network datasets and GÉANT network both based on real captured network traffic. We used two distinct platforms for the performance evaluation , first one is Spyder python 3.7 on an AMD Radeon and second one is Jupyter Notebook which is available online for simulations . The PC's specifications are: Intel Core i7 7770 @3.60 GHz, 32 GB RAM (DDR4). All of our simulations were thoroughly tested on Colab, which runs TensorFlow and Keras on GPU Tesla K80. Unless otherwise indicated, we utilize the initial first week dataset (1500 samples) for training and same week dataset's final 500 samples were used for the testing purposes.

We employ recognized metrics, extensively used in the relative current research work [94] to compare the performance of our approach. The following are the definitions of these metrics:

$$SRE(i) = \frac{\|\hat{X}_i - X_i\|_2}{\|X_i\|_2} \tag{5.5.2}$$

Where X_i shows the OD flows (actual), $\hat{X}_i$ and represent the OD flows (estimated) for i ϵ 1,2,...,n^2 at time t.

$$TRE(t) = \frac{\|\hat{X}_t - X_t\|_2}{\|X_t\|_2} \tag{5.5.3}$$

Where X_t represent the OD flow (actual) and $\hat{X}_t$ shows the OD flow (estimated) for time t and T denotes testing period length.

$$Bias(i) = \frac{1}{T}\sum_{i=1}^{T}(\hat{X}_{i,t} - X_{i,t}) \tag{5.5.4}$$

$$SD(i) = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}((\hat{X}_{i,t} - X_{i,t}) - bias(i))^2} \tag{5.5.5}$$

where $X_{i,t}$ represents the OD flow (actual), and $\hat{X}_{i,t}$ represents the OD flow (estimated) for i ϵ 1,2,...,n^2 at time t, while T represents the length of the testing time. It is found that this architecture performs well given training data

Error Scenario	% of Noise in L2AM Entries	Locations (Noise) in all Time Indices	Distribution Applied
E1	30 %	Random	Gaussian N(0, 25e12)

Table 5.2: E1: Errors simulated in training data L2AM

has no errors. For the L2AM matrix training data, a random noise for 30% of the non-zero entries is created, respectively. It referred as Error Scenario (E1) because a Gaussian distribution with a mean equals to zero and a standard deviation of $5x10^6$ to create noise in 30% locations (random) of the non-zero values in the training data.

5.6 Performance Evaluation with Different Number of CNN and FCN Layers for Datasets

To evaluate our design, we first introduce faults in the 2-D training L2AMs data based on the error scenario E1 presented in Table 5.1 to test the performance of our suggested method. This allows us to test the model while simulating the impacts of sparsity and training data mistakes. Using the supplied hidden layers, we change the number of CNN layers and FCN layers from 3 to 5 as shown in Table 5.3 and 5.4 for Abilene network and GEANT network respectively.

CNN Layers (2x2)	FCN Layers
3	81:144
2	81:100:144
2	81:100:121:144
2	81:90:100:121:144
1	81:144

Table 5.3: CNN layers and FCN Layers for Abilene Network

It can be seen in Figure 5.5 and 5.6 that the performance of SRE varies marginally based on the count of CNN layers and FCN layers for Abilene network. How-

CNN Layers (2x2)	FCN Layers
3	81:529
2	81:100:529
2	81:100:121:529
2	81:90:100:121:529
1	81:529

Table 5.4: CNN layers and FCN Layers in GEANT Network

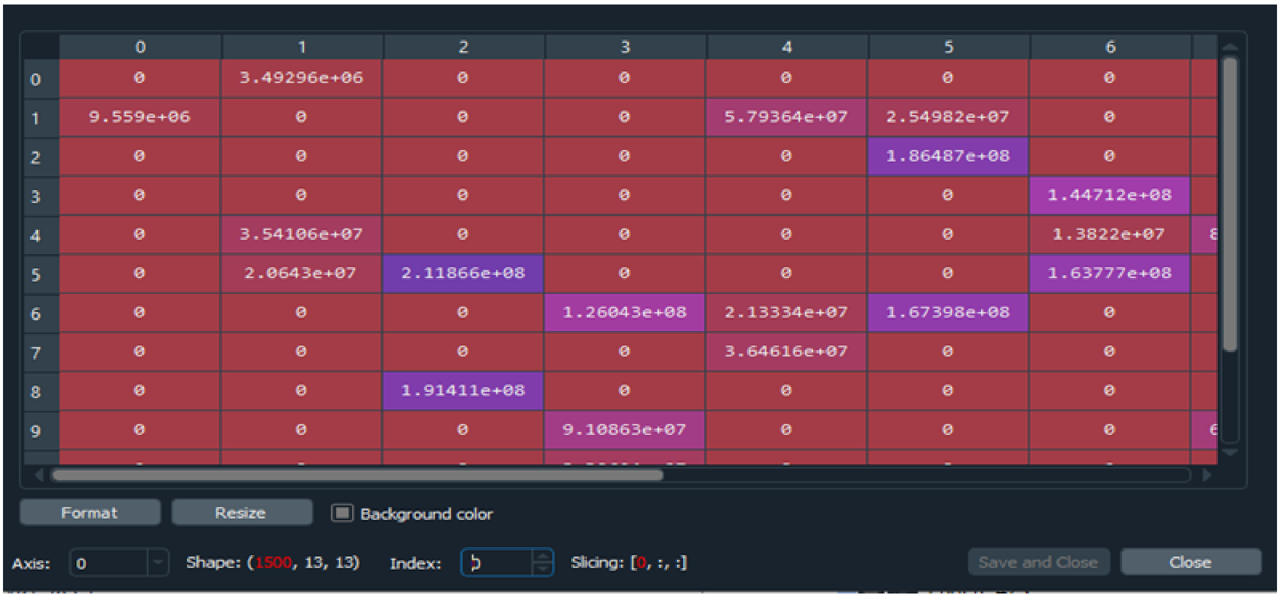

Figure 5.4: E1 Error Scenario in SPYDER 3.7

ever, if the count of CNN layers and FCN layers is raised, we see a significant improvement in TRE performance; the best optimization for TRE metrics is observed with 3 CNN layers and 2 FCN layers. Similar SRE and TRE performance is shown in Figure 5.7 and 5.8 for GEANT network. Best performance evaluation for GEANT network is also for 3 CNN layers and 2 FCN layers. When there is increase in the neurons count in the FCN's hidden layers, it promotes overfitting, while having less neurons in the hidden layers is going to create underfitting problem.

As a result, the number of layers (CNN and FCN) cannot be raised imprecisely, because network anomaly prediction necessitates a good estimator in both di-

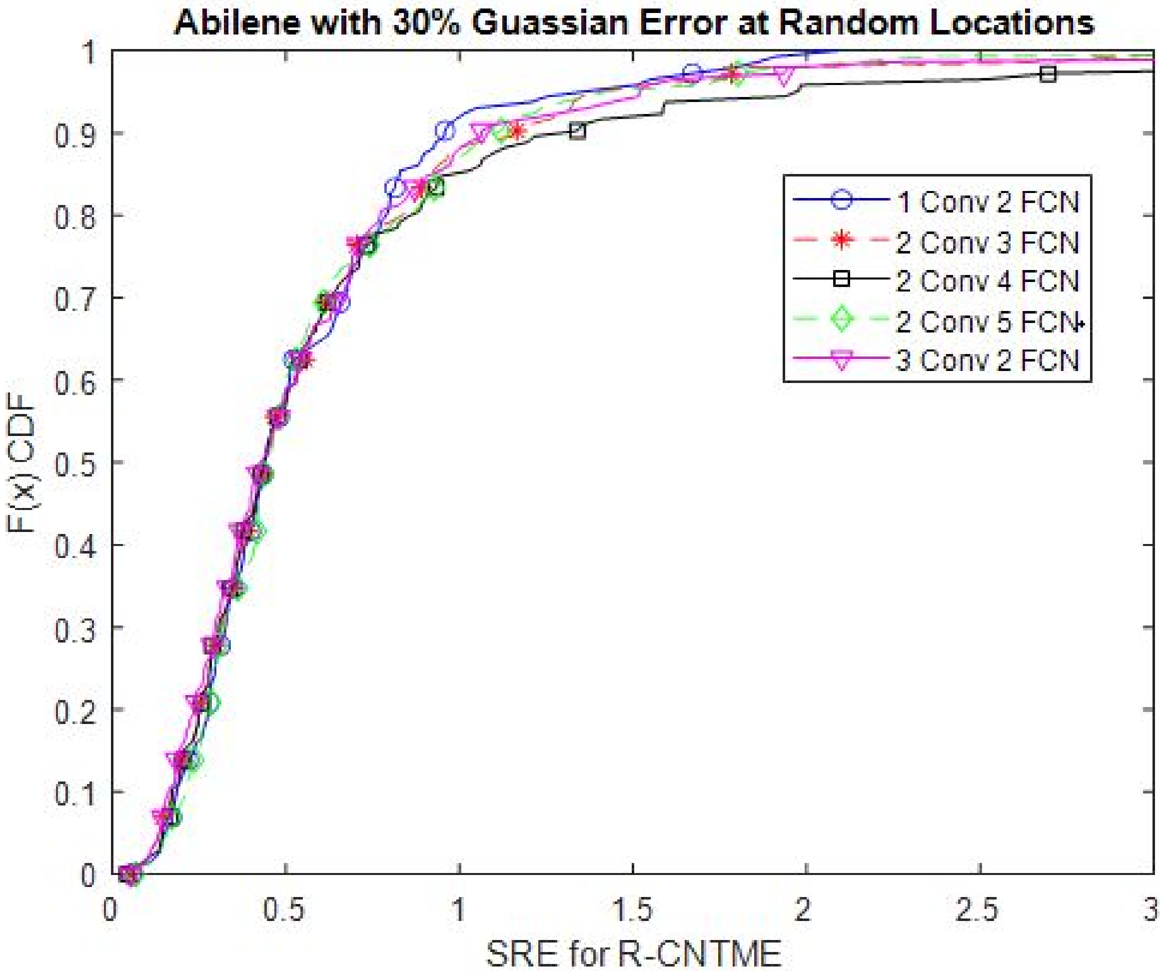

Figure 5.5: SRE of CNN and FCN variation in Abilene Network

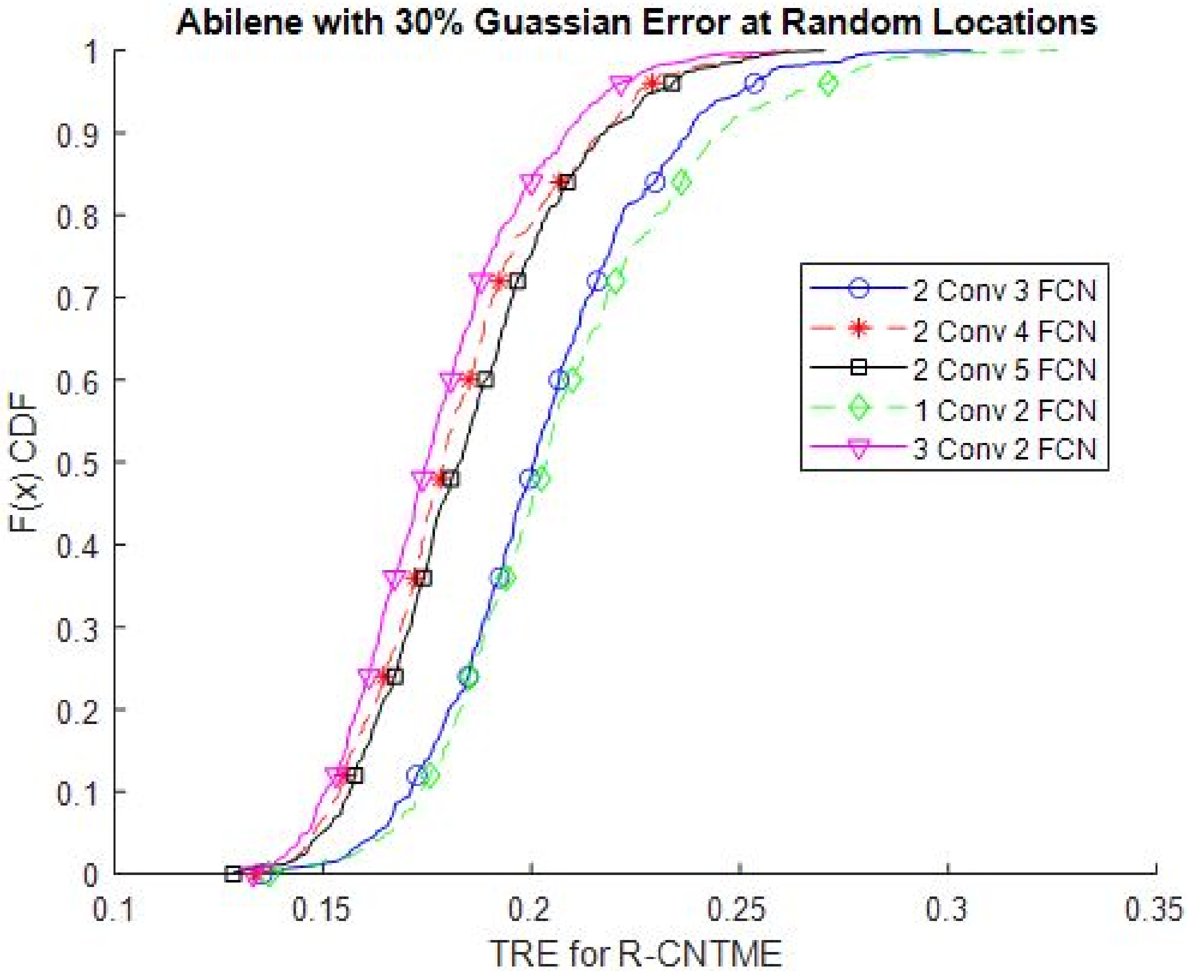

Figure 5.6: TRE of CNN and FCN variation in Abilene Network

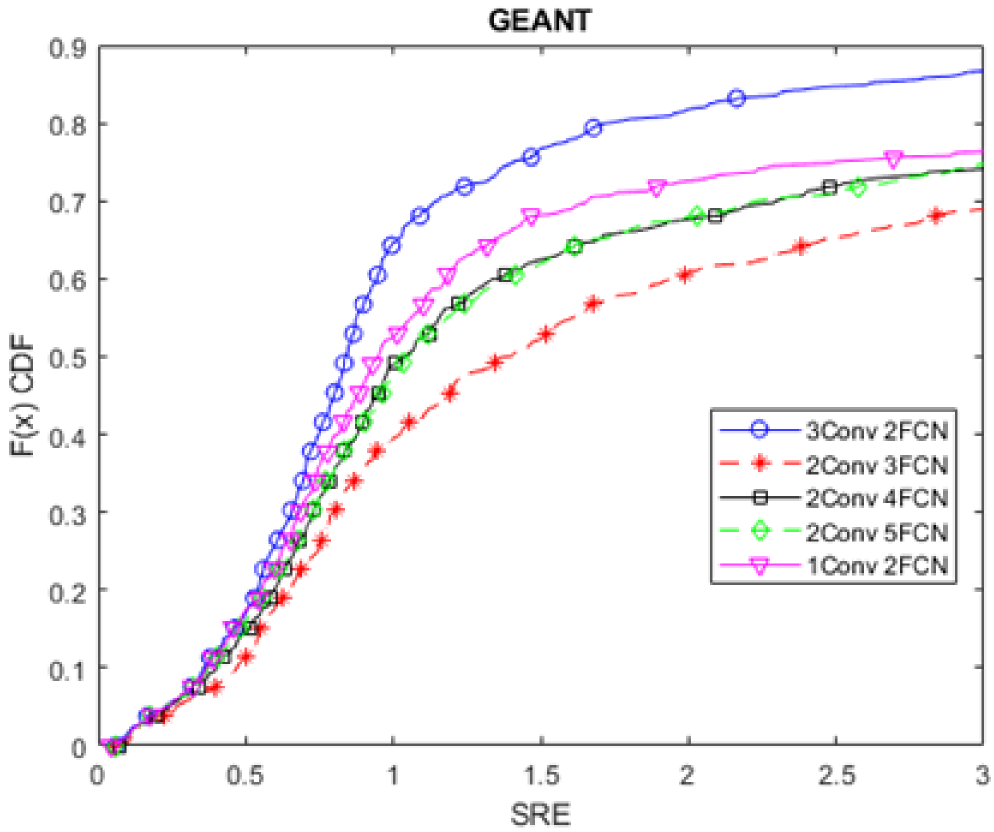

Figure 5.7: SRE of CNN and FCN variation in GÉANT Network

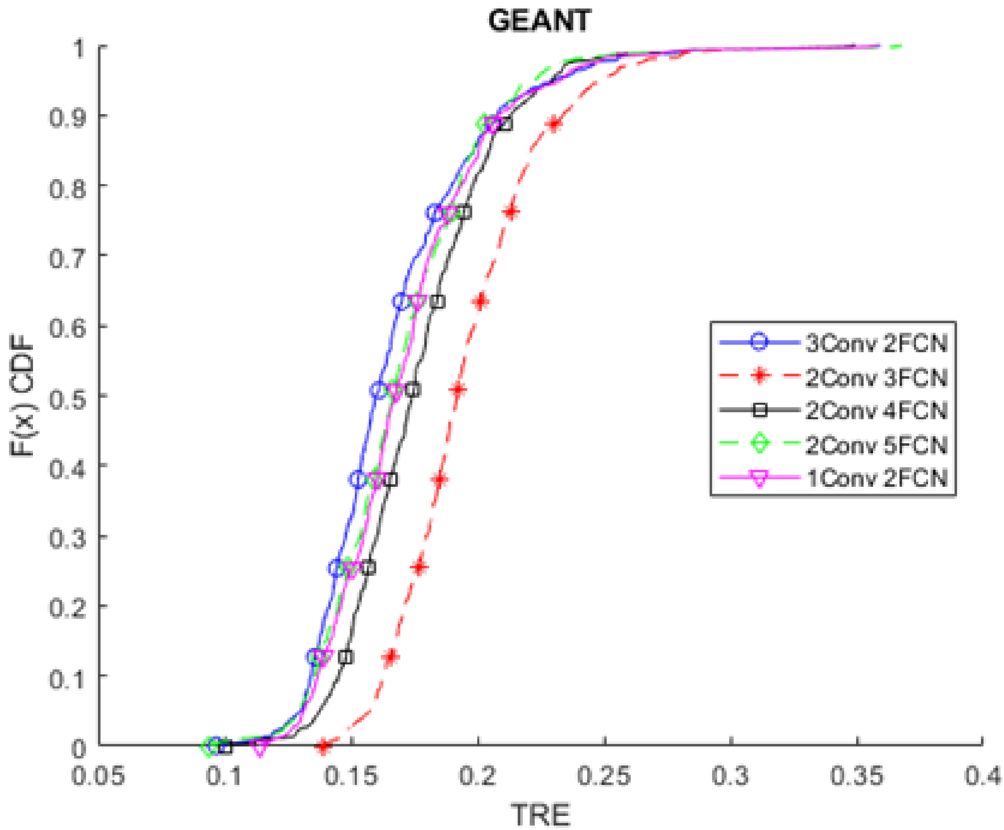

Figure 5.8: TRE of CNN and FCN variation in GÉANT Network

mensions (temporal and spatial). The additional hidden layers in the FCN portion may increase the system's SRE performance at the expense of TRE values, as estimator performance degrades on datasets due to problem of overfitting. Quick anomaly detection is one of the prime factor for the designing of new algorithms and systems in the Network Traffic Estimation Problem.

5.7 Performance Evaluation of R-CNTME and CNTME

5.7.1 Training Data Sparsity Impact

On the Abilene Dataset, we generated multiple sparsity factors for this experiment. The topology is synthetically modified to have twelve nodes (12), however the linkages is dictated by the factor of sparsity. The factor of sparsity is the total zero entries numbers divided by the maximum number of entries in the (link load adjacency) L2AM1 training dataset. We make sure that the topology is intact, with a path connecting individual OD pairs. The OD flows among the 144 OD pairs are identical to those of the Network (Abilene), but connection numbers are modified according to the simulated topology with various factors (sparsity). The Figures 5.9 and 5.10 presents the experiment results and outcomes. When the sparsity factor is decreased to 0.20, superior response is shown in R-CNTME TRE of 0.1870 or less in 90% of situations, whereas CNTME has a value of 0.270. When the sparsity factor is increased to 0.6, the prior values increases to 0.20 and 0.290, respectively. R-CNTME and CNTME demonstrate minor degradation as sparsity is raised in SRE graphs.

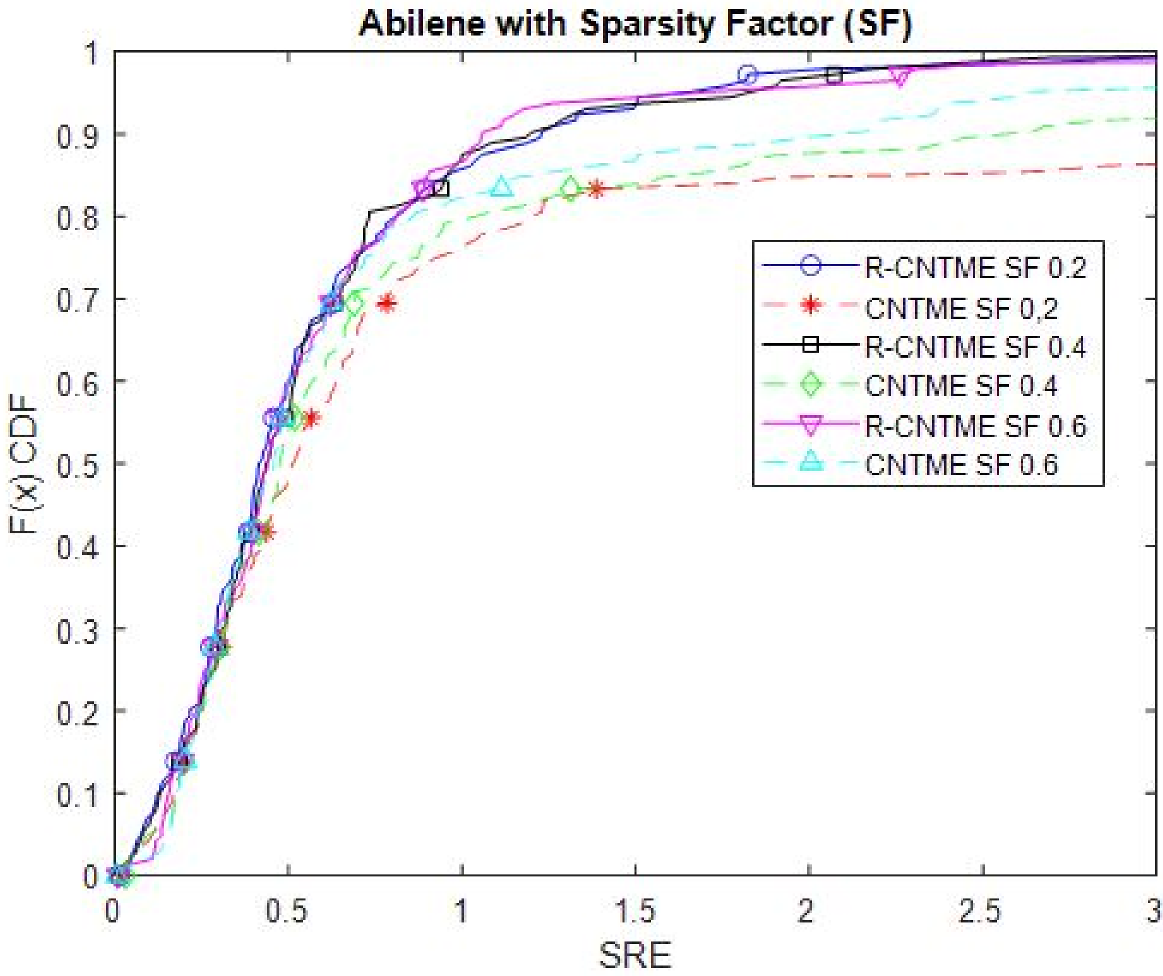

Figure 5.9: SRE Performance Evaluation and Comparison of R-CNTME AND CNTME with different sparsity

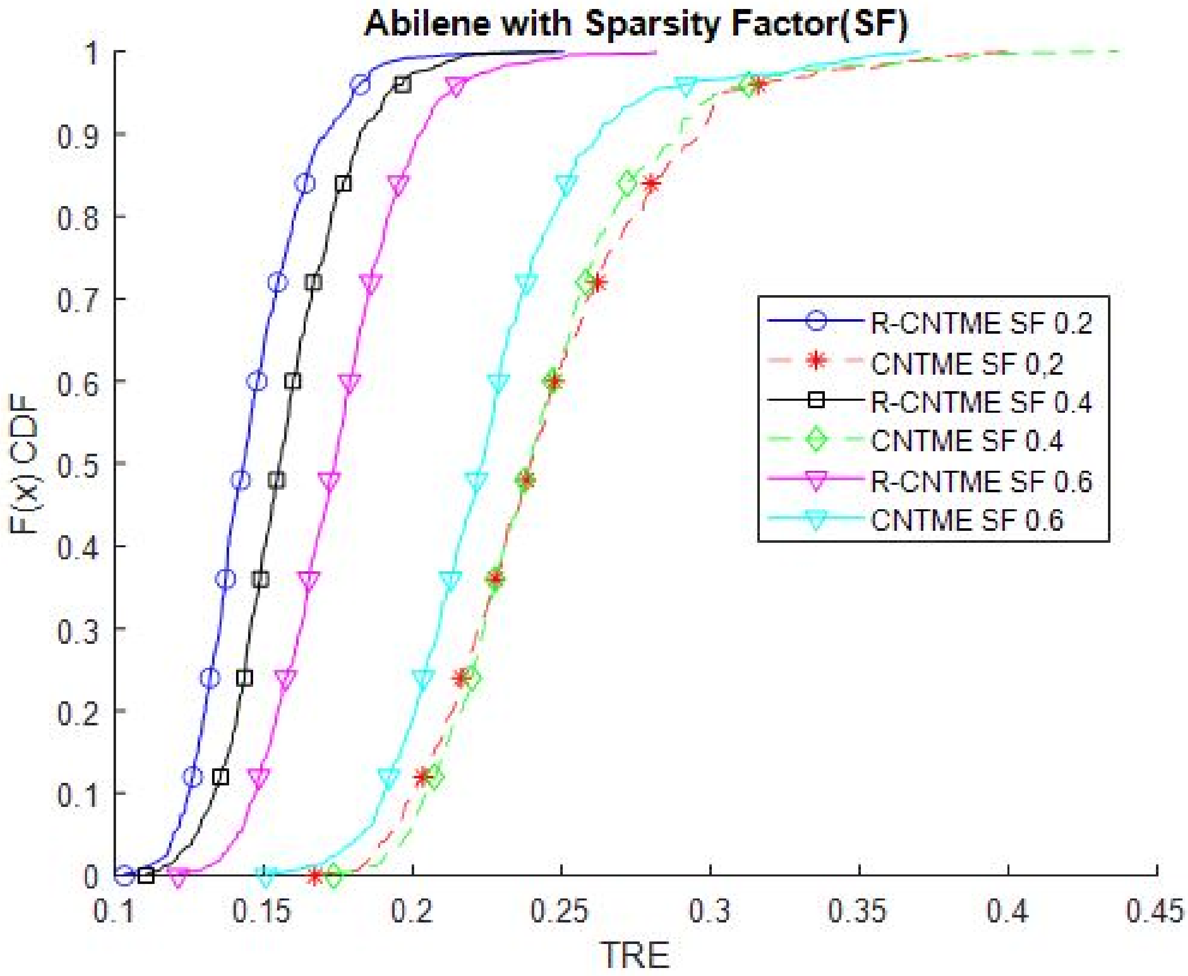

Figure 5.10: TRE Performance Evaluation and Comparison of R-CNTME AND CNTME with different sparsity factor

5.7.2 Training Data Size Impact

Figure 5.11 and 5.12 shows the SRE and TRE performance at variable training and test data sizes respectively. With increase in the training data size, R-CNTME's performance improves, as seen from TRE results. 90% of R-CNTME TRE values are ≥ 0.19 based on training data of 1500; which further drops to 0.2 when the data is decreased to 300. The TRE performance of CNTME is worse across all training data sets, and it also exhibits unstable behavior as a result of the model influenced by the problems like underfitting and overfitting as discussed before. It also demonstrates R-robustness CNTME's in the face of restricted training data availability. Given any ranges of training data size, R-CNTME outperforms CNTME in terms of SRE.

5.7.3 Performance Evaluation and Comparison

Figure 5.13 and 5.14 presents the CDFs for the clean Abilene dataset, whereas Figure 5.15 and 5.16 shows the CDFs for clean GEANT dataset. Both datasets are clean without any artificially generated errors effects or sparsity factor in the L2AM training data. For both datasets the TRE and SRE measurements show that R-CNTME has the best performance. The Abilene dataset performance presents that in 95% of cases, R-CNTME has an SRE value of 1.5 or less, compared to 82% for CNTME. Similarly, R-CNTME's performance indicates TRE values of less than 0.20 for 90% of the predictions. CNTME has a comparable value of 0.24. For GEANT datasets 80 % cases in RCNTME reflects SRE value of 1.6 or less , however CNTME shows less then 60 % cases for SRE value of 1.6. R-CNTME's performace is also superior for TRE values ,showing value less

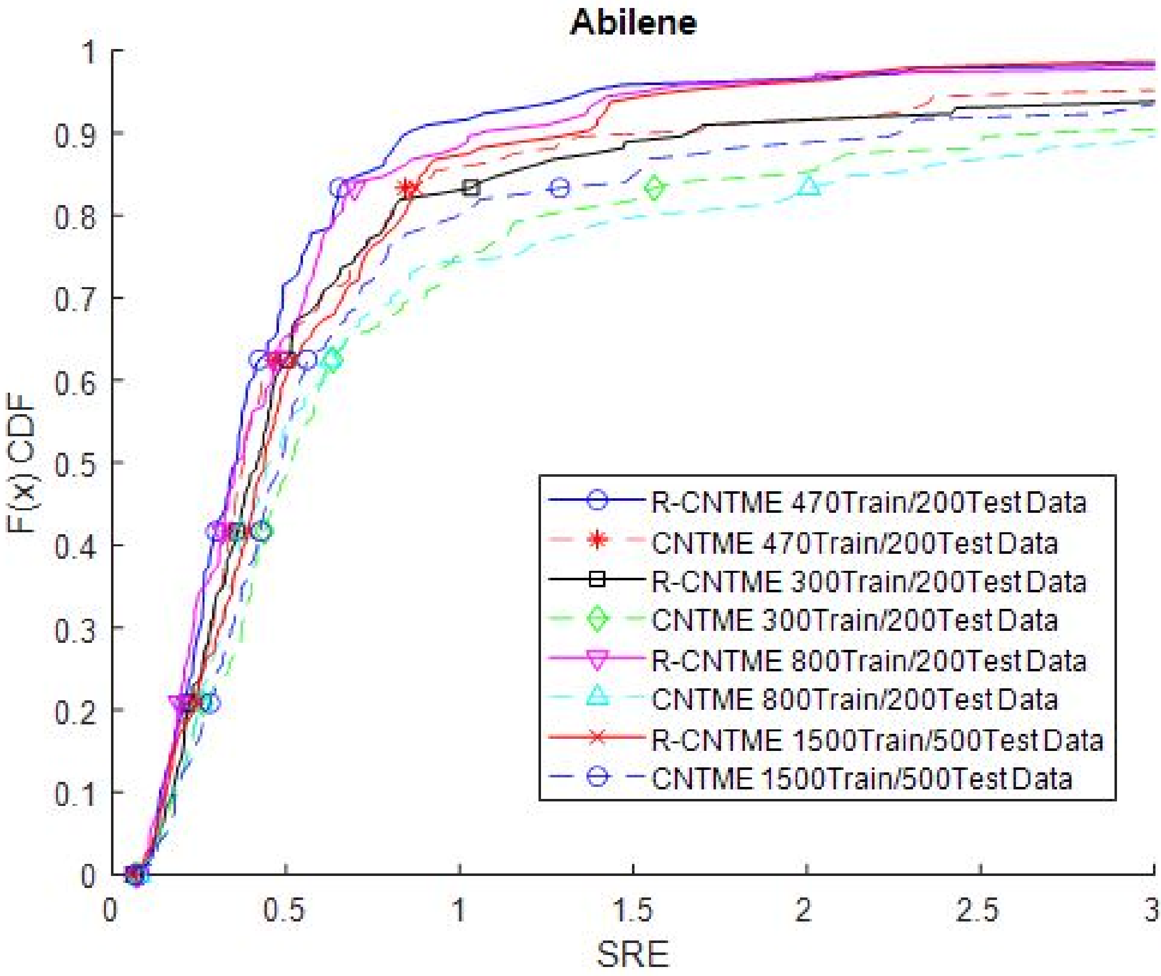

Figure 5.11: CDF of Spatial Relative Error for Dataset

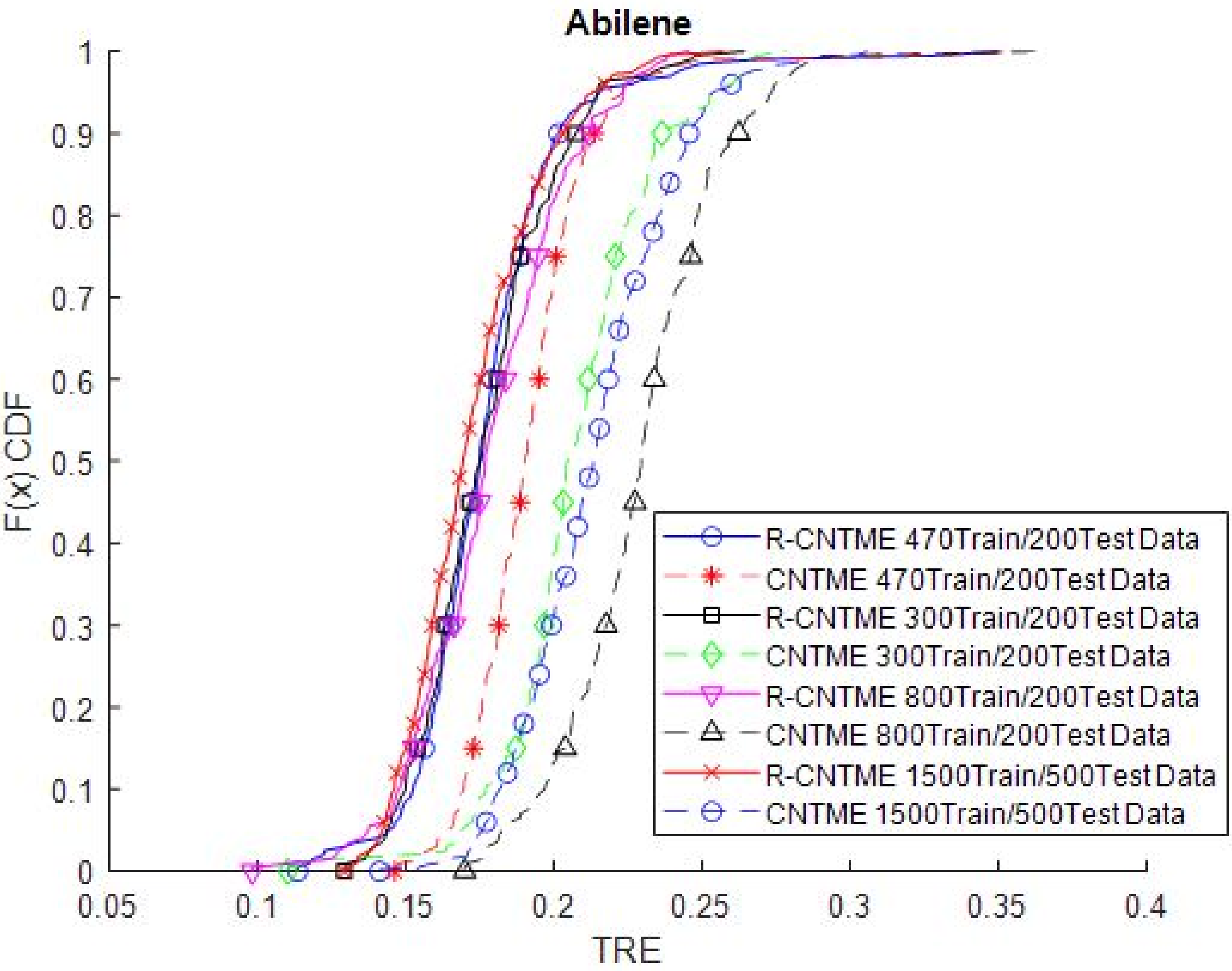

Figure 5.12: CDF of Temporal Relative Error (TRE) for Dataset

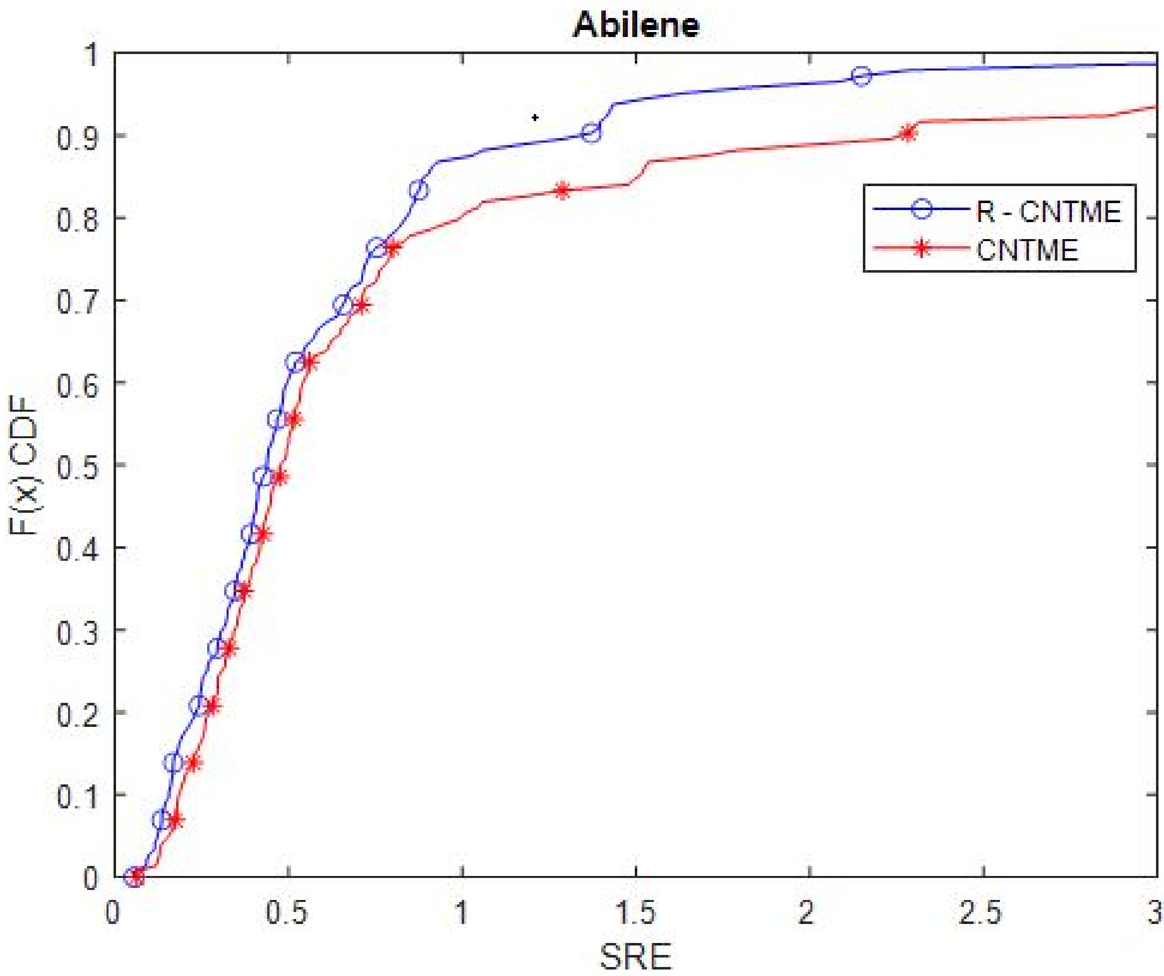

Figure 5.13: CDF of Spatial Relative Error (SRE) for Dataset

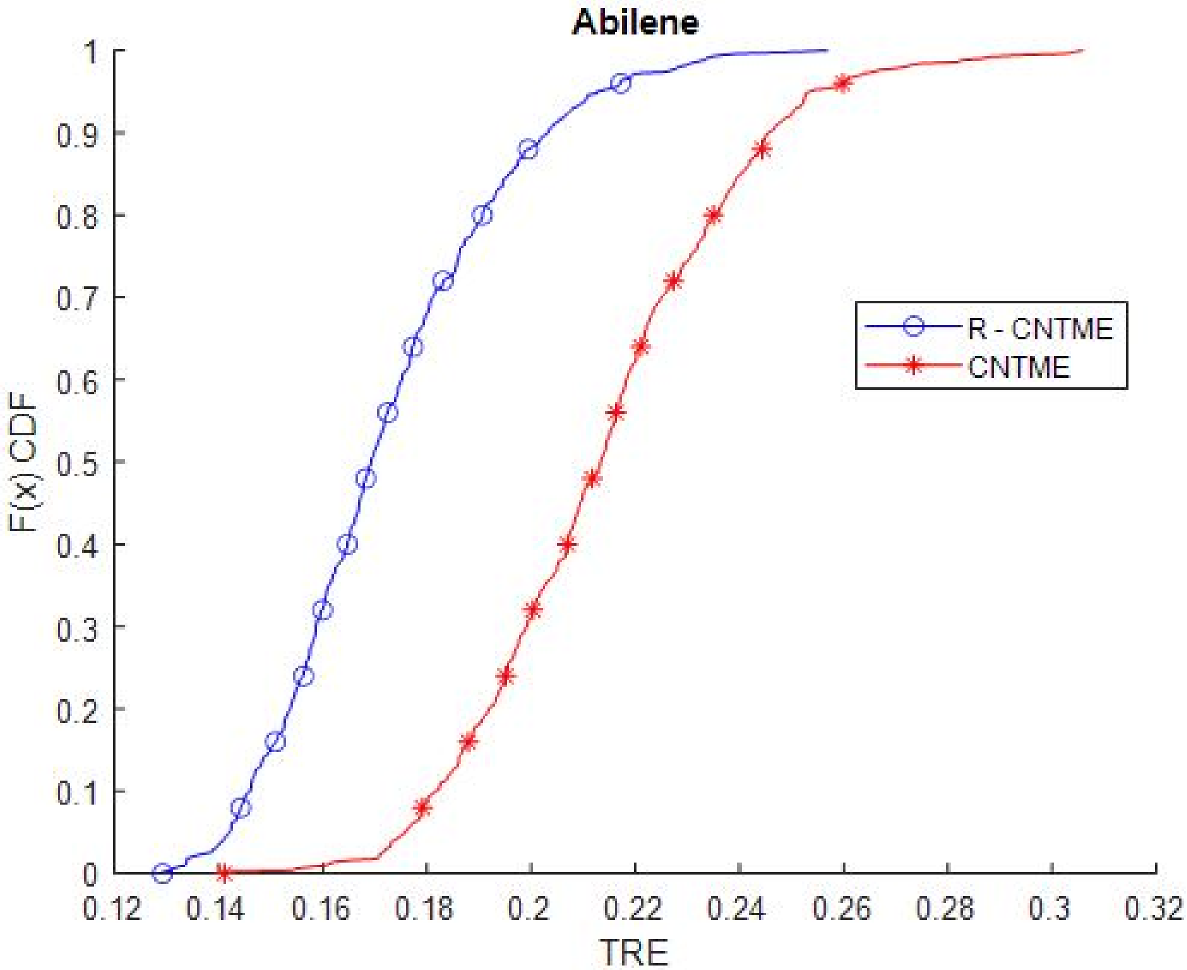

Figure 5.14: CDF of Temporal Relative Error (TRE) for Dataset

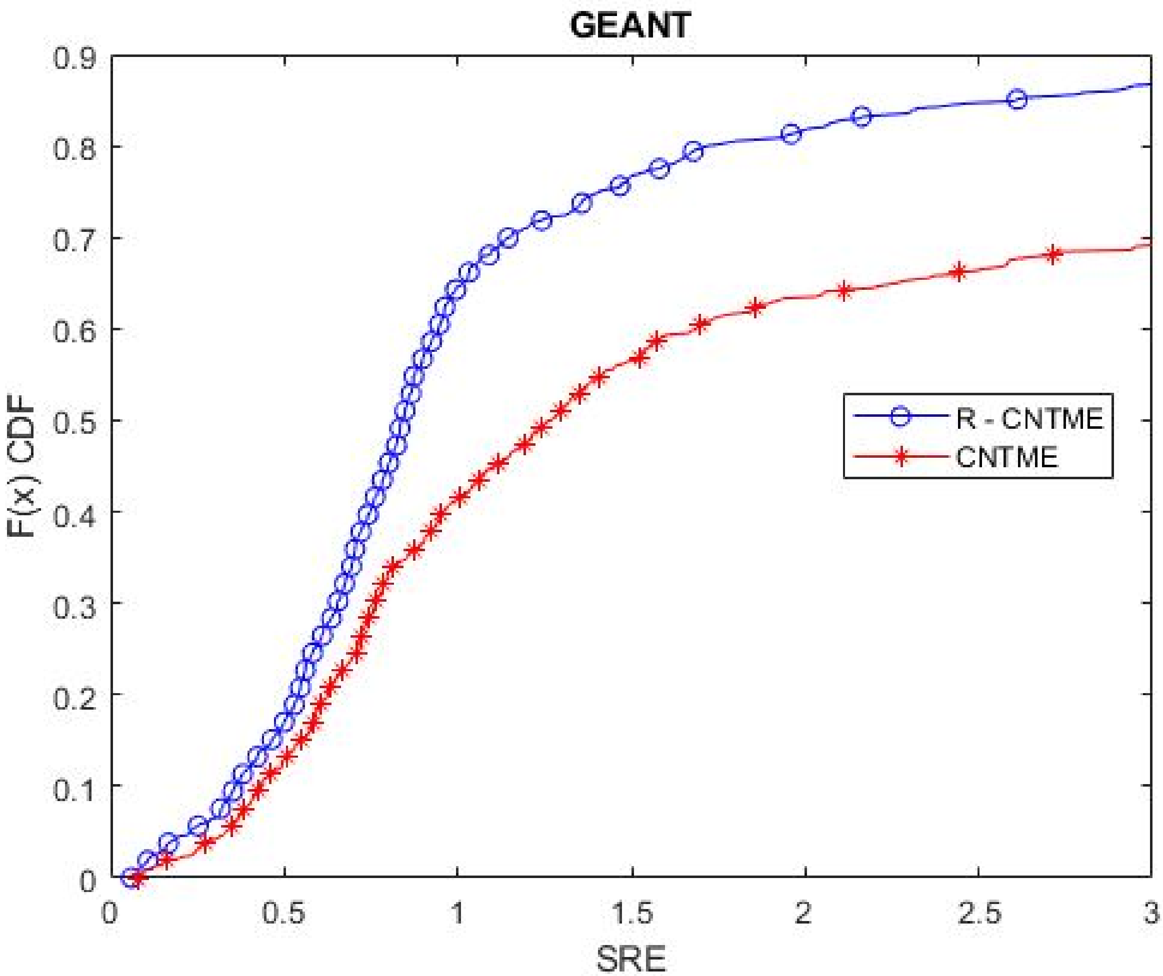

Figure 5.15: CDF for GÉANT Dataset

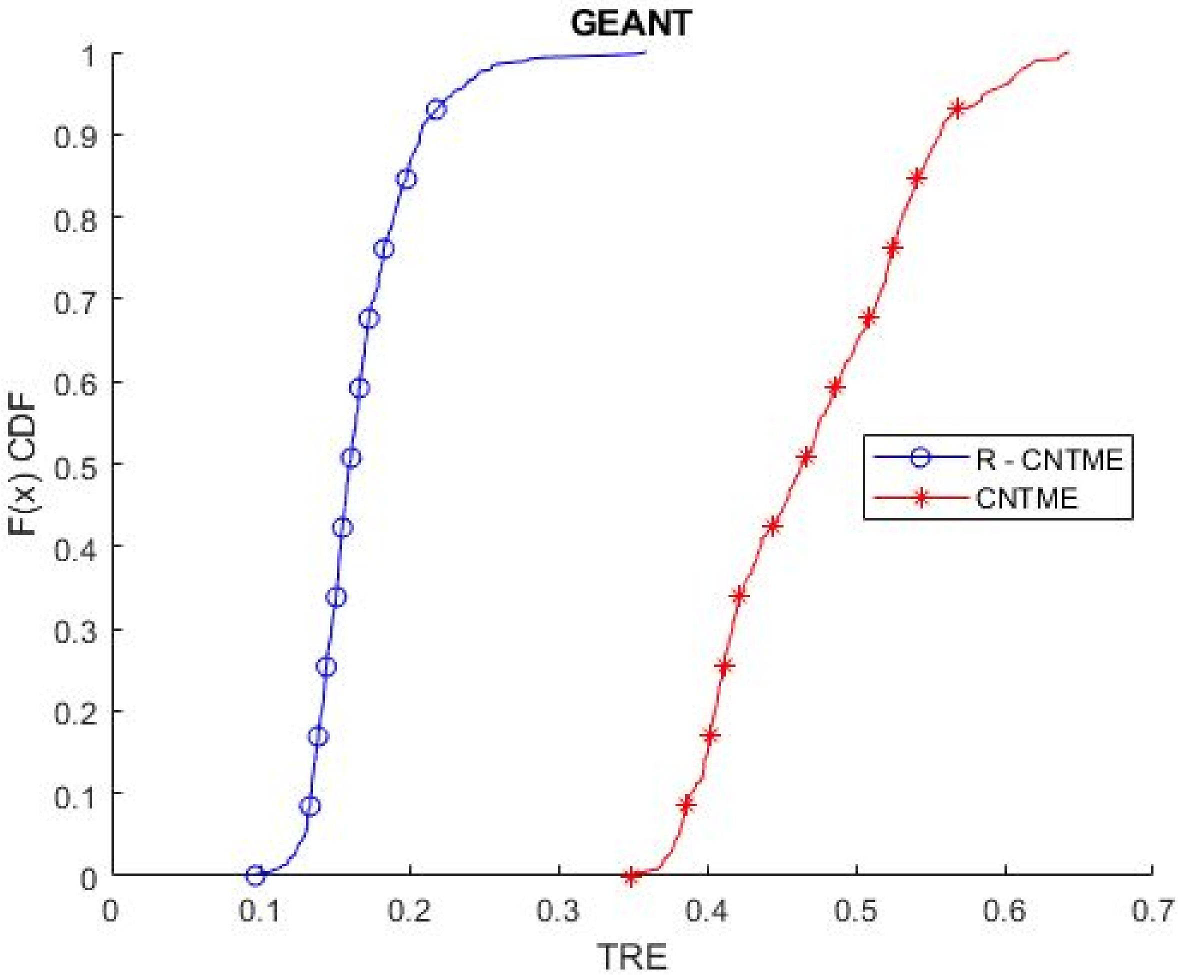

Figure 5.16: Temporal Relative Error (TRE) of for Dataset

then 0.2 for more then 90 % of cases, where as CNTME TRE value falls below 0.5 when compared with R-CNTME's performance.

The CDFs behavior is also depicted as though it were fitted to a standard pdf (Figure 5.17 and 5.18) for Abilene datasets. These graphs demonstrate that R-CNTME outperforms CNTME. Furthermore, performance is better in terms of bias and standard deviation for both datasets (Abilene and GEANT) for our proposed scheme R-CNTME than CNTME. Figure 5.19 and 5.20 shows the performance evaluation of R-CNTME in terms of bias and standard deviation for Abilene dataset. Similarly Figure 5.21 and 5.22 presents the bias and standard deviation for GEANT dataset.

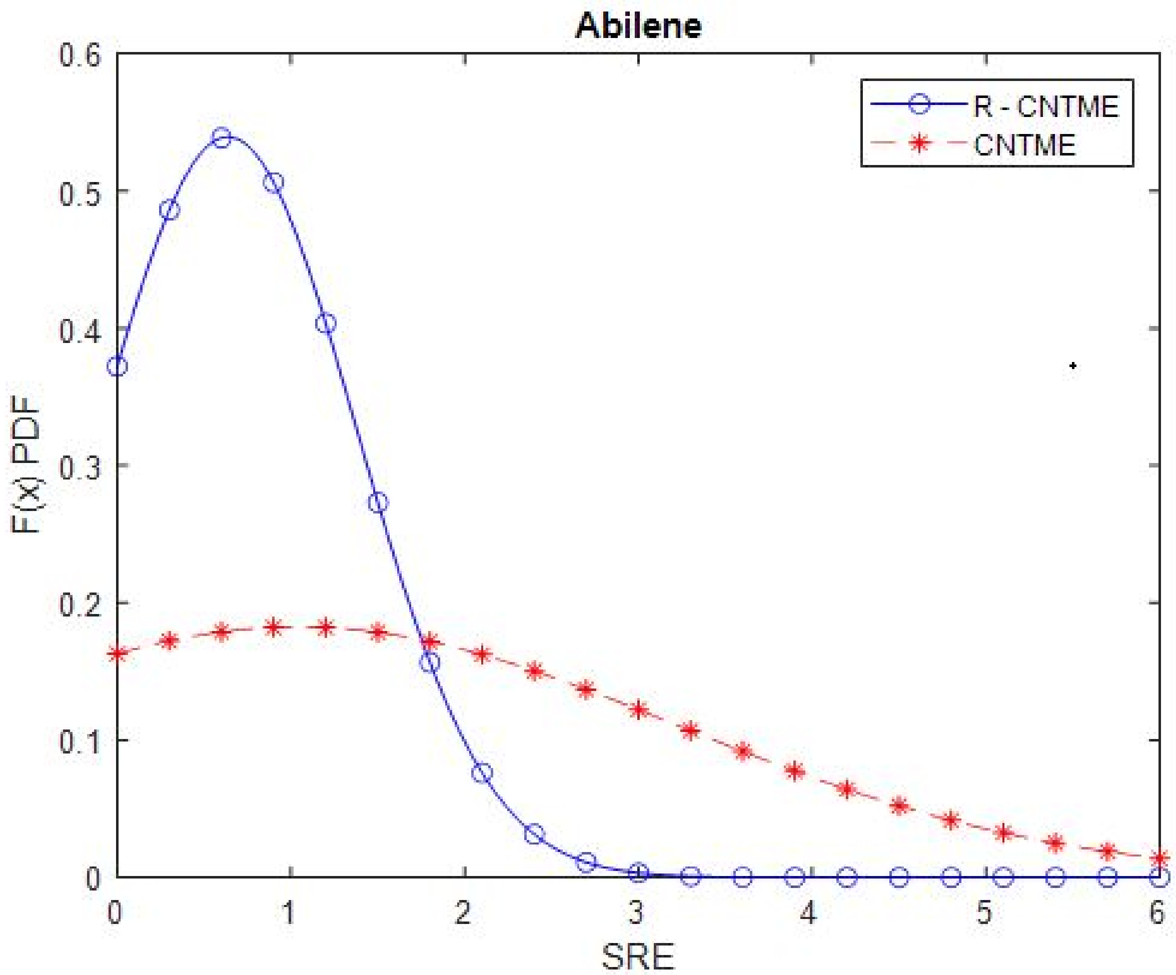

Figure 5.17: PDF of Spatial Relative Error for dataset

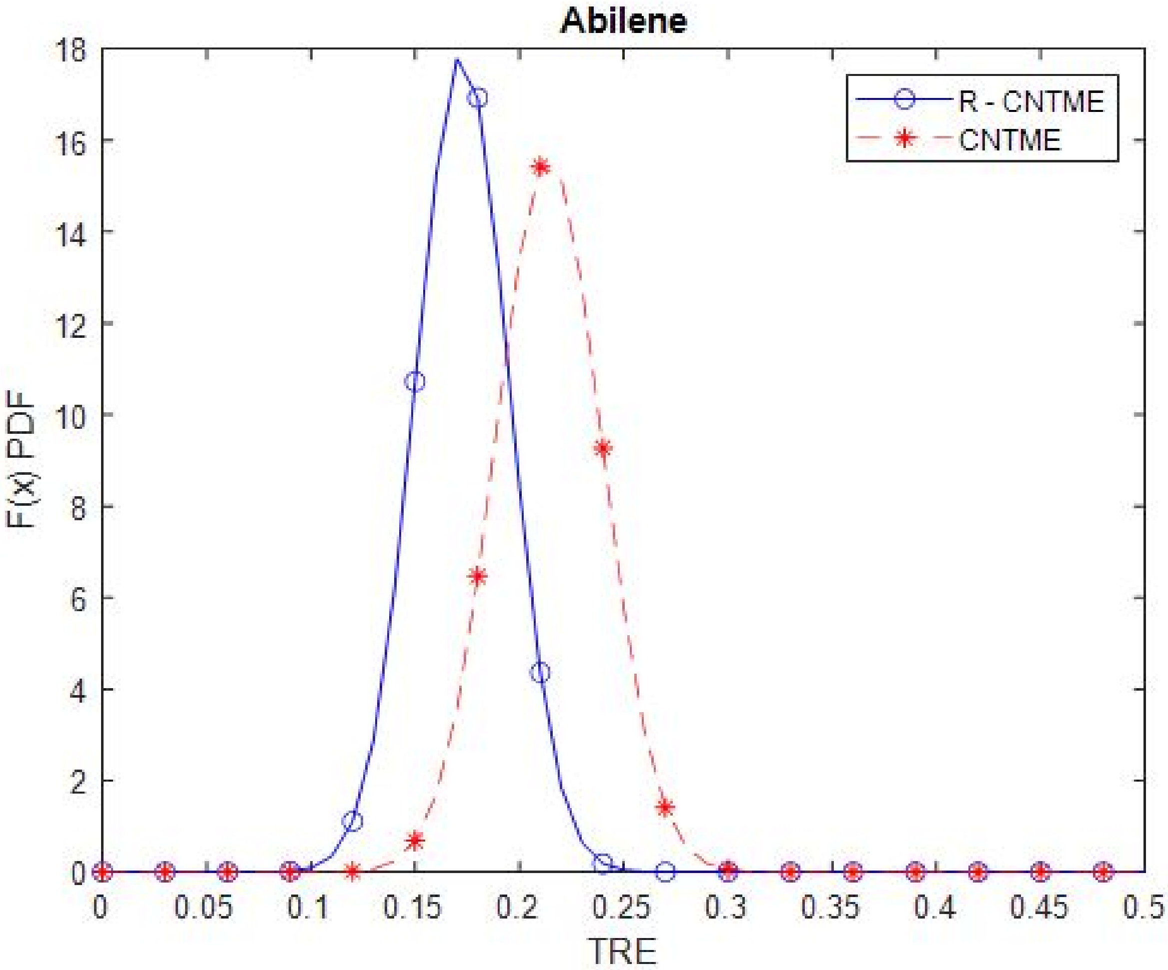

Figure 5.18: PDF of Temporal Relative Error for R-CNTME and CNTME

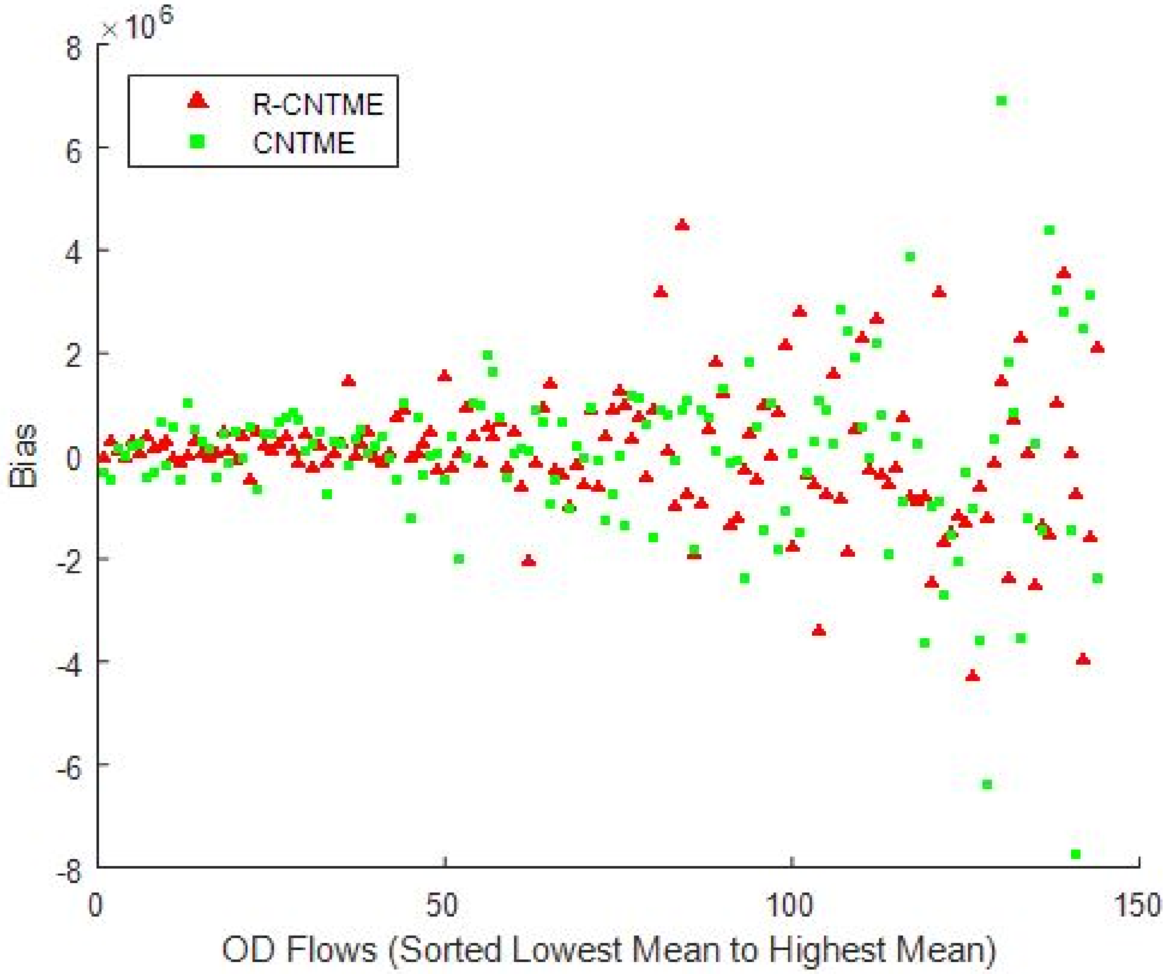

Figure 5.19: Bias and OD Flows performance evaluation for both algorithms

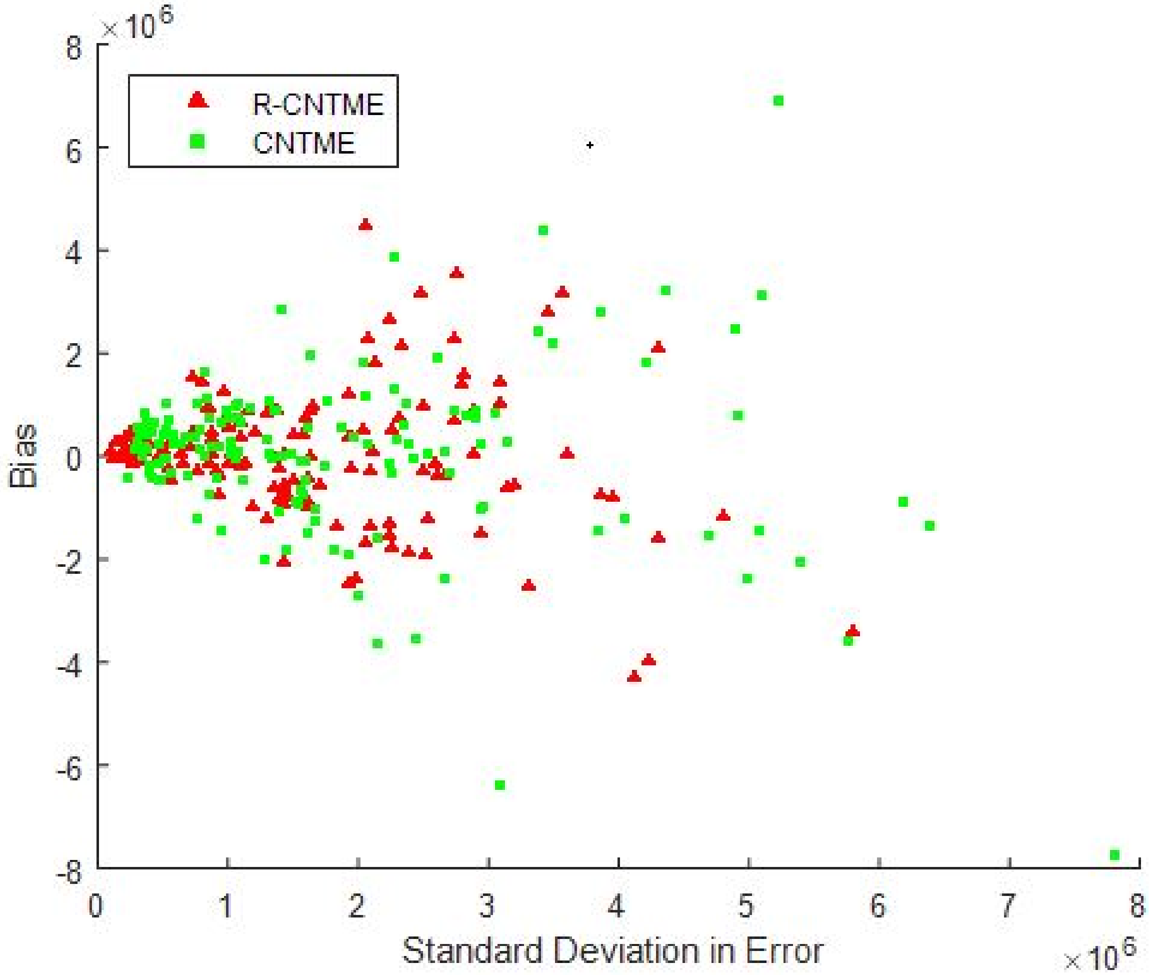

Figure 5.20: Performance in terms of Bias and standard deviation

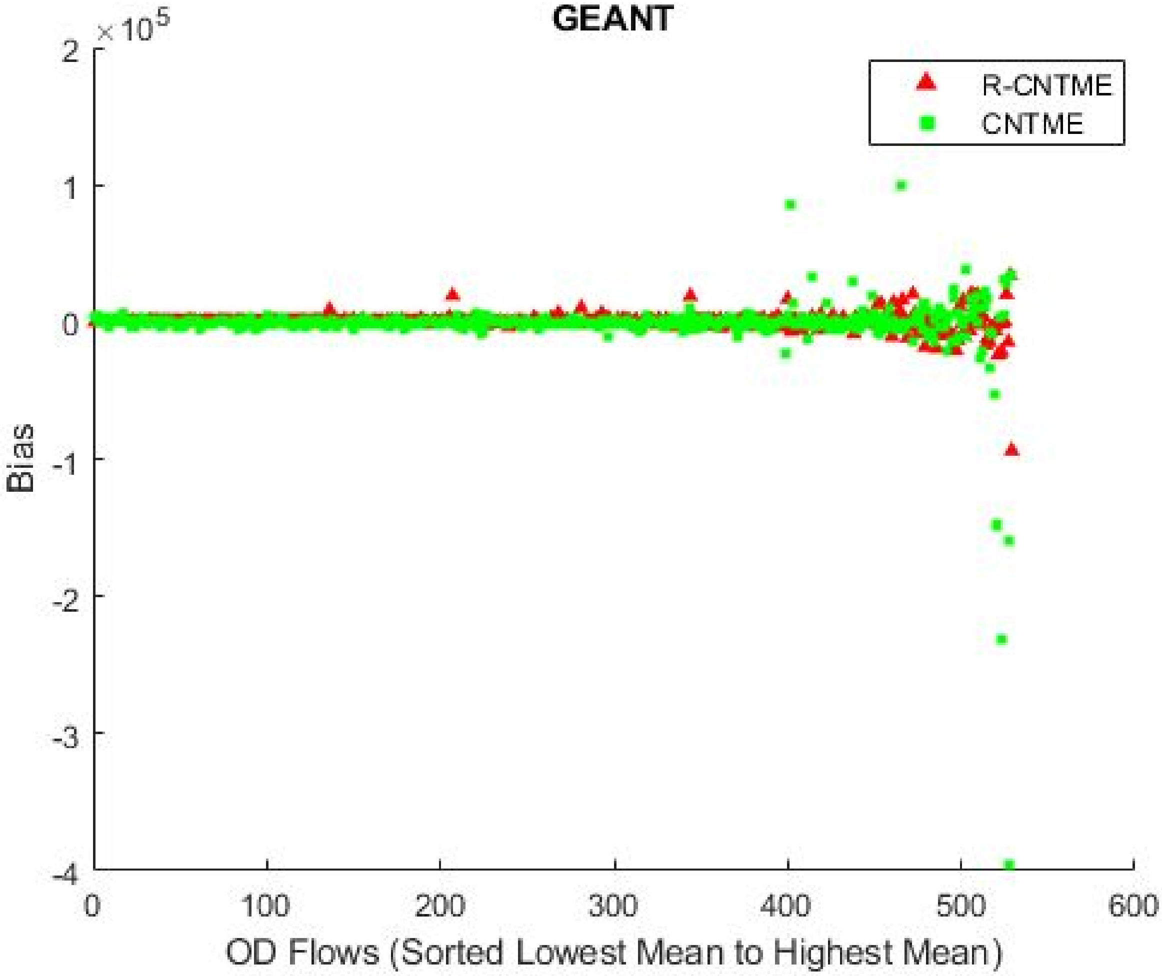

Figure 5.21: Bias: Performance Evaluation of dataset

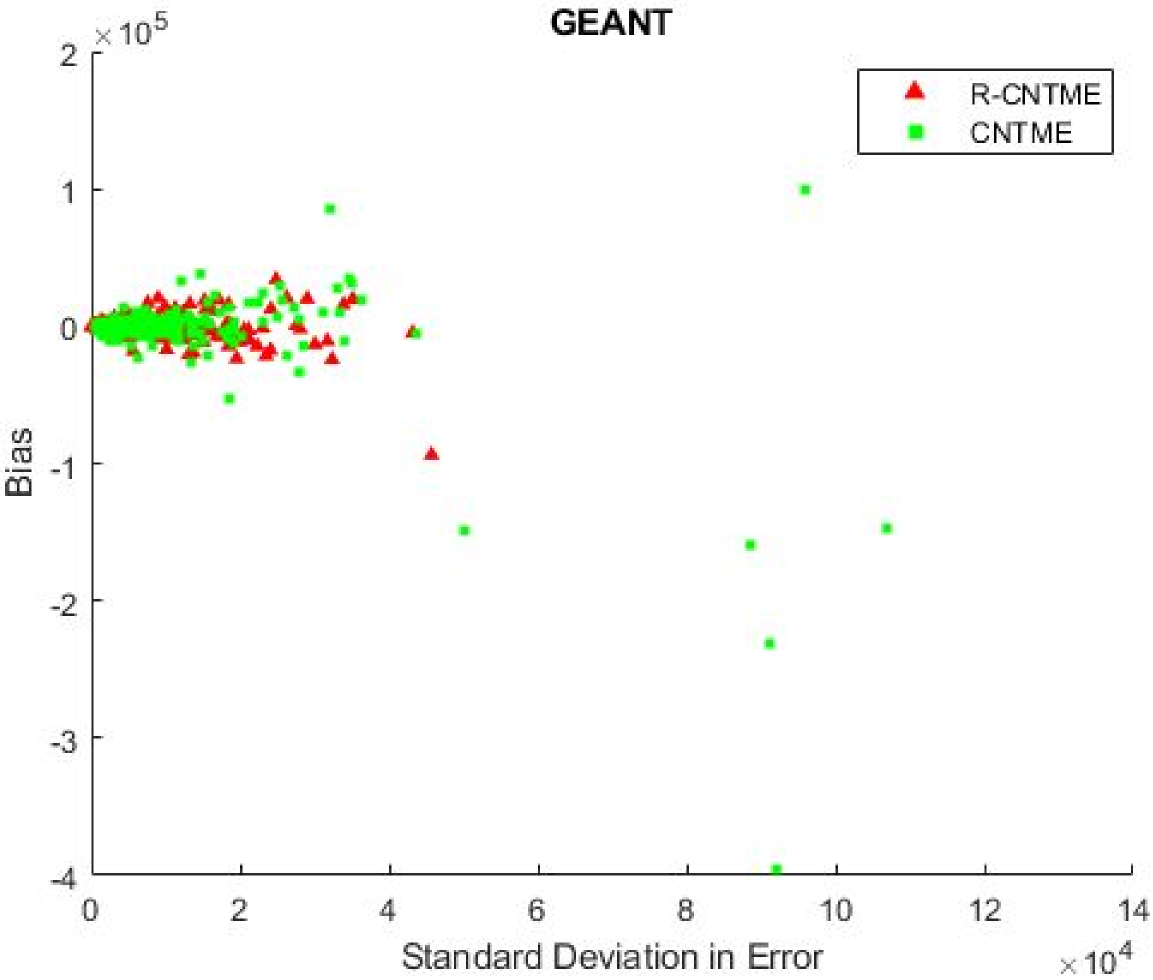

Figure 5.22: Standard Deviation: Performance Evaluation of dataset

5.7.4 Tracking Performance for OD flows

Performance of the OD flows prediction is displayed in Figure 5.23-5.28 for six random OD flows when compared to the ground truth for the Abilene datasets. It is obvious that R-CNTME outperforms CNTME when OD flow prediction comparison to CNTME. And it is obvious it accurately estimates OD flows, as well as tracks unusual changes in accordance with the ground reality. Similarly performance of OD prediction is shows in Figure 5.29-5.34 for six random OD flows when compared to the ground truth for GEANT network datasets. For this dataset R-CNTME again outperforms CNTME.

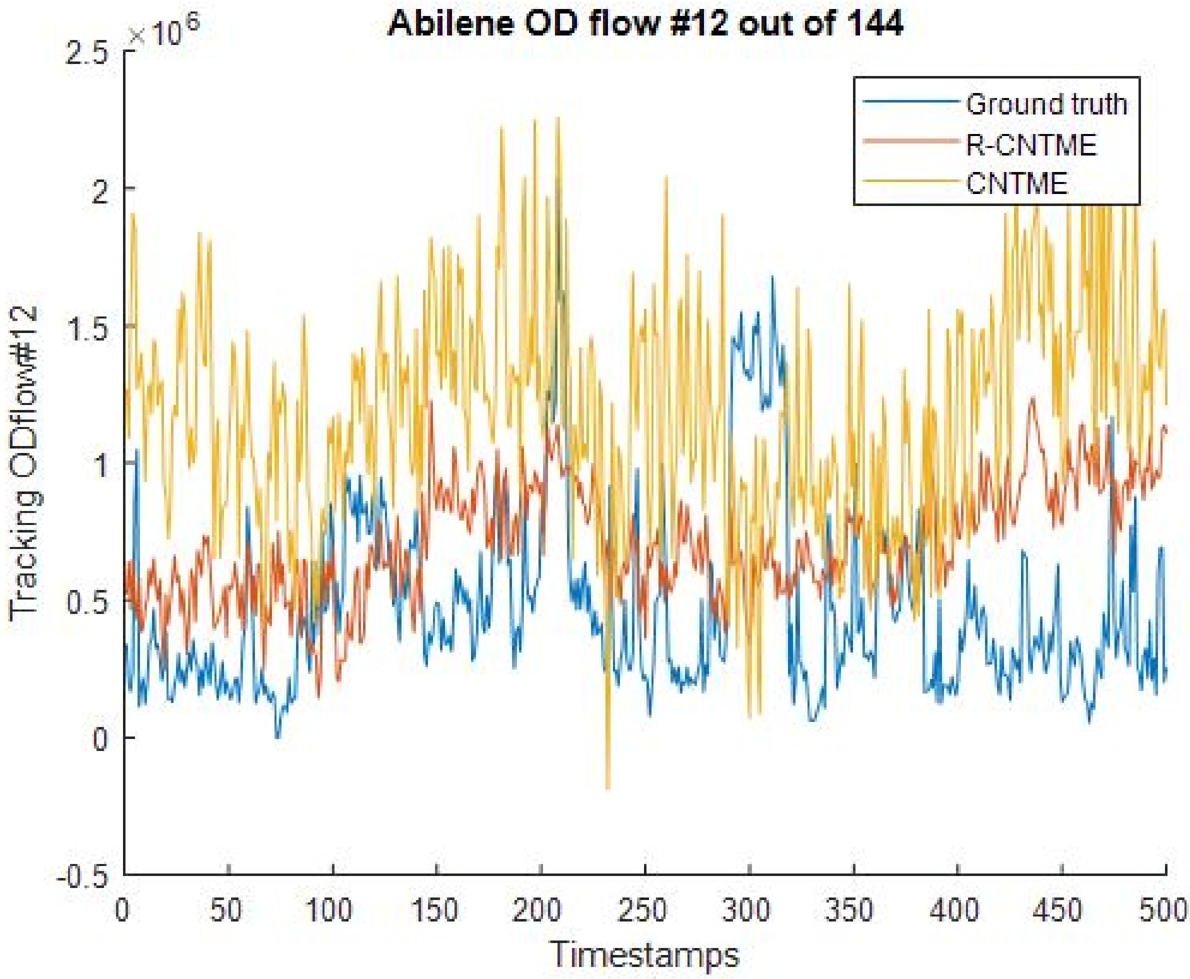

Figure 5.23: OD Tracking Performance for Abilene Network

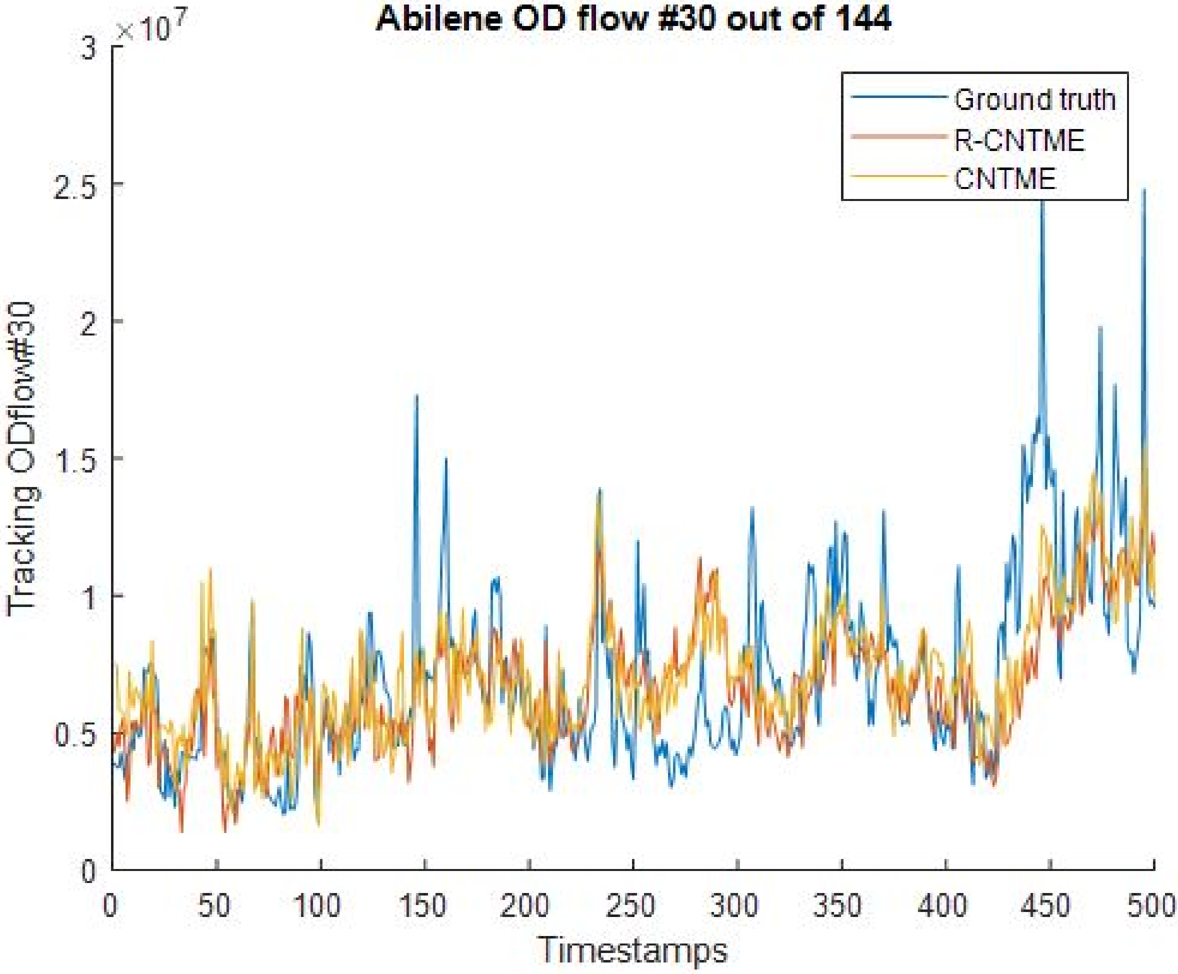

Figure 5.24: OD Tracking Performance for Abilene Network

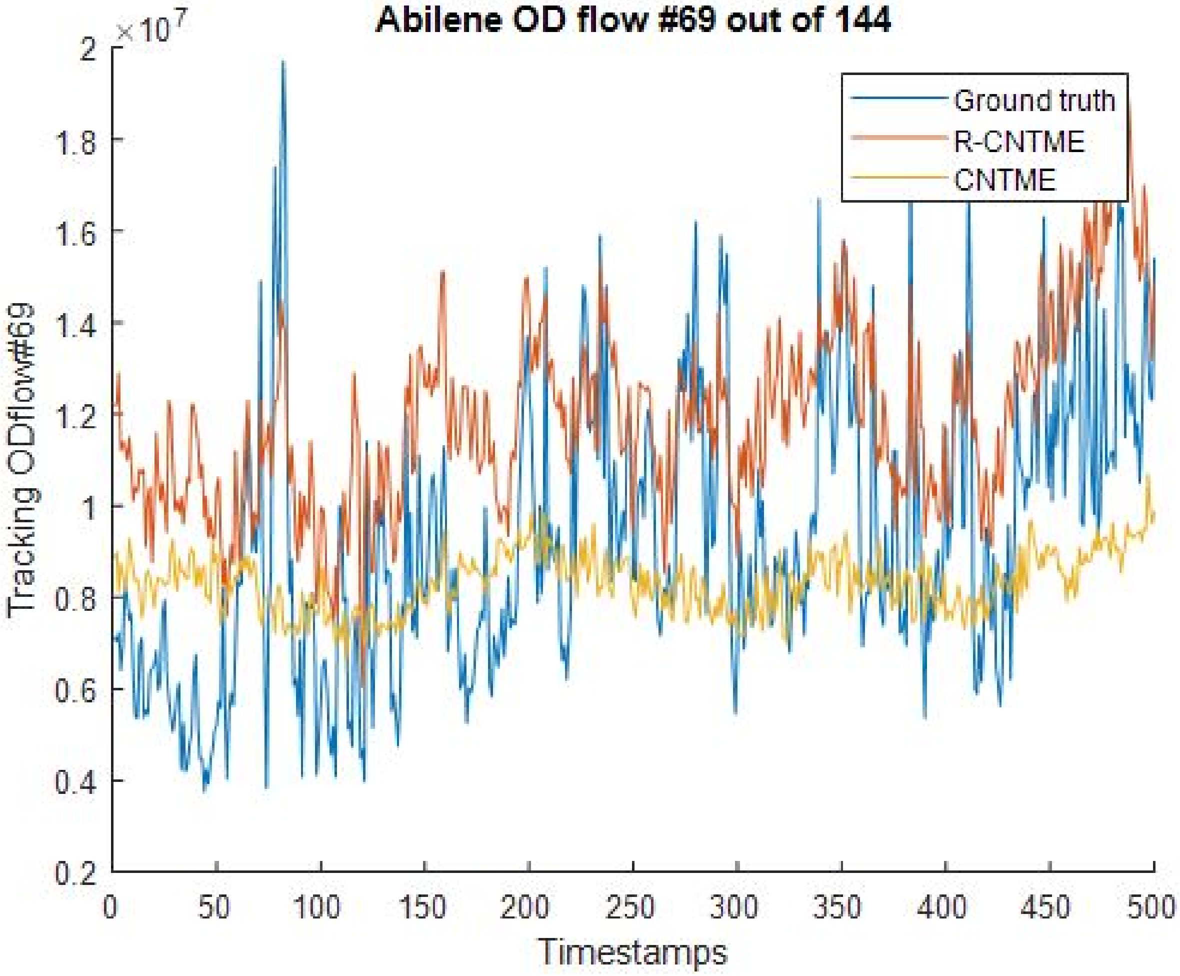

Figure 5.25: OD Tracking Performance for Abilene Network

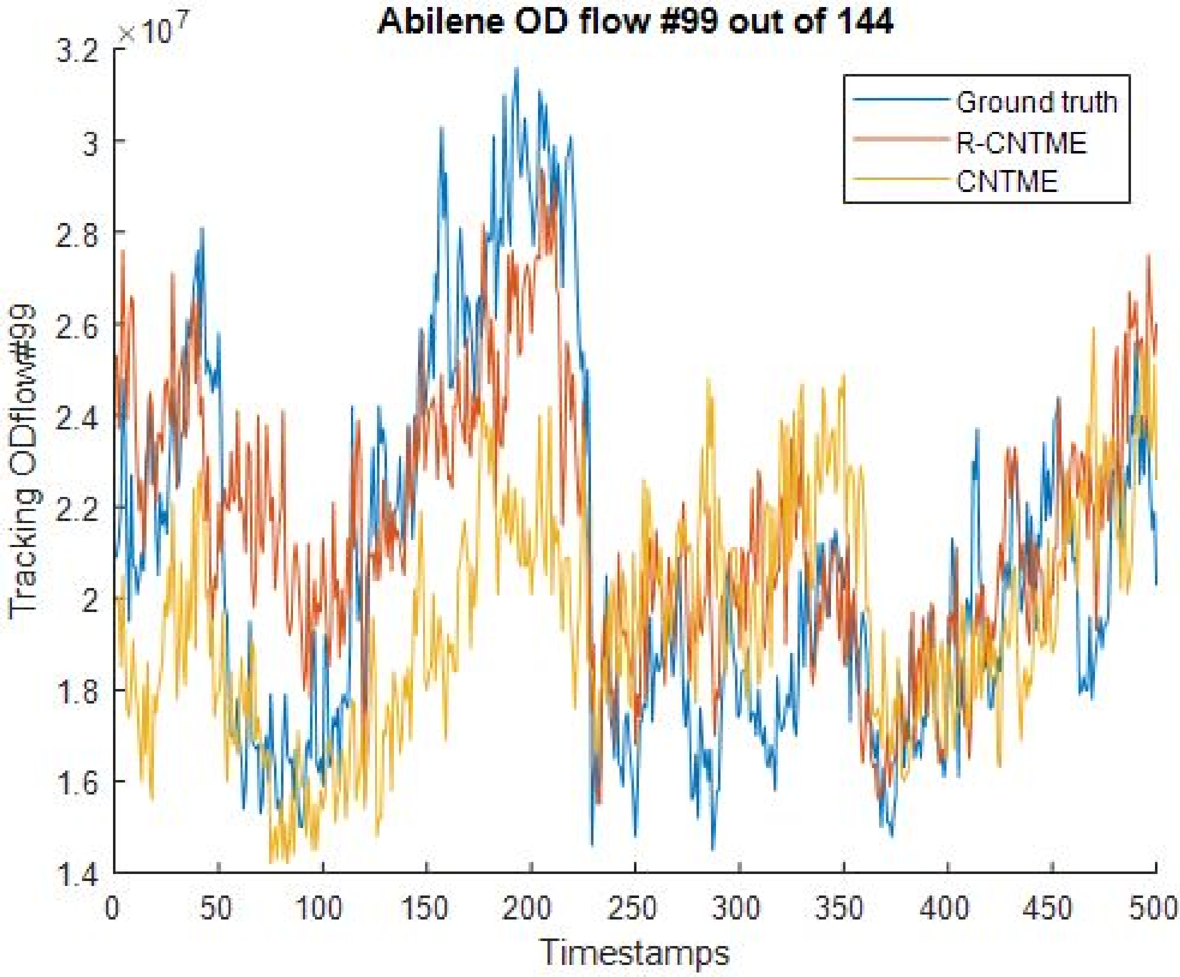

Figure 5.26: OD Tracking Performance for Abilene Network

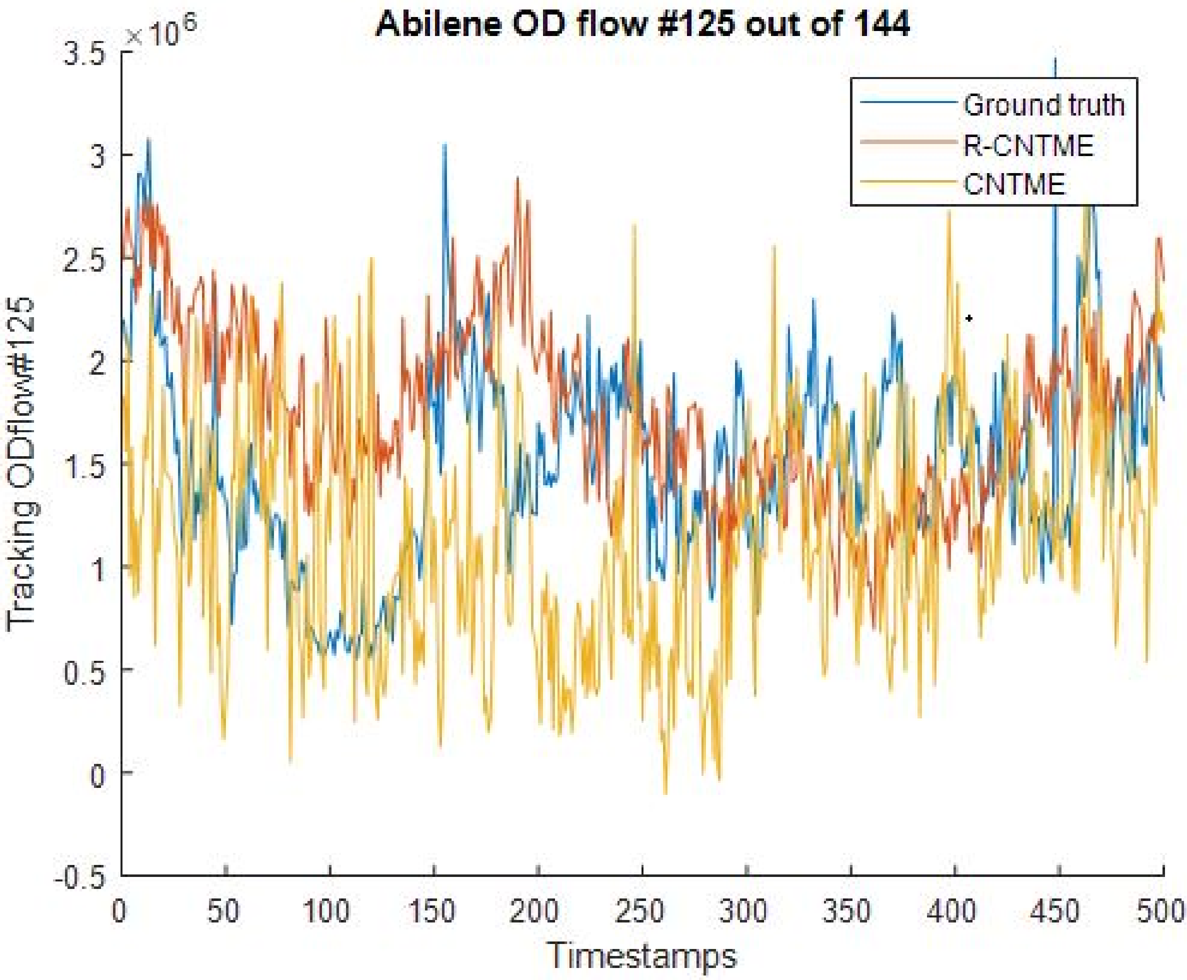

Figure 5.27: OD Tracking Performance for Abilene Network

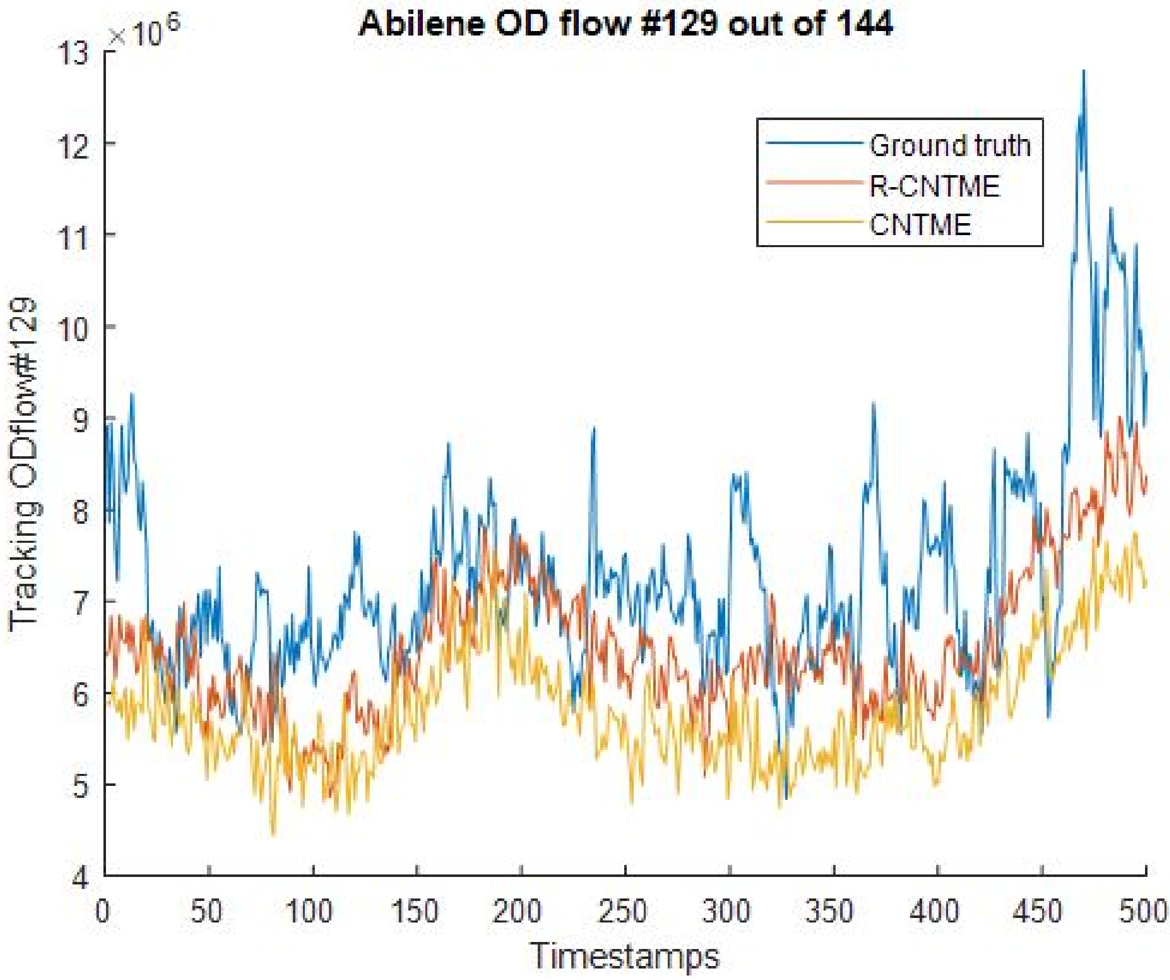

Figure 5.28: OD Tracking Performance for Abilene Network

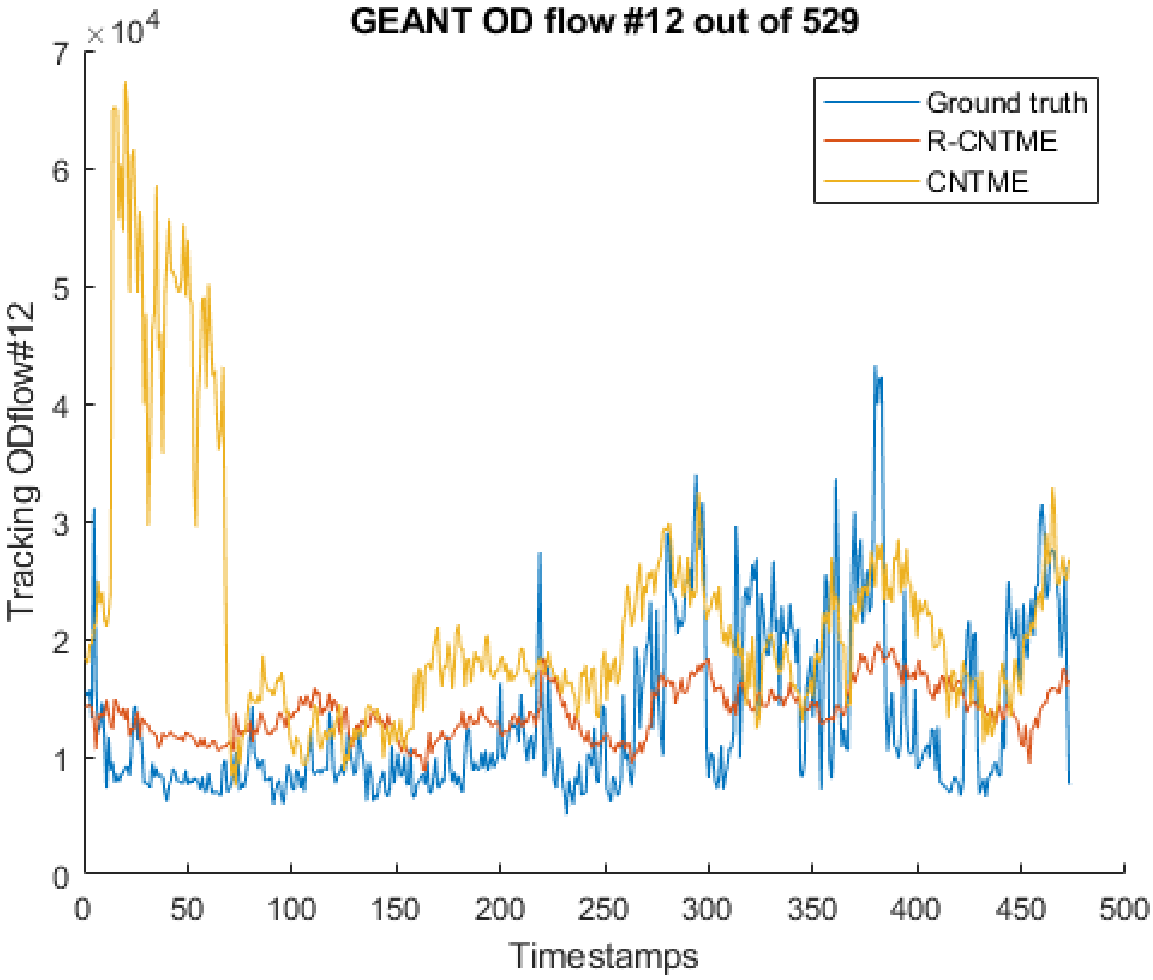

Figure 5.29: OD Tracking Performance for GEANT Network

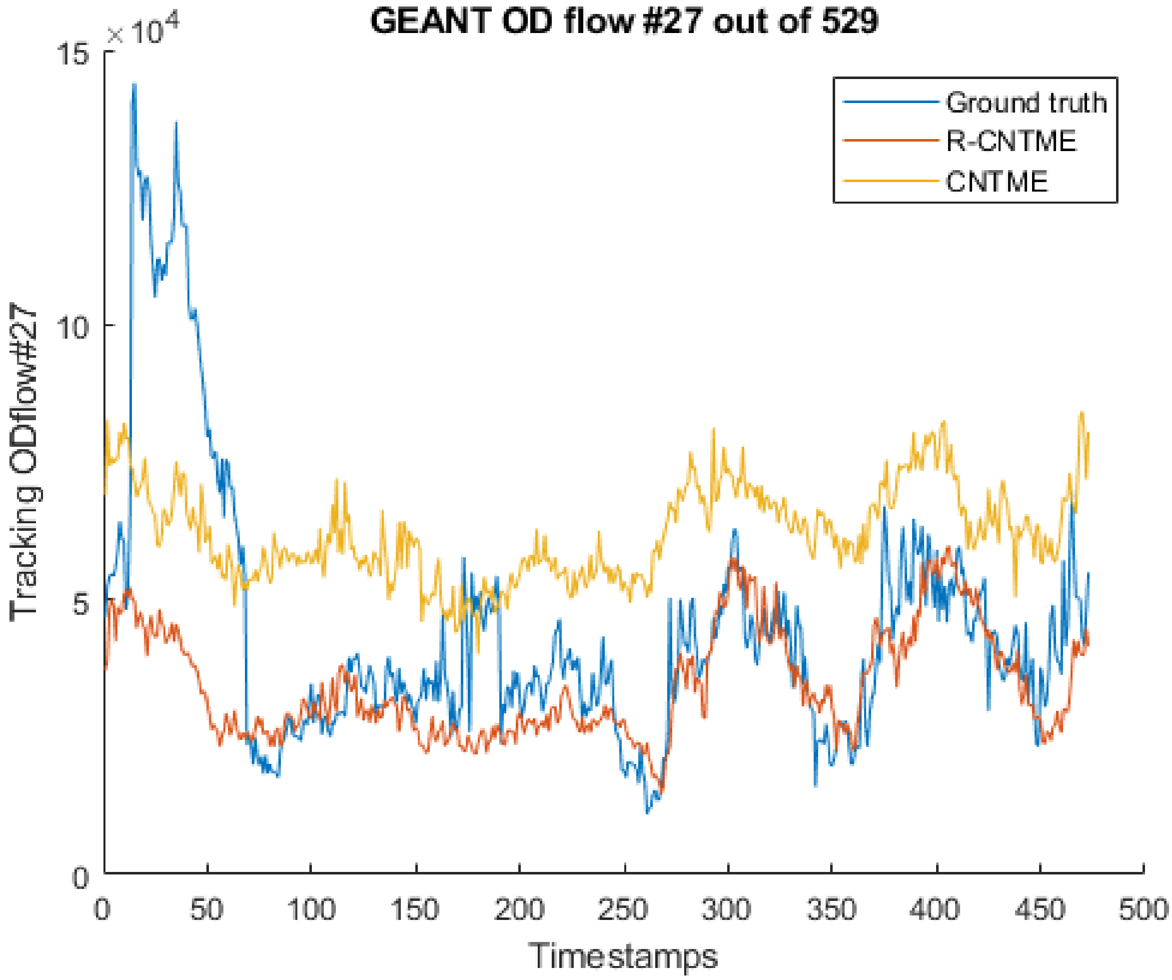

Figure 5.30: OD Tracking Performance for GEANT Network

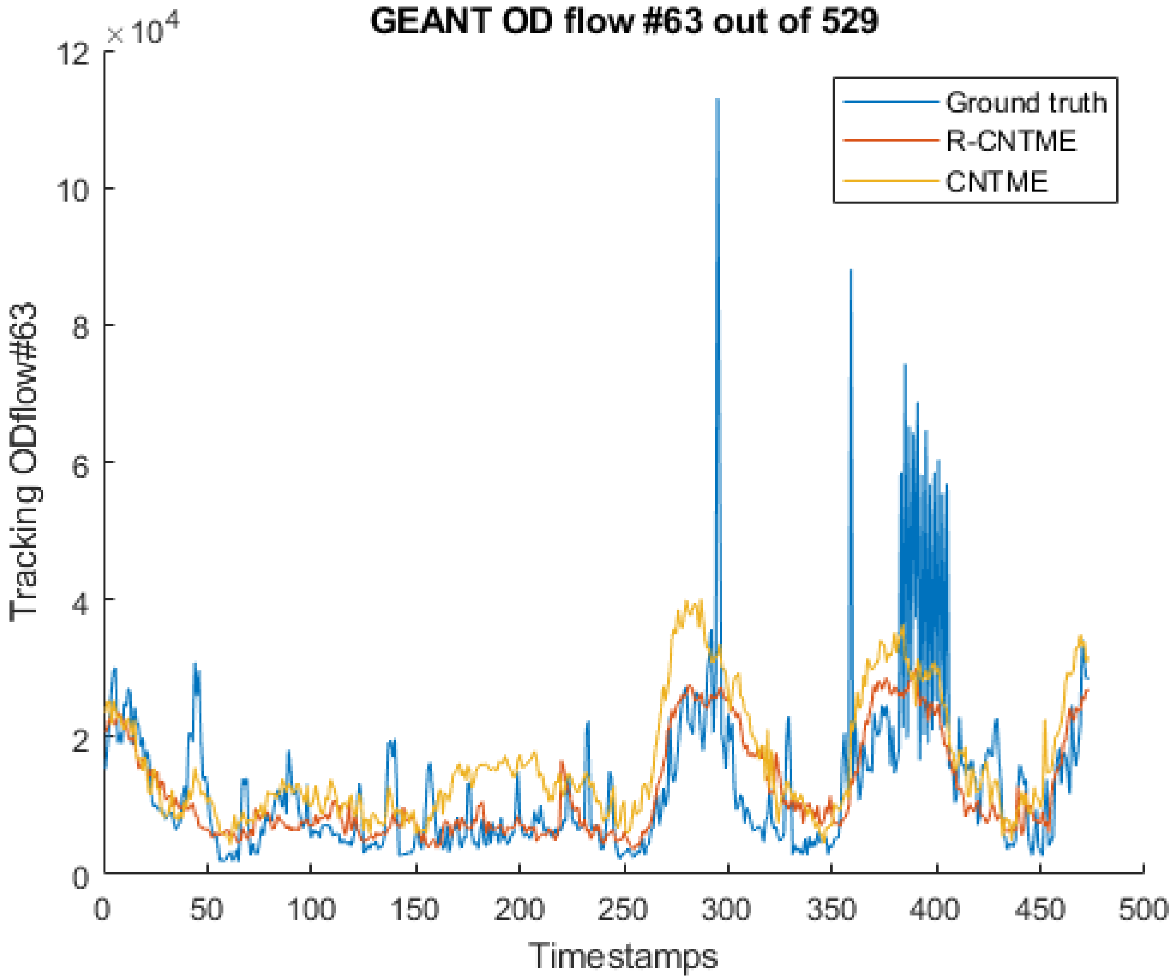

Figure 5.31: OD Tracking Performance for GEANT Network

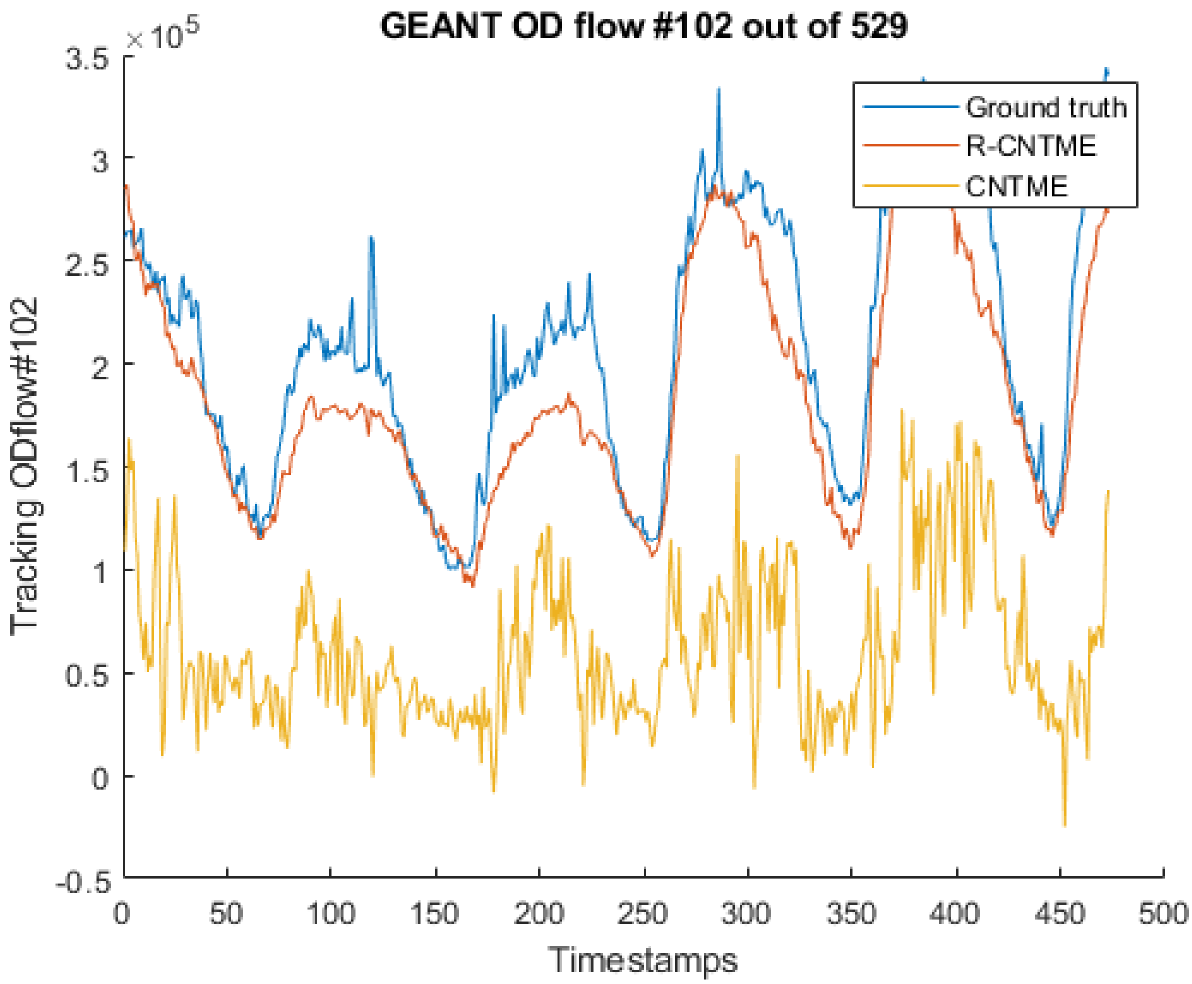

Figure 5.32: OD Tracking Performance for GEANT Network

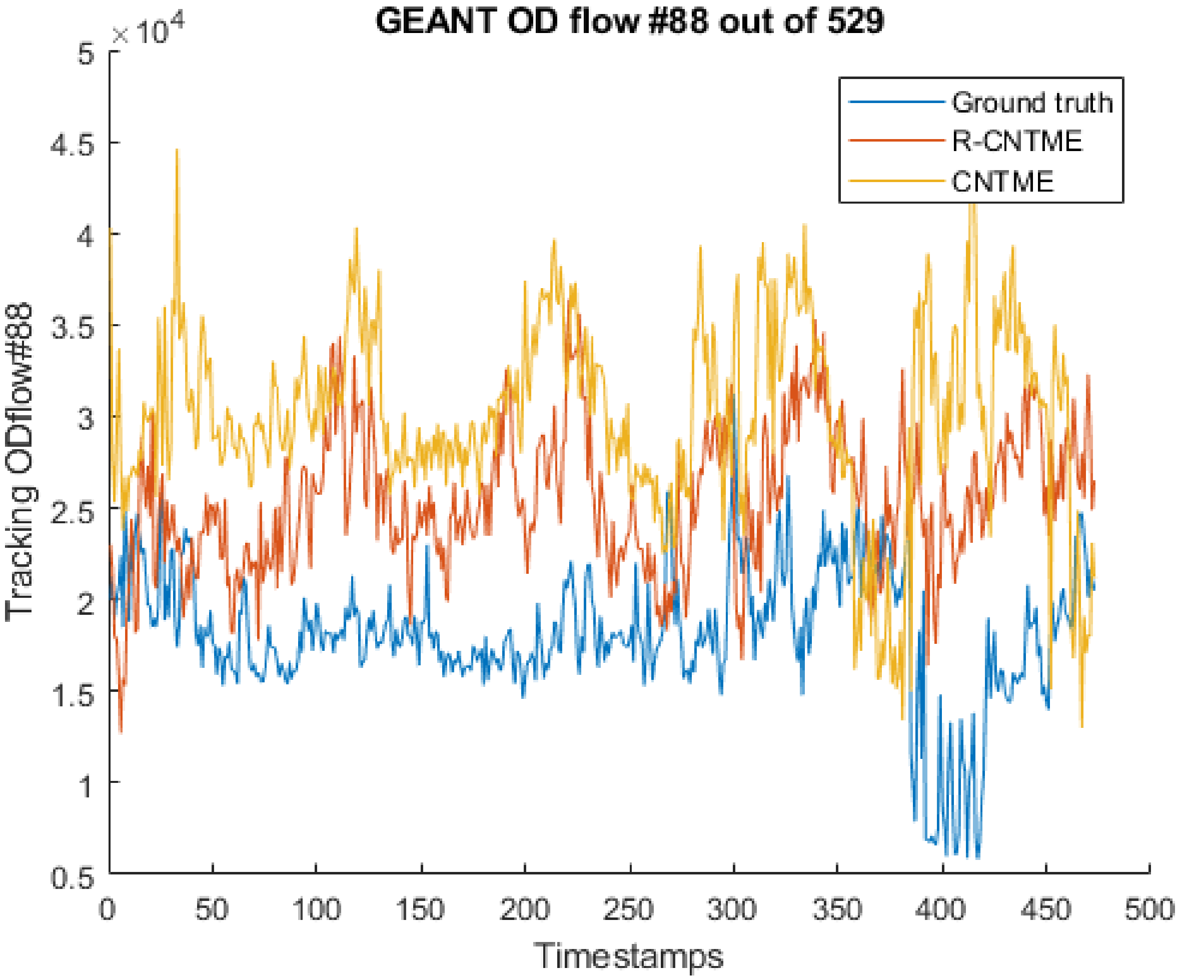

Figure 5.33: OD Tracking Performance for GEANT Network

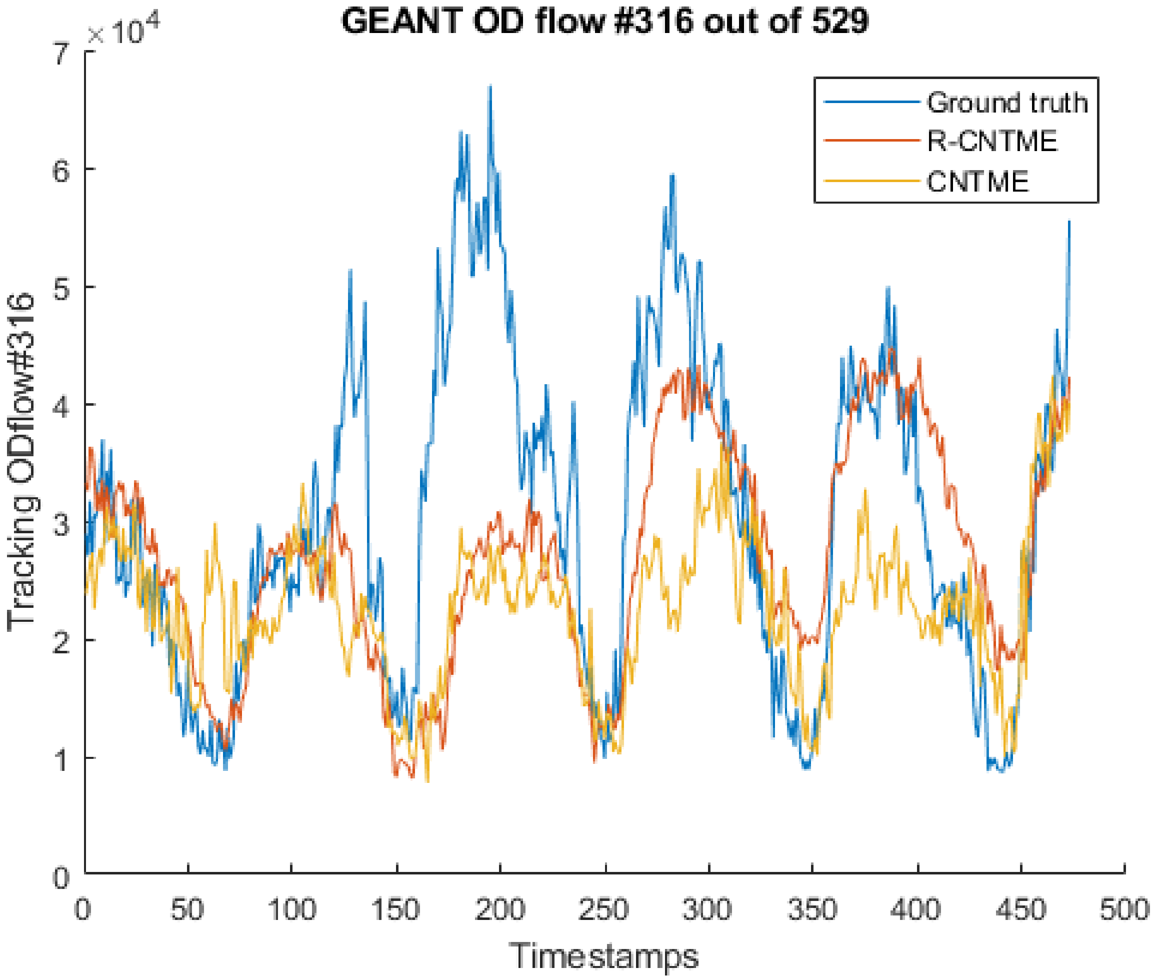

Figure 5.34: OD Tracking Performance for GEANT Network

5.8 Conclusion

In backhaul cloud networks, accurate traffic estimation is critical for early identification and prevention of anomalies. In this thesis, an architecture for Traffic Matrix Estimation was presented using Convolutional Neural Network for Cloud Networks Traffic Estimate, that demonstrates superior performance even with sparse data and random noise errors in 2-D training datasets, and limited training dataset availability than previous work for Network Traffic Matrix estimation. The suggested architecture offers stable performance with training data artefacts, as well as enhanced Error and Anomaly Detection performance, according to extensive simulations using real-world datasets.

Chapter 6

Conclusion and Future Work

Several network applications involve different types of traffic measurements and information. Network applications such as a network control with effective solutions for congestion, capacity planning for large networks, traffic management and engineering and network optimization required certain network measurements such as traffic volume in real time that includes both mean and variance of traffic volumes. Several applications require delay information. Therefore, network applications select an appropriate traffic matrix estimation technique to utilize the traffic information maintain the accuracy and efficiency of the estimation of traffic matrix.

In this thesis we investigated the traffic matrix that may experience over-dispersion and formulation of a two-step optimization approach with appropriate accuracy and additional constraint. We developed a novel architecture that demonstrates superior results for the estimation and prediction of traffic matrix applying and neural network technique.

Initially estimation techniques are classified in two categories 1) the deterministic and 2) the statistical. For the deterministic approach the characteristics of the prior traffic matrix are required to recognize a accurate traffic matrix due to under-constrained or ill posed feature traffic matrix estimation. However the statistical approach required more information about traffic flows between Origin-Destination (OD) nodes for the true prediction and estimation of traffic matrix.

For the first contribution we referred the over dispersion problem where statistical methods usually fail when accuracy is considered. Statistical prediction approaches do not provide a reasonably accurate solution when faced with the problem of excessive dispersion. This work shows on actual dataset dispersion causes serious problem to small data flows. As a result, a two-stage optimization strategy is proposed with simulations in which large data flows are predicted with reasonable accuracy in the first step with more conservative estimates for small flows which are dispersed. A second optimization step with an extra restriction refines the solution for dispersed small flows. Experimental results verified that for ill-estimated flows, estimation and prediction can be increased to 4 orders of values.

For second contribution, a robust traffic matrix estimation framework is developed with guaranteed superior performance with availability of limited training data or outlier measurements. Moreover we investigated the limiting training data challenges and develop a new algorithms with architecture that can give solution for these difficulties and guaranteed better performance. This approach provides superior outcomes for estimation of traffic matrix using convolutional neural network based technique with limited training data availability and outlier end-to-end measurements.

For future work , a robust technique with artificial neural network algorithms is considered for a large back-bone network which can perform with more evaluation factors involved. Large network for a institutional campus will be considered for performance evaluation. Furthermore , more constraints can be applied for compressed sensing techniques for traffic matrix estimation.

References

[1] Abilene-Corbato, “Abilene,” 2019. Last accessed 20 December 2020.

[2] T. A. N. Topology, “Geant,” 2017. Last accessed 16 September 2020.

[3] M. Emami, R. Akbari, R. Javidan, and A. Zamani, “A new approach for traffic matrix estimation in high load computer networks based on graph embedding and convolutional neural network,” *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 6, p. e3604, 2019.

[4] M. Polverini, A. Iacovazzi, A. Cianfrani, A. Baiocchi, and M. Listanti, “Traffic matrix estimation enhanced by sdns nodes in real network topology,” in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 300–305, IEEE, 2015.

[5] A. Svigelj, R. Sernec, and K. Alic, “Network traffic modeling for load prediction: a user-centric approach,” *IEEE network*, vol. 29, no. 4, pp. 88–96, 2015.

[6] Z. Hu, Y. Qiao, and J. Luo, “Atme: Accurate traffic matrix estimation in both public and private datacenter networks,” *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 60–73, 2015.

[7] K. Xu, M. Shen, Y. Cui, M. Ye, and Y. Zhong, “A model approach to the estimation of peer-to-peer traffic matrices,” *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 5, pp. 1101–1111, 2013.

[8] Y. Niu and H. Tian, “Study on a new model for network traffic matrix estimation,” in *2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 152–154, IEEE, 2014.

[9] D. Kumlu and İ. Hökelek, “Network traffic estimation using markov chain and incremental gaussian mixture,” in *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pp. 1187–1190, IEEE, 2015.

[10] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, “Traffic matrices: balancing measurements, inference and modeling,” in *Proceedings*

of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 362–373, 2005.

[11] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, no. 1, pp. 206–217, 2003.

[12] O. Goldschmidt, "Isp backbone traffic inference methods to support traffic engineering," in *Internet Statistics and Metrics Analysis (ISMA) Workshop*, pp. 1063–1075, 2000.

[13] S. Eum, J. Murphy, and R. Harris, "A fast accurate lp approach for traffic matrix estimation," in *International Teletraffic Congress ITC19*, 2005.

[14] S. Eum, R. J. Harris, and A. Kist, "Generalized kruithof approach for traffic matrix estimation," in *2006 14th IEEE International Conference on Networks*, vol. 2, pp. 1–6, IEEE, 2006.

[15] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 301–312, 2003.

[16] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American statistical association*, vol. 91, no. 433, pp. 365–377, 1996.

[17] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data," *Journal of the American Statistical Association*, vol. 93, no. 442, pp. 557–573, 1998.

[18] J. Cao, D. Davis, S. Vander Wiel, and B. Yu, "Time-varying network tomography: Router link data," *Journal of the American statistical association*, vol. 95, no. 452, pp. 1063–1075, 2000.

[19] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 161–174, 2002.

[20] D. Oran, "Osi isis intradomain routing protocol," *RFC 1142*, 1990.

[21] J. Moy *et al.*, "Ospf version 2," 1998.

[22] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[23] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[24] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[25] G. Dantzig, *Linear programming and extensions.* Princeton university press, 2016.

[26] I. Dikin, "Iterative solution of problems of linear and quadratic programming," in *Doklady Akademii Nauk*, vol. 174, pp. 747–748, Russian Academy of Sciences, 1967.

[27] P. G. Bergan and R. W. Clough, "Convergence criteria for iterative processes," *AIAA journal*, vol. 10, no. 8, pp. 1107–1108, 1972.

[28] S. Nordbotten, "Allocation in stratified sampling by means of linear programming," *Scandinavian Actuarial Journal*, vol. 1956, no. 1, pp. 1–6, 1956.

[29] R. Frisch, "La résolution des problèmes de programme linéaire par la méthode du potentiel logarithmique," *Cahiers du Seminaire D'Econometrie*, pp. 7–23, 1956.

[30] P. Huard, "Resolution of mathematical programming with nonlinear constraints by the method of centers," *Nonlinear programming*, pp. 207–219, 1967.

[31] P. Huard, "A method of centers by upper-bounding functions with applications," in *Nonlinear programming*, pp. 1–30, Elsevier, 1970.

[32] L. Khachain, "A polynomial time algorithm for linear programming," in *Doklady Akademii Nauk SSSR, ns*, vol. 244, pp. 1063–1096, 1979.

[33] G. Astfalk, I. Lustig, R. Marsten, and D. Shanno, "The interior-point method for linear programming," *IEEE software*, vol. 9, no. 4, pp. 61–68, 1992.

[34] K. Kim and J. L. Nazareth, "A primal null-space affine-scaling method," *ACM Transactions on Mathematical Software (TOMS)*, vol. 20, no. 3, pp. 373–392, 1994.

[35] C. Roos, T. Terlaky, and J.-P. Vial, *Theory and algorithms for linear optimization: an interior point approach.* Wiley Chichester, 1997.

[36] G. K. Zipf, "Some determinants of the circulation of information," *The American journal of psychology*, vol. 59, no. 3, pp. 401–421, 1946.

[37] J. Tinbergen, "Shaping the world economy; suggestions for an international economic policy," 1962.

[38] P. Pöyhönen, "A tentative model for the volume of trade between countries," *Weltwirtschaftliches Archiv*, pp. 93–100, 1963.

[39] D. E. Low, "New approach to transportation systems modeling," *Traffic quarterly*, vol. 26, no. 3, 1972.

[40] M. Gaudry and L. Lamarre, "Estimating origin-destination matrices from traffic counts: a simple linear intercity model for quebec," *Logistics and Transportation Review*, vol. 15, no. 5, 1979.

[41] J. Holm, T. Jensen, S. Nielsen, A. Christensen, B. Johnsen, and G. Ronby, "Calibrating traffic models on traffic census results only," *Traffic Engineering & Control*, vol. 17, no. 4, 1976.

[42] J. P. Kowalski and B. Warfield, "Modelling traffic demand between nodes in a telecommunications network," in *in ATNAC'95*, Citeseer, 1995.

[43] D. Lam, D. C. Cox, and J. Widom, "Teletraffic modeling for personal communications services," *IEEE communications magazine*, vol. 35, no. 2, pp. 79–87, 1997.

[44] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in measuring internet backbone traffic variability: Models metrics, measurements and meaning," in *Teletraffic Science and Engineering*, vol. 5, pp. 379–388, Elsevier, 2003.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[46] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annual review of statistics and its application*, vol. 6, pp. 355–378, 2019.

[47] L. Tan and X. Wang, "A novel method to estimate ip traffic matrix," *IEEE Communications Letters*, vol. 11, no. 11, pp. 907–909, 2007.

[48] I. J. Craig and J. C. Brown, "Inverse problems in astronomy. a guide to inversion strategies for remotely sensed data," *Bristol: Hilger*, 1986.

[49] G. Wahba, "Constrained regularization for ill posed linear operator equations, with applications in meteorology and medicine," in *Statistical decision theory and related topics III*, pp. 383–418, Elsevier, 1982.

[50] J. Kruitohf, "Telefoonverkeersrekening. de ingenieur ('s gravenhage)," in *ACM SIGMETRICS*, vol. 8, pp. 1–50, 1937.

[51] S. Eum, J. Murphy, and R. Harris, "Tomokruithof vs tomogravity for backbone networks," in *ATNAC*, Citeseer, 2004.

[52] B. Lamond and N. F. Stewart, "Bregman's balancing method," *Transportation Research Part B: Methodological*, vol. 15, no. 4, pp. 239–248, 1981.

[53] B. Yegnanarayana, *Artificial neural networks.* PHI Learning Pvt. Ltd., 2009.

[54] J. Zhao, H. Qu, J. Zhao, and D. Jiang, "Towards traffic matrix prediction with lstm recurrent neural networks," *Electronics Letters*, vol. 54, no. 9, pp. 566–568, 2018.

[55] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[56] F. Chollet, "How convolutional neural networks see the world," *The Keras Blog*, vol. 30, 2016.

[57] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture," *Computer Networks*, vol. 171, p. 107138, 2020.

[58] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[59] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[60] Y. Bengio, *Learning deep architectures for AI.* Now Publishers Inc, 2009.

[61] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[62] K. Rajawat, E. Dall'Anese, and G. B. Giannakis, "Dynamic network delay cartography," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2910–2920, 2014.

[63] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed network monitoring for ip and all-optical networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 241–252, 2007.

[64] D. Jiang and G. Hu, "Large-scale ip traffic matrix estimation based on backpropagation neural network," in *2008 First International Conference on Intelligent Networks and Intelligent Systems*, pp. 229–232, IEEE, 2008.

[65] M. Mardani and G. B. Giannakis, "Estimating traffic and anomaly maps via network tomography," *IEEE/ACM transactions on networking*, vol. 24, no. 3, pp. 1533–1547, 2015.

[66] L. Nie, "A novel network tomography approach for traffic matrix estimation problem in large-scale ip backbone networks," in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, pp. 97–101, IEEE, 2015.

[67] L. He, L. Yan, and J. Zhao, "Research of traffic matrix estimation in ip networks," in *2011 International Conference on Internet Technology and Applications*, pp. 1–4, IEEE, 2011.

[68] M. Roughan, "Simplifying the synthesis of internet traffic matrices," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 93–96, 2005.

[69] R. A. Memon, S. M. Atif, and S. Qazi, "First elephants then mice: A two-stage robust estimation technique for traffic matrix in large cloud networks in presence of over dispersion," in *TENCON 2018-2018 IEEE Region 10 Conference*, pp. 1301–1306, IEEE, 2018.

[70] K. Christodoulopoulos, N. Sambo, and E. Varvarigos, "Exploiting network kriging for fault localization," in *Optical Fiber Communication Conference*, pp. W1B–5, Optical Society of America, 2016.

[71] D. B. Chua, E. D. Kolaczyk, and M. Crovella, "Network kriging," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2263–2272, 2006.

[72] K. Rajawat, E. Dall'Anese, and G. B. Giannakis, "Dynamic network kriging," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 121–124, IEEE, 2012.

[73] V. W. Bandara, A. P. Jayasumana, and R. Whitner, "An adaptive compressive sensing scheme for network tomography based fault localization," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1290–1295, IEEE, 2014.

[74] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, pp. 73–84, 2004.

[75] L. Nie, D. Jiang, and Z. Lv, "Modeling network traffic for traffic matrix estimation and anomaly detection based on bayesian network in cloud computing networks," *Annals of Telecommunications*, vol. 72, no. 5, pp. 297–305, 2017.

[76] T. He, A. Gkelias, L. Ma, K. K. Leung, A. Swami, and D. Towsley, "Robust and efficient monitor placement for network tomography in dynamic networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1732–1745, 2017.

[77] X. Liu, W. Sun, W. Lou, Q. Pei, and Y. Zhang, "One-tag checker: Message-locked integrity auditing on encrypted cloud deduplication storage," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[78] T. He, L. Ma, A. Gkelias, K. K. Leung, A. Swami, and D. Towsley, "Robust monitor placement for network tomography in dynamic networks," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[79] W. Dong, Y. Gao, W. Wu, J. Bu, C. Chen, and X.-Y. Li, "Optimal monitor assignment for preferential link tomography in communication networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 210–223, 2016.

[80] N. Ogino, T. Kitahara, S. Arakawa, G. Hasegawa, and M. Murata, "Decentralized boolean network tomography based on network partitioning," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 162–170, IEEE, 2016.

[81] S. Qazi and T. Moors, "On the impact of routing matrix inconsistencies on statistical path monitoring in overlay networks," *Computer Networks*, vol. 54, no. 10, pp. 1554–1572, 2010.

[82] S. Qazi and T. Moors, "Practical issues of statistical path monitoring in overlay networks with large, rank-deficient routing matrices," in *2008 5th International Conference on Broadband Communications, Networks and Systems*, pp. 396–403, IEEE, 2008.

[83] S. Qazi, R. A. Memon, and A. A. Farooqui, "Evaluation of performance of genetic algorithms for network tomography," *Journal of Internet Technology*, vol. 16, no. 1, pp. 75–85, 2015.

[84] M. Koot and F. Wijnhoven, "Usage impact on data center electricity needs: A system dynamic forecasting model," *Applied Energy*, vol. 291, p. 116798, 2021.

[85] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in sdn-openflow networks," *Computer Networks*, vol. 71, pp. 1–30, 2014.

[86] J. Mao, X. Li, Q. Lin, and Z. Guan, "Deeply understanding graph-based sybil detection techniques via empirical analysis on graph processing," *China Communications*, vol. 17, no. 10, pp. 82–96, 2020.

[87] N. E. Rad, Y. Ephraim, and B. L. Mark, "Delay network tomography using a partially observable bivariate markov chain," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 126–138, 2016.

[88] D. Daniel-Simion and G. Dan-Horia, "Traffic shaping and traffic policing impacts on aggregate traffic behaviour in high speed networks," in *2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 465–467, IEEE, 2011.

[89] D. Jiang, Z. Zhao, Z. Xu, C. Yao, and H. Xu, "How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 10, pp. 915–925, 2014.

[90] F. Qian, G. Hu, and J. Xie, "A recurrent neural network approach to traffic matrix tracking using partial measurements," in *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pp. 1640–1643, IEEE, 2008.

[91] S. Qazi, S. M. Atif, and M. B. Kadri, "A novel compressed sensing technique for traffic matrix estimation of software defined cloud networks," *arXiv preprint arXiv:1811.02165*, 2018.

[92] P. Tune and M. Roughan, "Spatiotemporal traffic matrix synthesis," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 579–592, 2015.

[93] X. Fan and X. Li, "Network tomography via sparse bayesian learning," *IEEE Communications Letters*, vol. 21, no. 4, pp. 781–784, 2017.

[94] L. Nie and D. Jiang, "A compressive sensing-based network tomography approach to estimating origin–destination flow traffic in large-scale backbone networks," *International Journal of Communication Systems*, vol. 28, no. 5, pp. 889–900, 2015.

List of Publications

1. Rashida A. Memon, S Qazi, and B M Khan, "Design and Implementation of a Robust Convolutional Neural Network-Based Traffic Matrix Estimator for Cloud Networks", Wireless Communications and Mobile Computing, June 2021 Volume 2021, Article ID 1039613, 11 pages (I.F = 2.33)

https://doi.org/10.1155/2021/1039613

2. Rashida A. Memon, S M Atif, and S Qazi "First Elephants then Mice: A Two-stage Robust Estimation Technique for Traffic Matrix in Large Cloud Networks in Presence of Over Dispersion", in proceedings of the IEEE Region 10 Conference TENCON 2018, pp. 1301-1306, Jeju, South Korea, 2018.

https://doi.org/10.1109/TENCON.2018.8650063