CHAPTER 1

INTRODUCTION

This chapter includes objective of the project, the problems to be solved, purpose, scope and challenges to be solved by the project and organisation of the report.

1.1 OVERVIEW

Online social networks, World Wide Web, media and technological networks, and other types of so-called information networks are ubiquitous nowadays. These information networks are inherently heterogeneous and dynamic. They are heterogeneous as they consist of multi-typed objects and relations, and they are dynamic as they are constantly evolving over time. One of the challenging issues in such heterogeneous and dynamic environments is to forecast those relationships in the network that will appear in the future. In this project,try to solve the problem of continuous-time relationship prediction in dynamic and heterogeneous information networks. This implies predicting the time it takes for a relationship to appear in the future, given its features that have been extracted by considering both heterogeneity and temporal dynamics of the underlying network. To this end, introduce a feature extraction framework that combines the power of meta-path-based modelling and recurrent neural networks to effectively extract features suitable for relationship prediction regarding heterogeneity and dynamically of the networks. Next, propose a supervised non-parametric approach, called Non-Parametric Generalised Linear Model (Np-Glm), which infers the hidden underlying probability distribution of the relationship building time given its features.

1.2 OBJECTIVES

- Extending the link prediction problem in homogeneous networks to relationship prediction in heterogeneous networks, by systematically defining the target relation and topological features in heterogeneous networks.

- Extending the traditional prediction problem from “whether it will happen” to “when will it happen”, and directly model the relationship building time as a function of topological features.

1.3 PROBLEM STATEMENT

- Effective utilisation of temporal evolution of the topological similarity measures for link prediction.

- Prediction of new and recurring links.

- Temporal link prediction model based on the supervised learning techniques to utilise the correlations among similarity measures.

1.4 ORGANISATION OF THE REPORT

In this report, **Chapter 2** gives background on interactive discussion of some related works. **Chapter 3** describes the overall system design, algorithm and detailed description about each module. **Chapter 4** describes about dataset used for the implementation of the proposed framework, results of the experiments conducted, test cases, performance evaluation and comparison of the results of proposed system with existing system. **Chapter 5** discusses about the conclusion and scope for future work and finally the references.

CHAPTER 2

LITERATURE REVIEW

This chapter discusses the various ideas, methodologies that have incorporated in this project.

2.1 TEMPORAL LINK PREDICTION

Dunlavy et al.(2017) focused on the problem of periodic temporal link prediction. They concentrated on bipartite graphs that evolve over time and also considered a weighted matrix that contained multilayer data and tensor-based methods for predicting future links. Oyama et al. solved the problem of cross-temporal link prediction, in which the links among nodes in different time frames are inferred. They mapped data objects in different time frames into a common low-dimensional latent feature space and identified the links on the basis of the distance between the data objects.

2.2 MULTIVARIATE TIME SERIES

Özcan et al.(2010) proposed a novel link prediction method for evolving networks based on NARX neural network. They take the correlation between the quasi-local similarity measures and temporal evolutions of link occurrences information into account by using NARX for multivariate time series forecasting. Yu et al. developed a novel temporal matrix factorisation model to explicitly represent the network as a function of time. They provided results for link prediction as a specific example and showed that their model performs better than the state-of-the-art techniques.

2.3 CONTINUOUS TIME RELATIONSHIP PREDICTION

Aggarwal et al.(2012) tackle the link prediction problem in both dynamic and heterogeneous information networks using a dynamic clustering approach alongside content-based and structural models. However, they aim to solve the conventional link prediction problem, not the continuous-time relationship prediction studied in this paper. The authors proposed a feature set, called TMLP, well suited for link prediction in dynamic and heterogeneous information networks. Although their proposed feature set copes with both dynamicity and heterogeneity of the network, it cannot be extended for the generalised problem of relationship prediction and is only designed for solving the simpler link prediction problem.

2.4 GENERALISED LINEAR MODEL

Sun et al.(2011) proposed work that has focused on the continuous-time relationship prediction problem,in which a generalised linear model based framework is suggested to model the relationship building time. They consider the building time of links as independent random variables coming from a pre-specified distribution and model the expectation as a function of a linear predictor of the extracted topological features. A shortcoming of this model is that we need to exactly specify the underlying distribution of relationship building times. Came over this problem by learning the distribution from the data using a non-parametric solution. Furthermore, this project considered the temporal dynamics of the network which has been entirely ignored in their work.

CHAPTER 3

SYSTEM DESIGN

This chapter discusses the overall system architecture and detailed description of all modules.

3.1 PROPOSED SYSTEM

The proposed system of the project is to satisfy the below criteria's:

- The proposed feature extraction framework can tackle heterogeneity of the data as well as capturing the temporal dynamics of the network by incorporating meta-path-based features into a recurrent neural network based auto encoder.

- Our non-parametric model takes a unique approach toward learning the underlying distribution of relationship building time without imposing any significant assumptions on the problem.

- To the best of our knowledge, this paper is one which studies the continuous-time relationship prediction problem in both dynamic and heterogeneous network configurations.

3.2 BLOCK DIAGRAM

Figure 3.1 describes the overall system of the project where the details of the module used for semantic and structural similarity in order to predict the link between the co-authors.

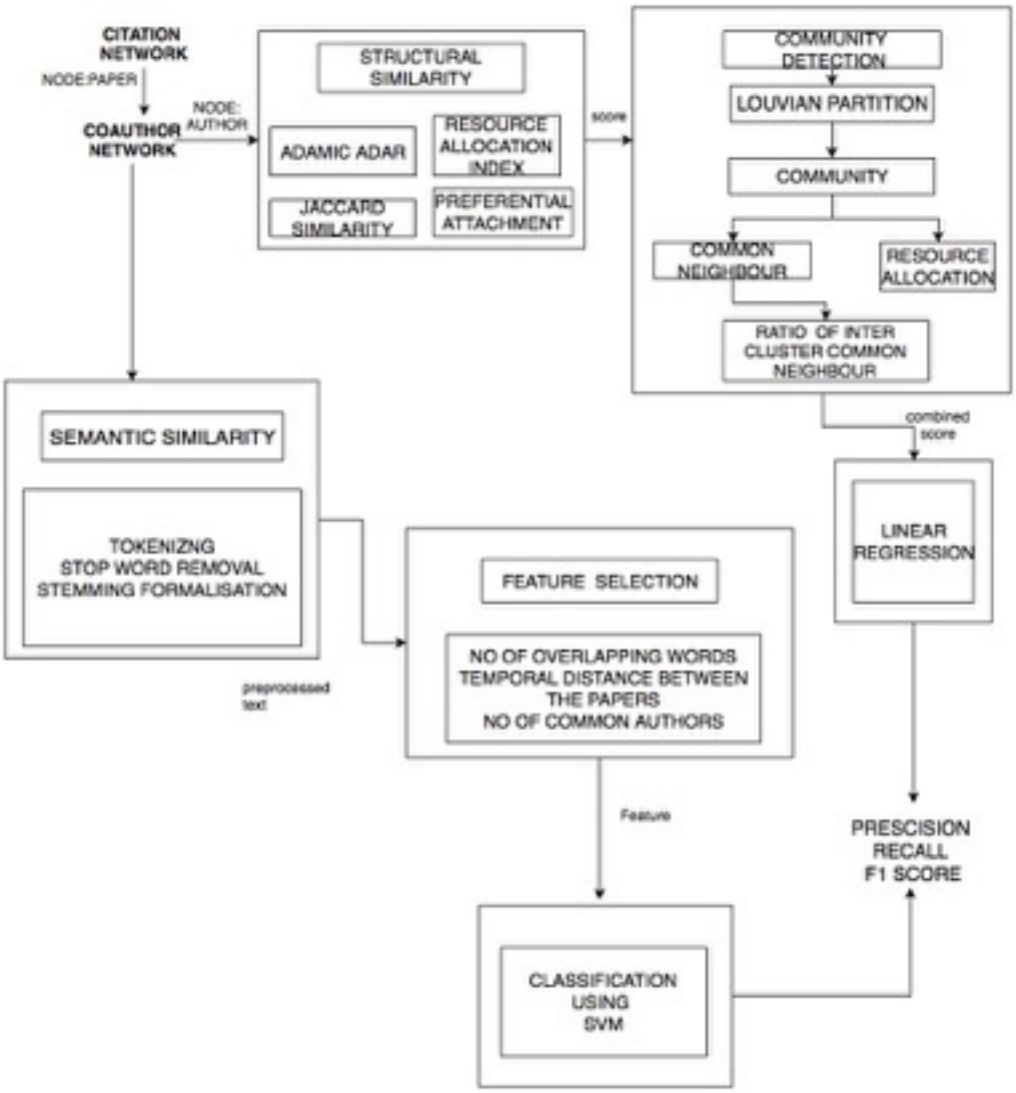

Fig 3.1 System architecture for phase1

Figure 3.1 This implies predicting the time it takes for a link to appear in the future, given its features that have been extracted at the current network snapshot. To this end, introduce a probabilistic non- parametric approach, called Non-Parametric Generalised Linear Model (NP-GLM), which infers the hidden underlying probability distribution of the link advent time given its features.

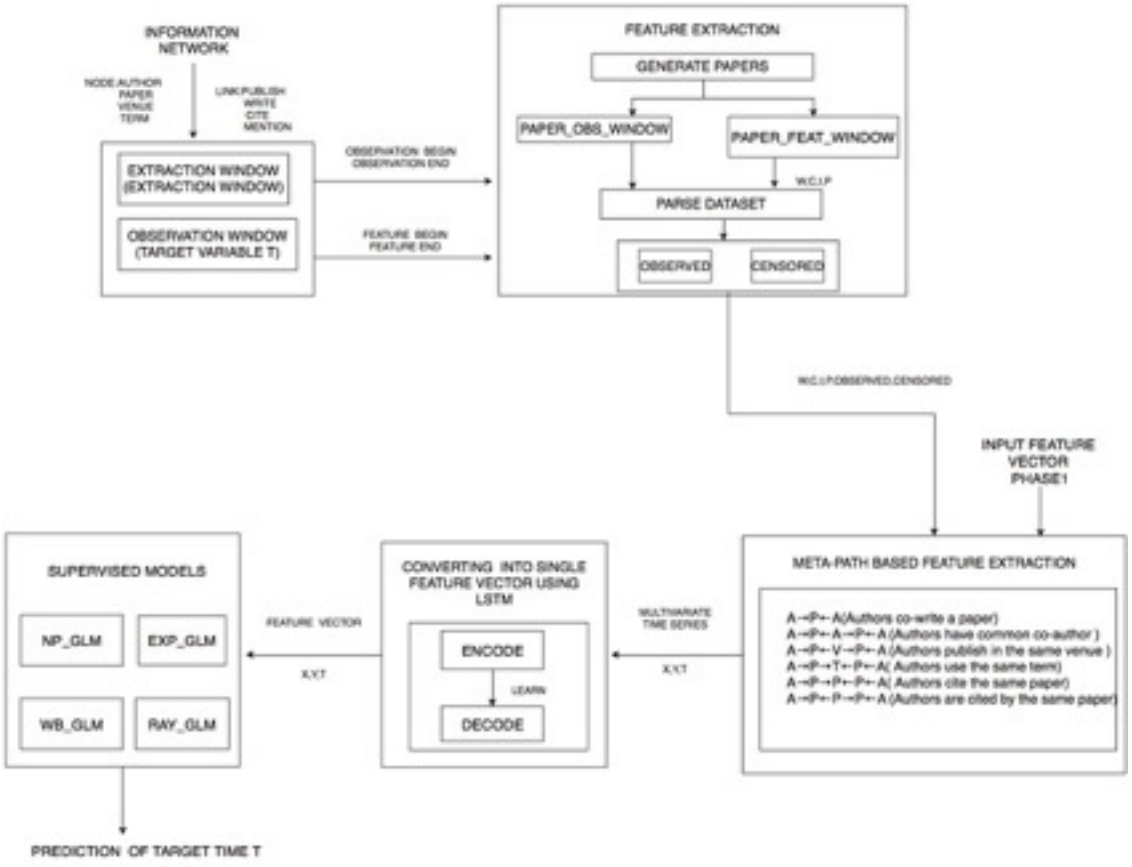

Fig 3.2 System architecture for phase2

3.3 LIST OF MODULES

This section elaborately describes about the various modules of the proposed system. It also provides the algorithm details of each modules and depicts the snapshot of the results obtained. The modules of the proposed system are:

- Splitting into Intervals
- Feature Extraction
- MetaPath Based Extraction
- Converting Into Single Feature Vector Using LSTM
- Supervised Model

3.3.1 Splitting into Intervals

In this module given a heterogeneous network G with schema $S_G = (\mathcal{V}, \mathcal{E})$, for each link type $\varepsilon \in E$ denoting the relation between node types $v_i, v_j \in \mathcal{V}$, the heterogeneous adjacency matrix M_ε is a binary $|V_{vi}| \times |V_{vj}|$ matrix representing whether nodes of type v_i are in relation with nodes of type v_j with link type ε or not. For instance, in the bibliographic network, the heterogeneous adjacency matrix M_{write} is a binary matrix where each row is associated with an author and each column is associated with a paper, and $M_{write}(i, j)$ indicates if the author i has written the paper j.

In this project, consider the case that an information network is both dynamic and heterogeneous. This means that all network entities are associated with a type, and can possibly have birth and death times, regardless of their types. The bibliographic network is an example of both dynamic and heterogeneous one. Whenever a new paper is published, a new Paper node will be added to the network, alongside with

the corresponding new Author, Term, and Venue nodes (if they don't exist yet). New links will be formed among these newly added nodes to indicate the write, publish and mention relationships. Some linkages might also form between the existing nodes and the new ones, like new cite links connecting the new paper with the existing papers in its reference list.After creating the heterogeneous network it need to be split intervals which is described in algorithm 3.3.1.

ALGORITHM FOR SPLITTING INTO INTERVALS:

- **STEP1**: Split interval into two parts: the first part for extracting the feature x, and the second for determining the target variable t.
- **STEP2**: We refer to the first interval as Feature Extraction Window whose length is denoted by Φ, and the second as Observation Window, whose length is denoted by Ω.
- **STEP3**: Based on the existence of the target relationship in the observation window, all the node pairs in the network will fall within either one of the following three different groups:

(1) Node pairs that form the target relationship before the beginning of the observation window (in the feature extraction window).

(2) Node pairs that form the target relationship in the observation window for the first time (not existing before in the feature extraction window).

(3) Node pairs that do not form the target relationship (neither in the feature extraction window nor in the observation window).

3.3.2 Feature Extraction

In this module,presenting our feature extraction framework, algorithm is described below, that is designed to have three major characteristics: First, it effectively considers different type of nodes and links available in a heterogeneous information network and regards their impact on the building time of the target relationship. Second, it takes the temporal dynamics of the network into account and leverages the network evolution history instead of simply aggregating it into a single snapshot. Finally, the extracted features are suitable for not only the link prediction problem but also the generalised relationship prediction.Will incorporate these features in the proposed non-parametric model in Section 4 to solve the continuous-time relationship prediction problem.

ALGORITHM FOR FEATURE EXTRACTION:

• **STEP1:** Using generate_paper method we extract the author, author_id, paper_id, year of publication, conference and references from feature extraction widow and observation window.

• **STEP2:** Send extracted information to the parse dataset method which output the relationship between each author (write, cite, include and publish).

• **STEP3:** The feature extraction window is divided into k fragments of size Δ. In paper_obs_window, The target relationship between them has been created at a time like $tr \in (t0 + \Phi, t1]$. set $t = tr - (t0 + \Phi)$ as the time it takes for the relationship to form since the beginning of the observation window.

• **STEP4:** Haven't seen their exact building time, but we know that it should be definitely after t1. For such samples, that we call censored samples, we set $t = t1 - (t0 + \Phi)$ that is equal to the length of the observation window Ω.

• **STEP5:** After splitting samples into observed and censored samples along with write, cite, publish and include is passed for meta-path feature extraction.

3.3.3 Meta-Path Based Extraction

In this module how to utilise the temporal history of the network in the feature extraction window in order to extract features for continuous-time relationship prediction problem. We first begin with the meta-path-based feature set for heterogeneous information networks, and then incorporate these features into a recurrent neural network based auto encoder to exploit the temporal dynamics of the network as well. The concept of meta-path defined in the algorithm below, In a heterogeneous information, a meta-path is a directed path following the graph of the network schema to describe the general relations that can be derived from the network.

Formally speaking, given a network schema $S_G=(\mathcal{V},\mathcal{E})$,The sequence $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \ldots v_{k-1} \xrightarrow{e_{k-1}} v_k$ is a meta-path defined on S_G where $v_i \in \mathcal{V}$ and $e_i \in e$ where $i = 1 \ldots k$ Meta-paths are commonly used in heterogeneous information networks to describe multi-typed relations that have concrete semantic meanings. For example, in the bibliographic network whose schema is shown in Fig. 1a, can define the co-authorship relation by the following meta-path

Author $\xrightarrow{\text{write}}$ Paper $\xleftarrow{\text{write}}$ Author

or simply by A → P ← A. Another example is the author citation relation, which in this paper is used as the target relation for DBLP network. It can be specified as:

Author $\xrightarrow{\text{write}}$ Paper $\xrightarrow{\text{cite}}$ Paper $\xleftarrow{\text{write}}$ Author

Abbreviated as A→P →P ←A.

ALGORITHM FOR META-PATH BASED EXTRACTION:

• **STEP1:** A meta-path is a directed path following the graph of the network schema to describe the general relations given a network schema S_G=*(*$\mathcal{V}$*,*$\mathcal{E}$*)*,The sequence $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \ldots v_{k-1} \xrightarrow{e_{k-1}} v_k$ is a meta-path defined on S_G where $v_i \in \mathcal{V}$ and e_i where $i = 1 \ldots k$.

• **STEP2:** Relation between node types v_i, $v_j \in V$, the meta-path adjacency matrix MΨ is defined as:Dynamic heterogeneous network Gτ observed in a feature extraction window of size Φ (t0 < τ ≤ t0 + Φ), along with its network schema S_G = *(*$\mathcal{V}$*,* $\mathcal{E}$*)* and a target relation A ⇝ B.

• **STEP3:** Set an auxiliary variable y = 1 which indicates that we have observed their exact building time said to be OBSERVED SAMPLES.

• **STEP4: S**et y = 0 to indicate that the recorded time is, in fact, a lower bound on the true relationship building time said to be CENSORED SAMPLES. The feature extraction window is divided into k fragments of size Δ.

• **STEP5:** For a given pair of nodes a ∈ A and b ∈ B in $G^{t0+\Phi}$, and a meta-path Ψ defined on SG, the dynamic meta-path-based time series of (a, b) is calculated as: $X_\psi^i(a,b) = f_\psi^{t_0+i\Delta}(a,b) - f_\psi^{t_0+(i-1)\Delta}(a,b) \quad i = 1 \ldots k$. For each time step, the corresponding values from all the time series into a vector.

• **STEP6:** d feature meta-paths Ψ1 to Ψd, then each time step of the resulting time series for any node pair (a, b) will become: $X_{a,b}^i = \left[x_{\psi_1}^i(a,b), \ldots, x_{\psi_d}^i(a,b) \right]^T, i = 1 \ldots k$ this vector-valued time series referred as Multivariate Meta-Path-based Time Series.

3.3.4 Converting into Single Feature Vector Using LSTM

To convert the multivariate meta-path-based time series into a single feature vector so that we can use it as the input to our non-parametric model that will be discussed in the next section. A trivial solution would be to stack all the vector-valued time steps of the multivariate time series into a single vector. However, this approach will result in a very high dimensional vector as the number of time steps increases and can lead to difficulties in the learning procedure due to the curse of dimensionality. This is in contrast with our expectation that more time steps would bring more information about the history of the network and should result in a better prediction model. To overcome this problem, combine the power of recurrent neural networks, especially Long Short Term Memory (LSTM) units, which have proven to be very successful in handling time series and sequential data, with Auto encoders, which are widely used to learn alternative representations of the data such that the learned representation can reconstruct the original input. Our goal is to transform the multivariate meta-path-based time series into a compact vector representation such that the resulting vector holds as much information from the original multivariate time series as possible.

ALGORITHM FOR LSTM :

• **STEP1:** The auto encoder in this paper is to compress multivariate time series, the input to the encoder LSTM is a multivariate time series of length k.
• **STEP2:** The encoder accepts the vector-valued time steps of the input multivariate time series sequentially.
• **STEP3:** After receiving the kth time step, the output of the encoder LSTM will be the compressed feature vector that we will use as the input to the Np-Glm method.

• **STEP4:** To train the encoder to learn how to compress the input time series, it is matched with a decoder LSTM.

• **STEP5:** The decoder LSTM receives k copies of the compressed feature vector one after another, and with a proper loss function (such as mean squared error) it is forced to reconstruct the original multivariate time series in reverse order.

• **STEP6:** Reversing the output sequence will make the optimisation of the model easier since it causes the decoder to revert back the changes made by the encoder to the input sequence.

3.3.5 Supervised Model

In this module introducing proposed model, called Non-Parametric Generalised Linear Model, to solve the problem of continuous-time relationship prediction based on the extracted features. Since the relationship building time is treated as a continuous random variable, attempt to model the probability distribution of this time, given the features of the target relationship. Thus, if denoting the target relationship building time by t and its features by x, our aim is to model the probability density function f_T (t | x). A conventional approach to modelling this function is to x a parametric distribution for t (e.g. Exponential distribution) explained in algorithm 3.1.5 and then relate x to t using a Generalised Linear Model. The major drawback of this approach is that we need to know the exact distribution of the relationship building time, or at least, we could guess the best one that fits. The alternative way that follow is to learn the shape of f_T (t | x) from the data using a non-parametric solution. In the rest of this module, we first bring the necessary theoretical backgrounds related to the concept, then go through the details of the proposed model. In the end, explain the learning and inference algorithms of Np-Glm below.

ALGORITHM FOR SUPERVISED MODELS:

- **STEP1:** Set parameters converged ← False; threshold ← 10^{-4}; $\tau \leftarrow 0$;

  $L(\tau) = \infty$;

- **STEP2:** Initialise $w(\tau)$ with random values;

- **STEP3:** While Not converged do $\tau \leftarrow \tau + 1$;

  To obtain $H(\tau)$ using $w(\tau-1)$;

  To obtain $w(\tau)$ using $H(\tau)$;

  To obtain $L(\tau)$ using $w(\tau)$ and $H(\tau)$;

- **STEP4:** If $L(\tau) - L(\tau-1) <$ threshold then

converged ← True;

end

end $w \leftarrow w(\tau)$;

$H \leftarrow H(\tau)$

CHAPTER 4

IMPLEMENTATION AND RESULTS

This section discusses about the experiment adopted to efficiently evaluate the accuracy and performance of the proposed system against the existing approach.

4.1 DATASET DESCRIPTION

The dataset for citation network are downloaded from https://aminer.org/data from which co-author network were generated.

4.2 IMPLEMENTATION RESULTS

This section discusses the details about the modules of the proposed system and the output for each module.

4.2.1 Module Description for Splitting into intervals

To solve the problem of continuous-time relationship prediction in dynamic networks, we need to pay attention to the temporal history of the network data from two different points of view. First, we have to mind the evolutionary history of the network for feature extraction, so that the extracted features reflect the changes made in the network over time. Second, specify the exact relationship building time for each pair of nodes that have formed the target relationship. This is because our goal is to train a supervised model to predict a continuous variable, which in this case is the building time of the target relationship. Hence, for each sample pair of

nodes need a feature vector x, associated with a target variable t that indicates the building time of the target relationship between them.

Fig 4.1 The evolutionary timeline of the network data

Suppose that have observed a dynamic network Gτ recorded in the interval t0 < τ ≤ t1. According to Figure 4.1, we split this interval into two parts: the first part for extracting the feature x, and the second for determining the target variable t. Refer to the first interval as Feature Extraction Window whose length is denoted by Φ, and the second as Observation Window, whose length is denoted by Ω. Code for the above process is given in Figure 4.2.

```
    # delta = 1
    # observation_window = 3
    # n_snapshots = 5

    observation_end = 2016
    observation_begin = observation_end - observation_window
    feature_end = observation_begin
    feature_begin = feature_end - delta * n_snapshots

    papers_feat_window, papers_obs_window, counter =
generate_papers('data/dblp.txt', feature_begin, feature_end,

observation_begin, observation_end,

conf_list)
    file = open('split_interval.txt','w')
    file.write("------------------------
paper_feat_window--------------------------------")
    file.write(str(papers_feat_window))
    file.write("------------------------
paper_obs_window---------------------------------")
    file.write(str(papers_obs_window))
```

Fig 4.2 Code for splitting into intervals.

4.2.2 Module Description for Feature Extraction

In this project, considering the case that an information network is both dynamic and heterogeneous. This means that all network entities are associated with a type, and can possibly have birth and death times, regardless of their types. The bibliographic network is an example of both dynamic and heterogeneous one. Below Figure 4.3 shows the nodes and links of heterogeneous graph which is extracted from dataset.

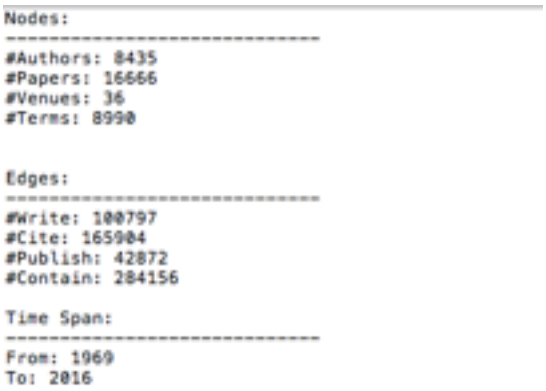

Fig 4.3Displays the result of heterogeneous graph.

Whenever a new paper is published, a new Paper node will be added to the network, alongside with the corresponding new Author, Term, and Venue nodes (if they don't exist yet). New links will be formed among these newly added nodes to indicate the write, publish and mention relationships. Some linkages might also form between existing nodes and the new ones, like new cite links connecting the new paper with the existing papers in its reference list.Before extracting features from feature extraction window parsing is done which is depicted in Figure 4.4.

Fig 4.4 Result after parsing the dataset.

In order to formally describe the state of a heterogeneous and dynamic network at any timestamp τ, project define the time-aware heterogeneous adjacency matrix. Following Figure 4.5 shows the result after splitting intervals authors in both feature extraction window and observation window are grouped into coo matrix and after parsing the dataset, from coo matrix authors who have value of 1 is displayed below.

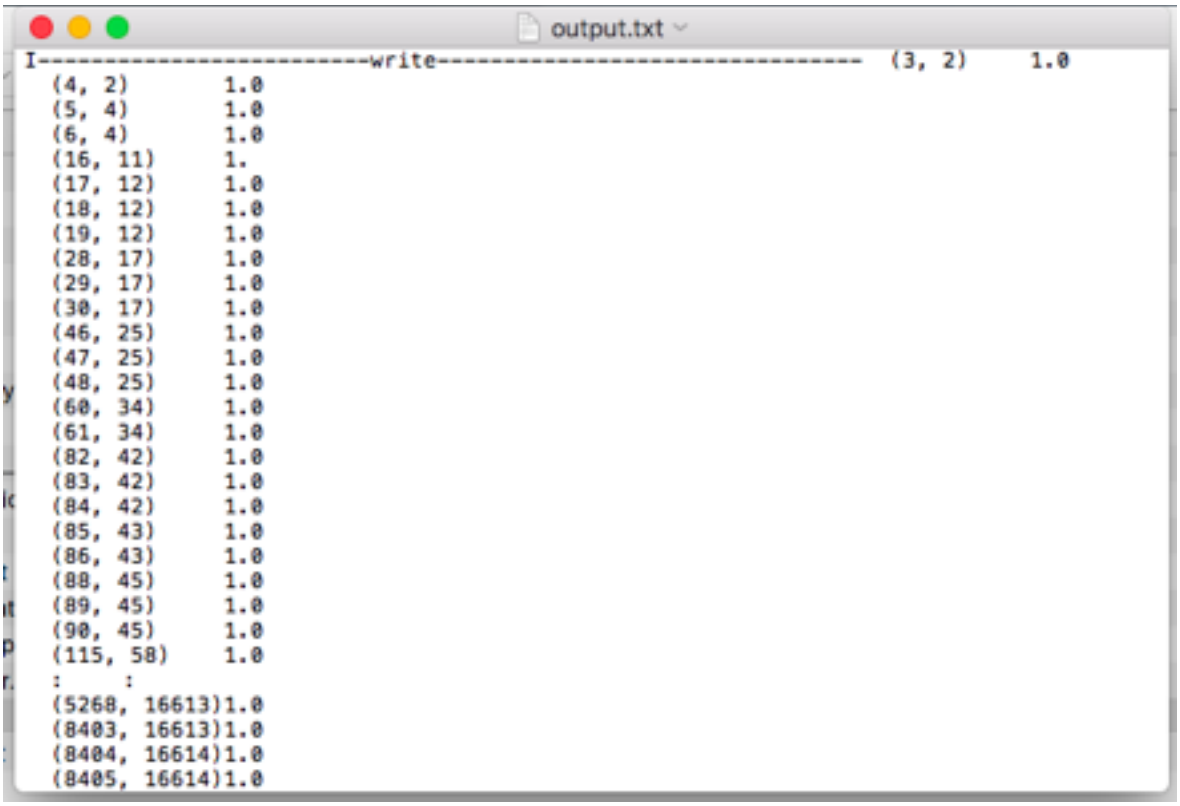

Fig 4.5 Displays the result for authors write, cite, include, publish and cite.

4.2.3 Module Description for MetaPath based Feature Extraction

Meta-path is a path defined over a network schema, and denotes a composite relation over a heterogeneous network. Each meta-path defines a unique topology between objects, and can be used to define the topological features with different semantic meanings. In case study of meta-path preparation for co-authorship prediction is given. In this section, we give a general frame- work in preparing reasonable meta path-based topological features for the target relation.Before obtaining our metapath based feature extraction samples are split into observed samples and censored which is depicted in figure For a target relation RT = ⟨A,B⟩, any meta- paths starting with type A and ending with type B other than the target relation itself can be used as the topological features. These meta-paths can be

obtained by traversing on the network schema, for example, using BFS (breadth-first search).

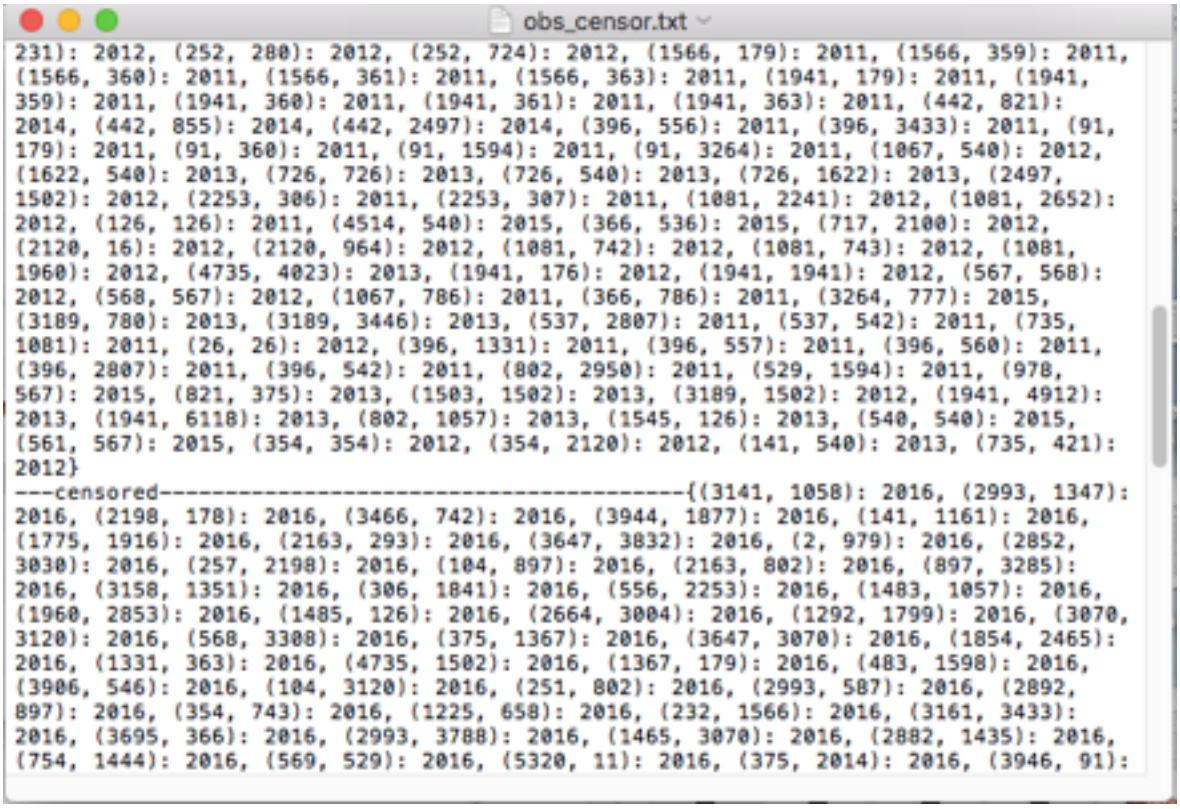

Fig 4.6 After converting into observed and censored sample.

Table 4.1 Meta-Paths Denoting Similarity Relations between Authors.

META-PATH	SEMANTIC MEANING
A→P←A	Authors co-write a paper
A→P←A→P←A	Authors have common co-author
A→P←V→P←A	Authors publish in the same venue
A→P→T←P←A	Authors use the same term
A→P→P←P←A	Authors cite the same paper
A→P←P→P←A	Authors are cited by the same paper

Taking the author citation relation, which is defined as A − P → P − A, as the target relation, consider 6 author-author similarity relations defined in Table 1. For each

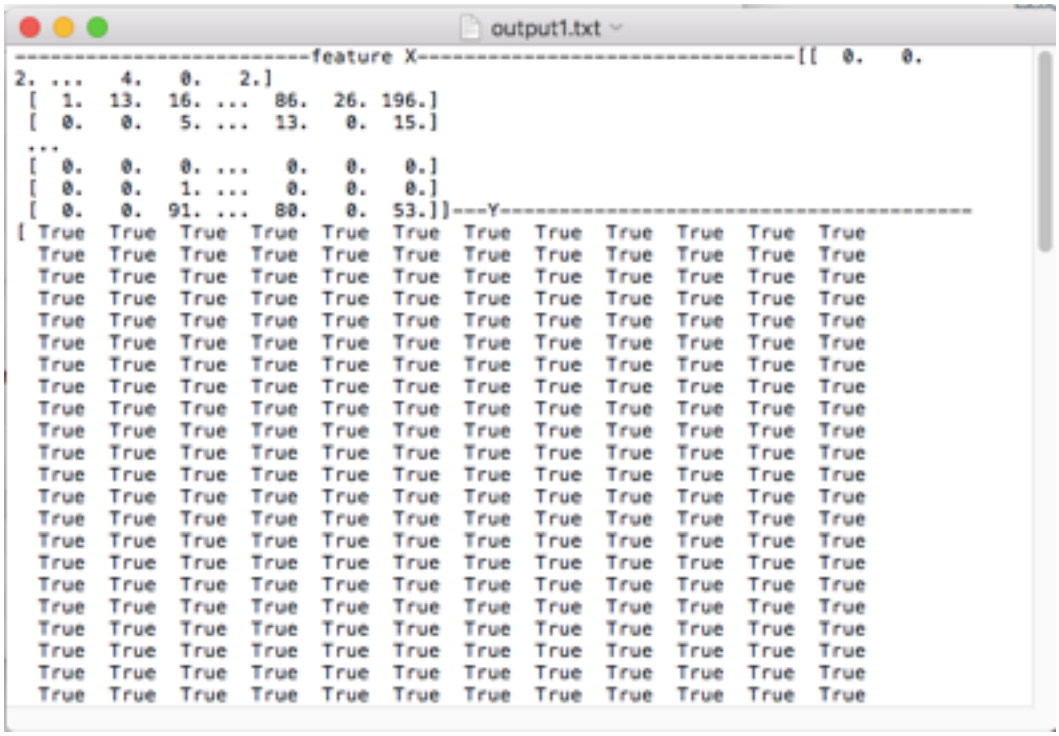

Fig 4.7 Meta path extracted feature (X,Y,T).

similarity relation, concatenate the target relation in its left side or in its right side. Then have 12 topology features with the form ARsimART B and ART BRsimB in total.

Besides, consider the concatenation of “author- cites-paper” relation (A − P → P) and “paper-cites-author” relation (P → P − A), as well as all the 6 similarity relations listed in Table 4.1, in the form of AR1CR2B.After obtaining the metapath based feature extraction output obtained in depicted in the Figure 4.6.

4.2.4 Module Description for LSTM:

The auto encoder consists of two components: (1) the encoder, which takes the input data and transforms it into a latent representation; and (2) the decoder, which takes the encoded representation and transforms it back to the input space. The auto encoder is trained in such a way that it can reconstruct the original input data. The purpose of using the auto encoder in this paper is to compress multivariate time series, instead of using simple feed-forward neural networks, both encoder and decoder are built using LSTMs. The input to the encoder LSTM is a multivariate time series of length k. The encoder accepts the vector-valued time steps of the input multivariate time series sequentially. After receiving the kth time step, the output of the encoder LSTM will be the compressed feature vector that we will use as the input to the Np-Glm method. In order to train the encoder to learn how to compress the input time series, it is matched with a decoder LSTM. The decoder LSTM receives k copies of the compressed feature vector one after another, and with a proper loss function (such as mean squared error) it is forced to reconstruct the original multivariate time series in reverse order. Reversing the output sequence will make the optimisation of the model easier since it causes the decoder to revert back the changes made by the encoder to the input sequence. The result obtained from lstm is depicted in Figure 4.7.

```
In [12]: X_raw = encode(X_list, epochs=100, latent_factor=2)

Epoch 1/100
556/556 [==============================] - 1s 3ms/step - loss: 0.4864
Epoch 2/100
556/556 [==============================] - 0s 102us/step - loss: 0.4634
Epoch 3/100
556/556 [==============================] - 0s 97us/step - loss: 0.4400
Epoch 4/100
556/556 [==============================] - 0s 91us/step - loss: 0.4149
Epoch 5/100
556/556 [==============================] - 0s 110us/step - loss: 0.3876
Epoch 6/100
556/556 [==============================] - 0s 95us/step - loss: 0.3588
Epoch 7/100
556/556 [==============================] - 0s 113us/step - loss: 0.3301
Epoch 8/100
556/556 [==============================] - 0s 108us/step - loss: 0.3035
Epoch 9/100
556/556 [==============================] - 0s 89us/step - loss: 0.2808
Epoch 10/100
556/556 [==============================] - 0s 90us/step - loss: 0.2624
Epoch 11/100
556/556 [==============================] - 0s 92us/step - loss: 0.2478
Epoch 12/100
556/556 [==============================] - 0s 98us/step - loss: 0.2362
Epoch 13/100
556/556 [==============================] - 0s 91us/step - loss: 0.2269
Epoch 14/100
556/556 [==============================] - 0s 90us/step - loss: 0.2194
Epoch 15/100
556/556 [==============================] - 0s 91us/step - loss: 0.2133
Epoch 16/100
556/556 [==============================] - 0s 105us/step - loss: 0.2084
Epoch 17/100
```

Fig 4.8 Converting multiple variate time series into single feature vector using LSTM.

The benefits of using the LSTM auto encoder is three-fold: (1)The auto encoder can re-construct the original time series, which reflects the temporal dynamics of the network get minimum information loss in the compressed feature vector; (2)Can set the dimensionality of the compressed feature vector to any desired value can evade the curse of dimensionality.

4.2.4 Module Description for supervised non-parametric model:

Non-Parametric Generalised Linear Model is used to solve the problem of continuous-time relationship prediction based on the extracted features. Since the relationship building time is treated as a continuous random variable, Attempt to model the probability distribution of this time, given the features of the target relationship. Thus, if denote the target relationship building time by t and its features by x, our aim is to model the probability density function $f_T(t \mid x)$. A conventional approach to modelling this function is to x a parametric distribution for t (e.g. Exponential distribution) and then relate x to t using a Generalised Linear Model . The major drawback of this approach is that need to know the exact distribution of the relationship building time, or at least, Could guess the best one that ts. The alternative way that follow is to learn the shape of $f_T(t \mid x)$ from the data using a non-parametric solution.

In the rest of this section, the necessary theoretical backgrounds related to the concept, then we go through the details of the proposed model. In the end, explain the learning and inference algorithms of Np-Glm.

Generally, the formation of a relationship between two nodes in a network can simply be considered as an event with its occurring time as a random variable T coming from a density function $f_T(t)$ Regarding this, can have the following definitions:

Survival Function:Given the density $f_T(t)$, the survival function denoted by S(t), is the probability that an event occurs after a certain value of t , which means:

$$S(t)=P(T>t)=\int_t^{\infty} f_T(t)\,dt \qquad (4.1)$$

Intensity Function:The intensity function (or failure rate function), denoted by λ(t), is the instantaneous rate of event occurring at any time t given the fact that the event has not occurred yet:

$$\boldsymbol{\lambda}(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T \leq t + \Delta t \mid T \geq t)}{\Delta t} \tag{4.2}$$

Cumulative Intensity Function: The cumulative intensity function, denoted by$\Lambda(t)$, is the area under the intensity function up to a point t :

$$\wedge(t) = \int_0^t \boldsymbol{\lambda}(t)\,dt \tag{4.3}$$

The relations between density, survival, and intensity functions come directly from their definitions as follows:

$$\boldsymbol{\lambda}(t) = \frac{f_T(t)}{s(t)} \tag{4.4}$$

$$s(t) = \exp(-\wedge(t)) \tag{4.5}$$

$$f_T(t) = \boldsymbol{\lambda}(t)\exp(-\wedge(t)) \tag{4.6}$$

Looking at Eq.4.6, The density function can be specified uniquely with its intensity function. Since the intensity function often has a simpler form than the density itself, if learn the shape of the intensity function, then can infer the entire distribution eventually. Therefore, focus on learning the shape of the conditional intensity function λ(t | x) from the data, and then accordingly infer the conditional density function $f_T(t \mid x)$ based on the learned intensity. In order to reduce the

hypothesis space of the problem and avoid the curse of dimensionality, assuming that λ(t | x), which is a function of both t and x, can be factorised into two separate positive functions as the following:

$$\boldsymbol{\lambda}(t \mid x) = g(\boldsymbol{W}^{T} x)h(t) \tag{4.7}$$

where g is a function of x which captures the e ect of features via a linear transformation using coefficient vector w independent of t, and h is a function of t which captures the effect of time independent of x. This assumption, referred to as proportional hazards condition [3], holds in GLM formulations of many event-time modelling distributions, such as the ones shown in Table 4.2. Our goal is now to x the function g and then learn both the coefficient vector w and the function h from the training data. In order to do so, we begin with the likelihood function of the data which can be written as follows:

$$\prod_{i=1}^{N} f_T(t_i|x_i)^{y_i} P(T \geq t_i|x_i)^{1-y_i} \tag{4.8}$$

The likelihood consists of the product of two parts: The rest part is the contribution of those samples for which we have observed their exact building time, in terms of their density function. The second part on the other hand, is the contribution of the censored samples, for which use the probability of the building time being greater than the recorded one. By applying Eq.4.5 and 4.6 can write the likelihood in terms of the intensity function:

$$\prod_{i=1}^{N} \boldsymbol{\lambda}(t_i|x_i)\exp\{-\wedge(t_i|x_i)\}^{y_i}\exp\{-\wedge(t_i|x_i)\}^{1-y_i} \tag{4.9}$$

By merging the exponentials and applying Eq.4.3 and 4.7, the likelihood function becomes:

$$\prod_{i=1}^{N}[g(W^T X_i)h(t_i)]^{y_i}\exp\{-g(W^T X_i)\}^{1-y_i} \tag{4.10}$$

Since don't know the form of h(t), we cannot directly calculate the integral appeared in the likelihood function. To deal with this problem, treat h(t) as a non-parametric function by approximating it with a piecewise constant function that changes just in ti s. Therefore, the integral over h(t), denoted by H (t), becomes a series:

$$h(t_i)=\int_0^{t_i} h(t)dt \simeq \sum_{j=1}^{i} h(t_j)(t_j - t_{j-1}) \tag{4.11}$$

assuming samples are sorted by t in increasing order, without loss of generality. The function H (t)de ned above plays an important role in both learning and inference phases. In fact, both the learning and inference phases rely on H(t) instead of h(t), which will see later in this paper. Replacing the above series in the likelihood, taking the logarithm and negating,end up with the following negative log-likelihood function, simply called the loss function, denoted by L:

$$L(W,h)=\sum_{j=1}^{N}\left\{g(W^T X_i)\sum_{j=1}^{i} h(t_j)(t_j - t_{j-1}) - y_i\left[\log g(W^T X_i)+\log h(t)\right]\right\} \tag{4.12}$$

The loss function depends on both the vector w and the function h(t). In the next part, explain an iterative learning algorithm to learn both w and h(t) collectively.

Minimising the loss function (Eq.4.12) relies on the choice of the function g. There are no particular limits on the choice of g except that it must be a non-negative function. For example, both quadratic and exponential functions of $W^T X$ will do the trick. Here, we proceed with $g(W^T X)=\exp(W^T X)$ since it makes the loss function convex with respect to w. Subsequent equations can be derived for other choices of g analogously.

Setting the derivative of the loss function with respect to $h(t_k)$ to zero yields a closed form solution for $h(t_k)$:

$$h(t_k) = \frac{y_k}{(t_k - t_{k-1}) \sum_{i=k}^{N} \exp(W^T X_k)} \tag{4.13}$$

By applying Eq. 4.11, we get the following for $H(t_i)$:

$$H(t_i) = \sum_{j=1}^{i} \frac{y_j}{\sum_{k=j}^{N} \exp(W^T X_k)} \tag{4.14}$$

which depends on the vector w. On the other hand, Cannot obtain a closed form solution for w from the loss function. Therefore turn to use Gradient-based optimisation methods to find the optimal value of w. The loss function with respect to w is as follows:

$$L(W) = \sum_{i=1}^{N} \left\{ \exp(W^T X_i) H(t_i) - y_i W^T X_i \right\} + const \tag{4.15}$$

which depends on the function H . As the learning of both w and H depends on each other, they should be learned collectively. Here,Use an iterative algorithm to learn w and H alternatively. Begin with a random vector w(0) . Then in each iteration τ , we first update $H^{(\tau)}$ via Eq.4.14 using $W^{\tau-1}$ .Next,we optimise Eq.4.15 $H^{(\tau)}(t_i)$ using the values of to obtain w(τ).

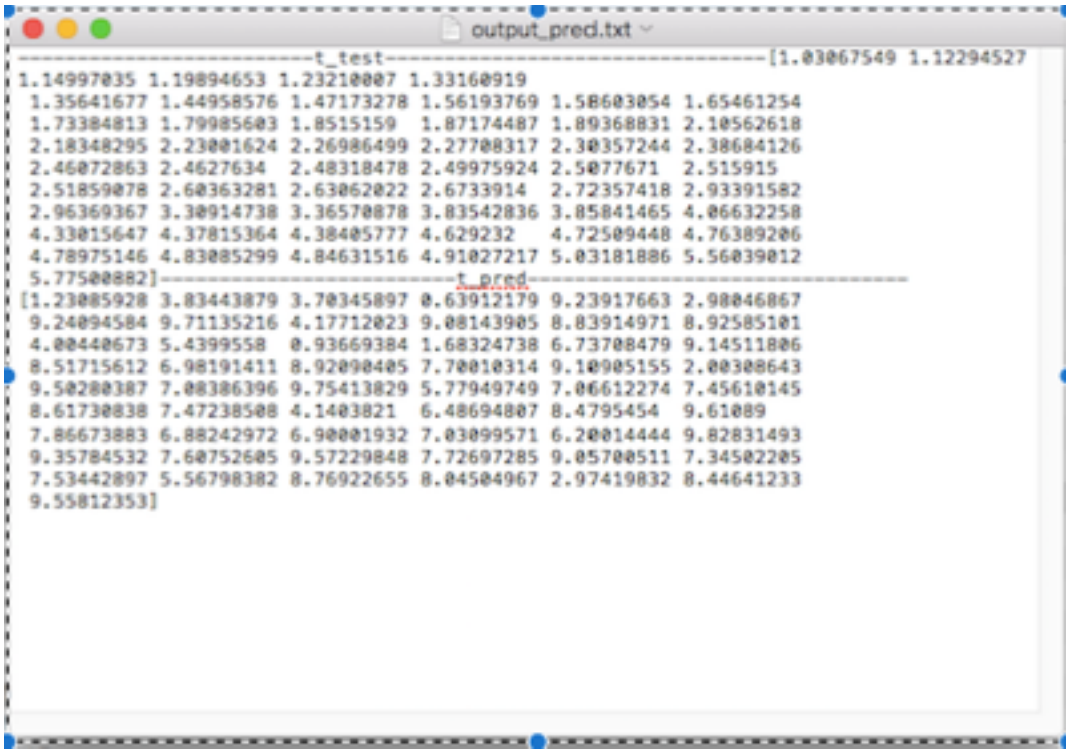

Fig 4.9 Prediction of time using Np-GLM giving censored and observed samples as test samples.

Test samples are given to prediction by giving censored and observed sample depicted in Fig 4.9,Since this procedure successively reduces the value of the loss function, and as the loss function (i.e. the negative log-likelihood) is bounded from below, the algorithm will ultimately converge to a stationary point. The pseudo code of the learning procedure is given in Algorithm 3.3.5.

```
556/556 [==============================] - 0s 76us/step - loss: 0.1884
Autoencoder Training Loss: 0.1884
556
[================================================================] 100%
--- 39.921630859375 seconds ---
```

	MAE	MRE	RMSE	MSLE	MDAE	CI	ACC-1	ACC-2	ACC-3	ACC-4	ACC-5	ACC-6
NP-Glm	2.18	1.04	2.58	0.36	2.22	0.53	0.13	0.27	0.35	0.44	0.58	0.69
Wbl-Glm	2.58	1.22	3.10	0.48	2.41	0.57	0.09	0.18	0.36	0.47	0.51	0.62
Exp-Glm	3.75	1.72	4.58	0.73	3.24	0.50	0.09	0.13	0.29	0.36	0.38	0.47
Ray-Glm	4.24	1.97	5.07	0.79	4.26	0.42	0.05	0.20	0.27	0.29	0.29	0.36
NP-Glm_stat	2.04	0.94	2.50	0.29	1.67	0.47	0.16	0.29	0.45	0.58	0.65	0.67
Wbl-Glm_stat	2.53	1.15	2.97	0.38	2.27	0.51	0.04	0.16	0.33	0.45	0.58	0.64
Exp-Glm_stat	3.38	1.46	4.00	0.54	2.90	0.41	0.02	0.15	0.25	0.33	0.40	0.51
Ray-Glm_stat	4.38	1.89	4.94	0.71	4.18	0.28	0.05	0.13	0.13	0.15	0.22	0.29

Fig 4.10 Accuracy results for all models for different accuracy threshold.

After obtaining the results from Lstm ,now feature set is converted into single feature vector.This single feature vector is given to different models (NP,Weibull,exponential and Rayleigh)and equation for all the above model is described in Table 4.2

Table 4.2 The density, survival, intensity, and cumulative intensity functions

Distribution	**Density function** $f_T(t)$	**survival function** $s(t)$	**Intensity function** $\lambda(t)$	**Cumulative Intensity** $\wedge(t)$
Exponential	$\alpha \exp(-\alpha t)$	$\exp(-\alpha t)$	α	αt
Rayleigh	$\frac{t}{\alpha^2}\exp\left(-\frac{t^2}{2\alpha^2}\right)$	$\exp\left(-\frac{t^2}{2\alpha^2}\right)$	$\frac{t}{\sigma^2}$	$\frac{t^2}{2\sigma^2}$
Gompertz	$\alpha e^t \exp\left\{-\alpha\left(e^t-1\right)\right\}$	$\exp\left\{-\alpha\left(e^t-1\right)\right\}$	αe^t	αe^t
weibull	$\frac{\alpha t^{\alpha-1}}{\beta^\alpha}\exp\left\{-\left(\frac{t}{\beta}\right)^\alpha\right\}$	$\exp\left\{-\left(\frac{t}{\beta}\right)^\alpha\right\}$	$\frac{\alpha t^{\alpha-1}}{\beta^\alpha}$	$\left(\frac{t}{\beta}\right)^\alpha$

Results obtained from each model for both static and dynamic data is depicted in Fig 4.10 and Test is conducted for both static set of data and dynamic set of data,in comparison with all models Np-glm gives higher accuracy.

4.3 TEST CASE

Table 4.3 Tabulation for Test Case

Test Case Name	**Test Case Description**	**Input**	**Expected Output**	**Actual Output**
Splitting into Intervals	the first part for extracting the feature x, and the second for determining the target variable t.	Only heterogeneou-s graph need to be given.	Intervals are splitter into observed range and censored range.	As expected
Feature extraction	Extract the author details from feature extraction widow and observation window.	Only heterogeneous graph need to be given.	observed and censored samples along with write, cite, publish and include.	As expected
Metapath based feature extraction	To extract features for continuous-time relationship prediction	write, cite, publish and include values are given.	Multivariate X,Y,T feature are obtained.	As expected
LSTM	Multivariate feature are converted into single feature.	Multivariate X,Y,T.	conversation into single feature vector.	As expected
Supervised models	Feature extracted from metapath are given to fit into different models.	X,Y,T feature.	Accuracy for different models are obtained.	As expected

4.4 PERFORMANCE MEASURE:

This section evaluates the performance of the proposed system based on precision,recall and accuracy. Implementation were performed using several parameter settings for the models for the same datasets.

- **Mean Absolute Error (MAE):** This metric measures the expected absolute error between the predicted time values and the ground truth:

$$MAE(t,\hat{t}) = \frac{1}{N}\sum_{i=1}^{N}\left|t_i - \hat{t}_i\right| \qquad (4.16)$$

- **Mean Relative Error(MRE):**This metric calculates the expected relative absolute error between the predicted time values and the ground truth:

$$MRE(t,\hat{t}) = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{t_i - \hat{t}_i}{t_i}\right| \qquad (4.17)$$

- **Root Mean Squared Error (RMSE):** This metric computes the root of the expected squared error between the predicted time values and the ground truth:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - \hat{t}_i)} \qquad (4.18)$$

- **Mean Squared Logarithmic Error (MSLE):** This measures the expected value of the squared logarithmic error between the predicted time values and the ground truth:

$$MSLE(t,\hat{t}) = \frac{1}{N}\sum_{i=1}^{N}\left(\log(1+t_i) - \log(1+\hat{t_i})\right)^2 \qquad (4.19)$$

- **Median Absolute Error (MDAE):** It is the median of the absolute errors between the predicted time values and the ground truth:

$$MDAE(t,\hat{t}) = median\left(\left|t_1 - \hat{t}_1\right| \ldots \left|t_N - \hat{t}_N\right|\right) \qquad (4.20)$$

- **Maximum Threshold Prediction Accuracy (ACC):** This measures for what fraction of samples, a model have a lower absolute error than a given threshold:

$$ACC(t,\hat{t}) = \frac{1}{N}\sum_{i=1}^{N} 1\left(\left|t_i - \hat{t}_i\right| < threshold\right) \qquad (4.21)$$

```
--- 37.32112526893616 seconds ---
                MAE    MRE    RMSE    MSLE    MDAE    CI
------------  -----  -----  ------  ------  ------  ----
NP-Glm         2.25   1.07    2.63    0.37    2.16  0.50
Wbl-Glm        2.77   1.32    3.46    0.53    2.27  0.58
Exp-Glm        4.93   2.45    9.50    0.96    3.18  0.52
Ray-Glm        4.58   2.16    5.67    0.87    3.91  0.45
NP-Glm_stat    2.04   0.98    2.48    0.30    1.67  0.53
Wbl-Glm_stat   2.50   1.19    2.92    0.39    2.43  0.58
Exp-Glm_stat   3.22   1.48    3.78    0.53    2.96  0.49
Ray-Glm_stat   4.28   1.94    4.83    0.72    4.39  0.29
```

Fig 4.11 All the error values are calculated for static and dynamic data.

After obtaining accuracy from all the models depicted in Figure 4.8,error values are calculated between predicted and the ground truth values are described in Figure 4.9 for both static and dynamic data,compared to all the models ,NP-Glm gives the lowest error value.

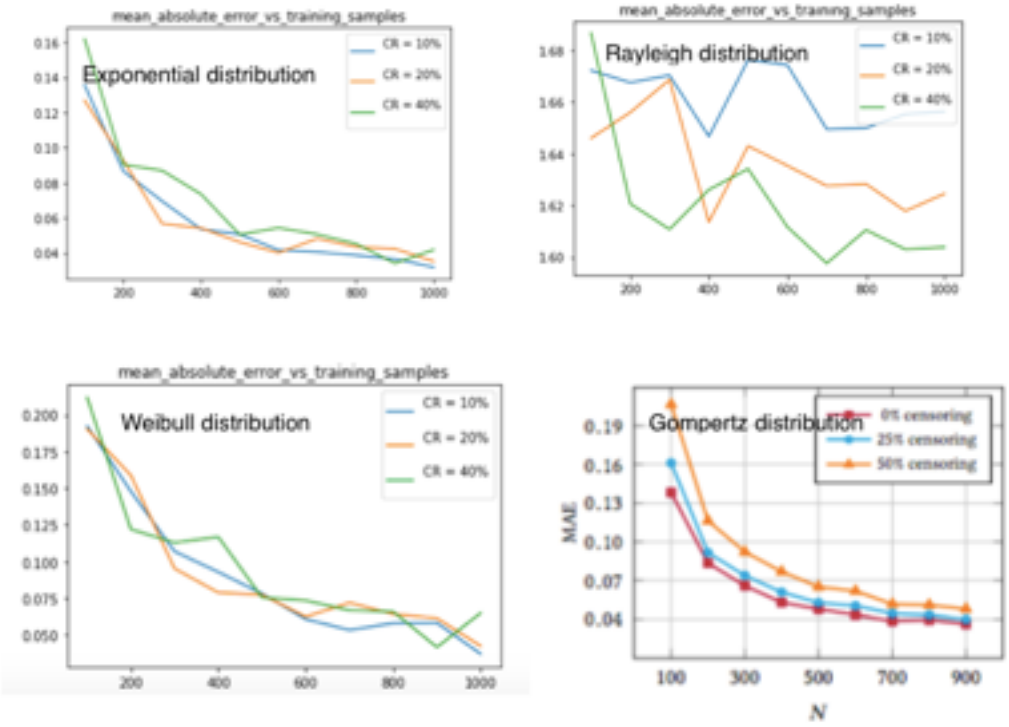

Fig 4.12 Np-Glm's mean absolute error (MAE) vs the number of training samples (N) for different censoring ratios.

MAE, MRE, RMSE, MSLE, MDAE, and CI of all models using both dynamic and static feature sets has been shown in Figure 4.10.In all three networks, Np-Glm with the LSTM Auto encoder feature set is superior to the other methods under all performance measures. For instance, our model Np-Glm can obtain an MAE of 1.99

for dataset, which is 15% lower than the MAE obtained by its closest competitor, Wbl-Glm. As of CI, Np-Glm achieves 0.62 on DBLP, which is 7% better than Wbl-Glm. On Delicious dataset, Np-Glm improves MAE and CI by 11% and 23%, respectively, relative to Wbl-Glm. Similarly, Np-Glm reduces MAE by 19% and increases CI by 25%. Comparable results hold for other performance measures as well. Accordingly, Wbl-Glm, which has two degrees of freedom, has shown a better performance compared to the other two " Fixed-shape" models, namely Exp-Glm and Ray-Glm. That is while Np-Glm, as a non-parametric model with highly tunable shape, outperforms all the other "less-flexible" models by learning the true distribution of the data. Using the dynamic features learned with the LSTM autoencoder has boosted the performance of all models over different datasets, and has outperformed the other feature extractors.The proposed feature extractor has achieved 7% less MAE and 17% more CI with Np-Glm. Np-Glm with LSTM-based features reduces MAE by about 7% and improves CI by 7%.

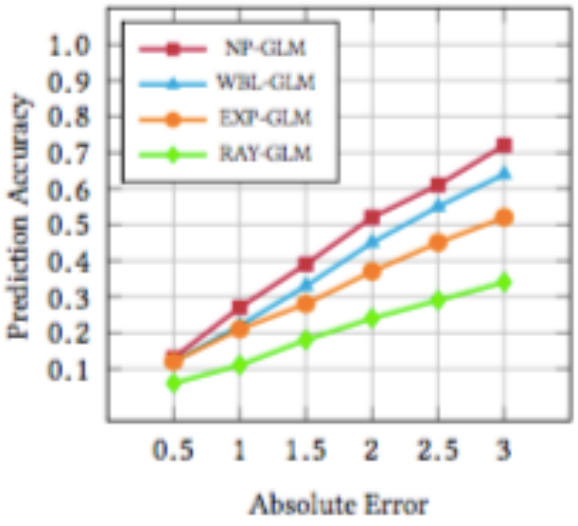

Fig 4.13 Prediction accuracy of different methods vs the maximum tolerated absolute error.

Investigated the performance of different models using the LSTM autoencoder feature extraction framework under maximum threshold prediction accuracy. To evaluate the prediction accuracy of a model, we record the fraction of test samples for which the di erence between their true times and predicted ones are lower than a given threshold, called tolerated error. The results are plotted in Figure 4.12 where we varied the tolerated error in the range {0.5, 1.0, . . . , 3.0}. We can see from the gure that Np-Glm and Wbl-Glm perform comparably, yet Np-Glm outperforms Wbl-Glm in all cases. Np-Glm with 3 months of tolerated error achieves around 80% accuracy, which is about 12% more than Wbl-Glm.

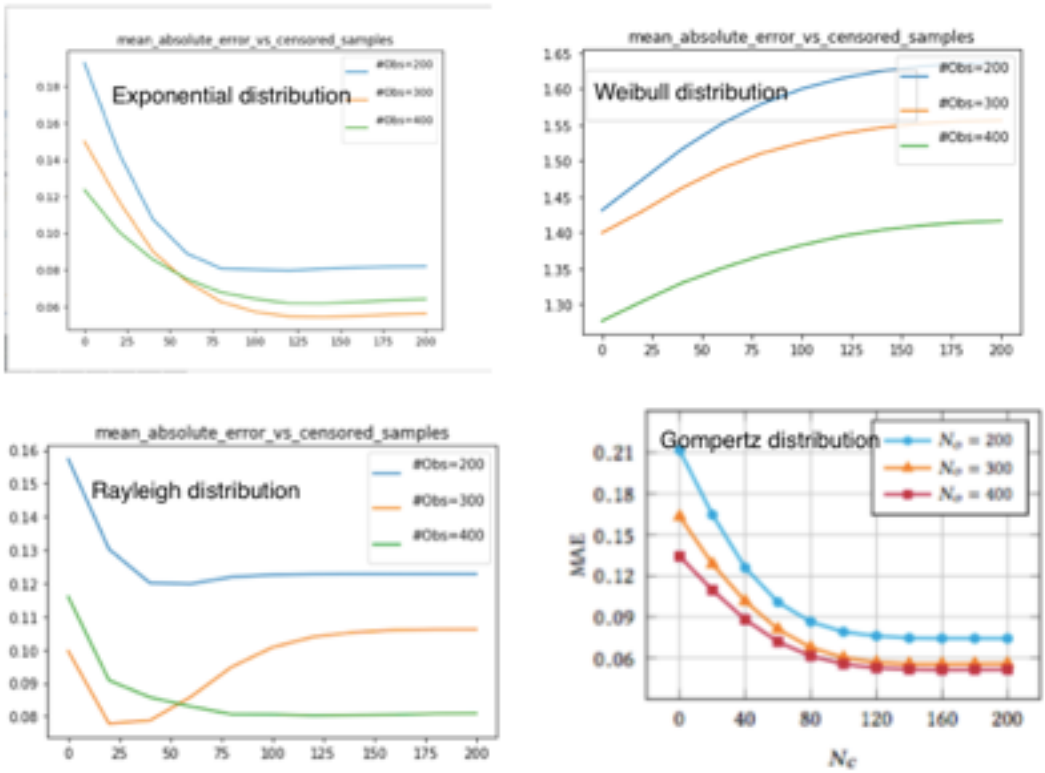

Fig 4.14 Np-Glm's mean absolute error (MAE) vs the number of censored samples (N_c) for different number of observed samples (N_o).

Investigated whether censored samples are informative or not. For this purpose, fixed the number of observed samples No and changed the number of censored samples from 0 to 200.Measure the MAE between $\hat{w}$ and the ground truth for $No \in \{200, 300, 400\}$. The result is shown in Figure 4.13. It clearly demonstrates that adding more of censored samples causes the MAE to dwindle up to an extent, after which get no substantial improvement. This threshold is depended on the underlying distribution.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In this project, The problem of continuous-time relationship prediction in both dynamic and heterogeneous information networks. To tackle this problem,Introduced a novel feature extraction framework based on meta-path modelling and recurrent neural network auto encoders to systematically extract features that take both the temporal dynamics and heterogeneous characteristics of the network into account for solving the continuous-time relationship problem. Then proposed a supervised non-parametric model, called Np-Glm, which exploits the extracted features to predict the relationship building time in information networks. The strength of our model is that it does not impose any significant assumptions on the underlying distribution of the relationship building time given its features, but tries to infer it from the data via a non-parametric approach. Extensive experiments conducted on a synthetic dataset demonstrated the correctness of our method and its effectiveness in predicting the relationship building time.

5.2 FUTURE WORK

Predicting relationship building time is still an open problem. It is worth of considering a broader range of target relations in different heterogeneous networks, and propose a systematic methodology of model selection and model comparison. More future work can be done along this line.

REFERENCES

[1] Niladri Sett, Saptarshi Basu, Sukumar Nandi, and Sanasam Ranbir Singh(2017). Temporal link prediction in multi-relational network. WorldWide Web, Vol. 20, No. 6, pp. 1–25.

[2] L. Lu¨ and T. Zhou(2017), "Link prediction in complex networks: A survey," Physical A: Statistical Mechanics and its Applications, Vol. 390, No.6, pp. 1150–1170.

[3] D. Liben-Nowell and J. Kleinberg(2010), "The link-prediction problem for social networks," journal of the Association for Information Science and Technology, Vol. 58, No.7, pp. 1019–1031.

[4] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu(2011), "Pathsim: Meta path- based top-k similarity search in heterogeneous information networks," Proceedings of the VLDB Endowment, Vol. 4, No.11, pp. 992–1003.

[5] Cham Aggarwal, Yan Xie, and Philip S Yu(2012).On dynamic link inference in heterogeneous networks. In Proceedings of the 2012 SIAM International Conference on Data Mining. SIAM, Vol. 16, No.3, pp. 415–426.

[6] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip(2017). A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering Vol. 29, No.2, pp. 17–37.

[7] Harald Steck, Balaji Krishnapuram, Cary Dehing-oberije, Philippe Lambin, and Vikas C Raykar(2008) On ranking in survival analysis: Bounds on the concordance index. In Advances in neural information processing systems. Vol. 39, No.5, pp. 1209–1216.

[8] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han(2011). Co-author relationship prediction in heterogeneous bibliographic networks. In Advances in Social Networks Analysis and Mining (ASONAM), International Conference on. IEEE, Vol. 29, No.8, pp. 121–128.

[9] BlissCA,FrankMR,DanforthCM,DoddsPS(2014)An evolutionary algorithm approach to link prediction in dynamic social networks. Journal of the Association for Information Science and Technology Vol. 58, No.7, pp. 750-764.

[10] Linyuan Lü and Tao Zhou.(2011). Link prediction in complex networks: A survey. Physical A: Statistical Mechanics and its Applications Vol. 390, No.6, pp. 1150–1170.

[11] DavidLiben-Nowell and JonKleinberg(2007).The link prediction problem for social networks.journal of the Association for Information Science and Technology Vol. 58, No. 2, pp. 1019–1031.