UNCOVERING NEW LINKS THROUGH INTERACTION DURATION

by

Laxmi Amulya Gundala

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

December 2017

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Laxmi Amulya Gundala

Thesis Title: Uncovering New Links Through Interaction Duration

Date of Final Oral Examination: 13 October 2017

The following individuals read and discussed the thesis submitted by student Laxmi Amulya Gundala, and they evaluated her presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Francesca Spezzano, Ph.D.	Chair, Supervisory Committee
Edoardo Serra, Ph.D.	Member, Supervisory Committee
Steven M. Cutchin, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Francesca Spezzano, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

DEDICATION

I dedicate the work of my thesis to my parents for always having faith in me and for motivating me in the hardest of the times. I would like to dedicate to my husband for standing by my side on every step of the journey to complete my Master's degree.

ACKNOWLEDGEMENTS

I want to acknowledge my advisor Dr. Francesca Spezzano for guiding me in completing my thesis effectively and also the Department of Computer Science at Boise State University for funding my master's program.

ABSTRACT

Link Prediction is the problem of inferring new relationships among nodes in a network that can occur in the near future. Classical approaches mainly consider neighborhood structure similarity when linking nodes. However, we may also want to take into account whether the two nodes we are going to link will benefit from that by having an active interaction over time. For instance, it is better to link two nodes □ and □ if we know that these two nodes will interact in the social network in the future, rather than suggesting □, who may never interact with □. Thus, the longer the interaction is estimated to last, i.e., persistent interactions, the higher the priority is for connecting the two nodes.

This current thesis focuses on the problem of predicting how long two nodes will interact in a network by identifying potential pairs of nodes (□, □) that are not connected, yet show some Indirect Interaction. “Indirect Interaction” means that there is a particular action involving both the nodes depending on the type of network. For example, in social networks such as Facebook, there are users that are not friends but interact with other user’s wall posts. On the Wikipedia hyperlink network, it happens when readers navigate from page □ to page □ through the search box (on the top right corner of page □), and there is no explicit link on page □ to □. This research explores cases that involved multiple interactions between □ and □ during an observational time interval [□$_{□}$, □$_{□}$). Two supervised learning approaches are proposed for the problem. Given a set of network-based predictors, the basic approach consists of learning a binary classifier to

predict whether or not an observed Indirect Interaction will last in the future. The second and more fine-grained approach consists of estimating how long the interaction will last by modeling the problem via Survival Analysis or as a Regression task. Once the duration is estimated, this information is leveraged for the Link Prediction task.

Experiments were performed on the longitudinal Facebook network and wall interactions dataset, and Wikipedia Clickstream dataset to test this approach of predicting the Duration of Interaction and Link Prediction. Based on the experiments conducted, this study's results show that the fine-grained approach performs the best with an AUROC of 85.4% on Facebook and 77% on Wikipedia for Link Prediction. Moreover, this approach beats a Link Prediction model that does not consider the Duration of Interaction and is based only on network properties, and that performs with an AUROC of 0.80 and 0.68 on Facebook and Wikipedia, respectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BSU	Boise State University
GC	Graduate College
OSN	Online Social Networks
RLTP	Reciprocal Link Prediction problem
AFT	Accelerated Failure Time
BJ	Buckley-James
Linear SVM	Linear Support Vector Machine
SVM	Support Vector Machine
RBF	Radial Basis Function
SVR	Support Vector Regression
AUC	Area under Curve
AUROC	Area under of ROC
TD-AUC	Time Dependent AUC
KNN	K-nearest Neighbors
CTR	Click-Through rate

CHAPTER 1: INTRODUCTION

Online social networks (OSN) have become popular among various age cohorts. People use them for not only socializing but also to gain insights on different day-to-day aspects, such as educational information, following the latest gossip, or interacting with peers around the clock. Some of these interactions can be Direct or Indirect. *Direct Interaction* is when a person/node exchanges information directly either through messages, emails, or calls, while the *Indirect Interaction* can be in many ways. People can have a third-party moderator to pass the word to connected family members and friends, or it can be strangers following up on a group conversation, and they help shape the social networks by creating new connections. While text messages and calls are traditional ways to interact, there are various forms of interactions on the Internet like Facebook's wall posts, comments, likes, and shares, or Twitter's tweets and re-tweets.

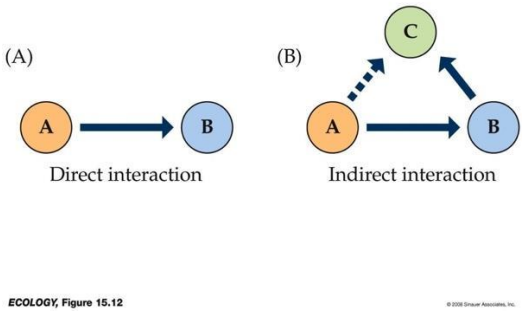


Figure 1. Indirect Interaction

By Indirect Interaction between nodes □ and □, there is a particular action depending on the type of network under study that involves both u and v (multiple times) during a given time interval [□□, □□). This study's interest is in the Indirect Interactions between nodes that are not connected. Examples of Indirect Interactions are:

a) On social networks such as Facebook where users can interact with wall posts, comments, group conversations and information-sharing with users that are not on their friends' list.

b) On Twitter, users can re-tweet or reply to tweets written by users who are not in their connections.

c) On the Wikipedia hyperlink network, readers can navigate from page □ to page □ through the search box (on the top right corner of page □) in case there is no explicit link on page □ to □. Some of these searches are casual and occasional, some last for a while because of current trending associations of topics, while others suggest the demand of a physical link from page □ to page □.

d) On the Amazon co-purchased products network, we can discover future co-purchased products by looking at users' search logs that may suggest examples of product recommendations: people who purchased (or searched for) product □ may also be interested in product □. Moreover, when considering a pair of products, if there are a relatively higher number of users purchasing those two products, then it will be helpful to allocate them in some warehouses.

e) On consumer review websites such as Yelp, people can write and read reviews on various products or businesses. As social networking websites, people make

connections with others and share information. We can identify Indirect Interactions in such networks based on their common reviewed products and predict users' future connections (friendships).

Indirect Interactions can be categorized as Node-Dependent and Node-Independent. Node-Dependent Interactions are those interactions that the nodes in the network are responsible for. For instance, on Facebook or Twitter, people create user profiles and use them to connect and communicate with other people. Node-Independent Interactions are between nodes in the network that happen because of external entities that use the network. For instance, on Wikipedia, pages are the nodes in the network and hyperlinks are the edges. People use these hyperlinks to navigate from one page to another. Amazon is another example of a network with Node- Independent Interactions, while Yelp is an example of a network with Node-Dependent Interactions.

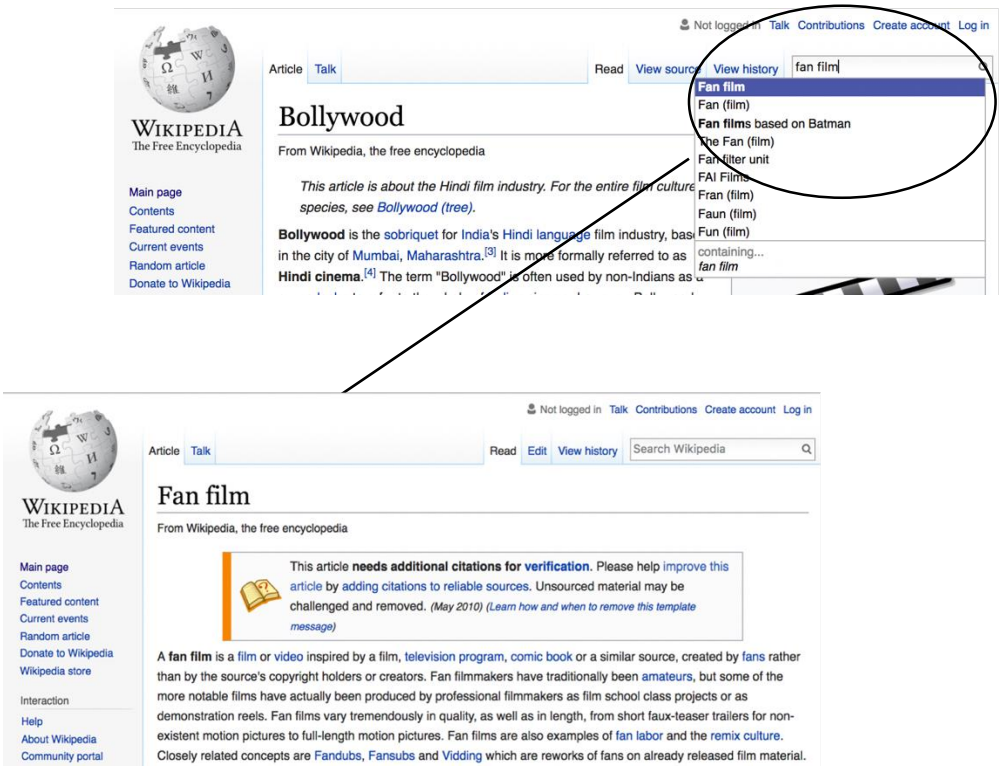


Figure 2. Indirect Interactions on Wikipedia. Source: https://en.wikipedia.org/wiki/Bollywood

The Indirect Interaction between nodes and during the time interval $[□_□, □_□)$ is an indication that they may have something in common and is convenient to link them. However, not all Indirect Interactions are useful. Casual Indirect Interactions are irregular and may be "one-time" interactions among different nodes. For instance, on Wikipedia, during February 2016, when Donald Trump was nominated as Republican nominee, users navigated from his Wikipedia page to various other Wikipedia pages like Trump University, Hollywood Walk of Fame, and Hillary Clinton. Though the number of interactions between those pages was in the thousands, it was a casual interaction as it did not continue after that period.

Persistent Indirect Interactions are indirect interactions that are continuous in a given time interval with interactivity always greater than a minimum threshold irrespective of the presence of an edge between them.

Persistent Indirect Interactions: Let $(□, □)$ be a pair of nodes having an Indirect Interaction during the time interval $[□_□, □_□)$. This Indirect Interaction is persistent during the time interval $[□_□, □_□]$, if for each time $□ \in [□_□, □_□]$, the number of Indirect Interactions between nodes $□$ and $□$ at that time $□$ is always greater than or equal to a threshold $□$.

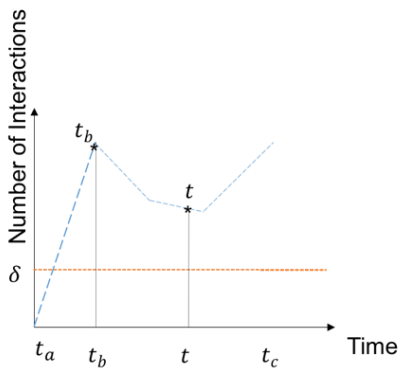


Figure 3. Persistent Indirect Interactions.

Persistent Interactions are the interactions that continue irrespective of an edge between them. Predicting connections can also be termed as Link Prediction but is different in a few aspects. Classical Link Prediction may or may not follow the interactivity between the pages, but it predicts future interactivity based on similarity. The scenarios below explain the Link Prediction on Wikipedia.

On Wikipedia, there were Indirect Interactions between two pages *Doctor Strange* (film), and *Baron Mordo* in February 2016 and users continued to navigate between these two pages even until April 2016 with a threshold always higher than 10. Later on, in April 2016, an edge was created from *Doctor Strange* (film) to *Baron Mordo*.

Wikipedia is a vast network of new users and new links are added daily [14]. For such a network, editors manually check which of the Wikipedia pages need to have hyperlinks between them based on page content and thus add those new links to those sites. However, these hyperlinks are subject to change in the future. For example, there are links that are deleted after just two days of creation while some remain active. While some links are still unchanged, some are changed every day (like the Main_page [2] of Wikipedia where its content is updated every day). Oftentimes, there are a few links that exist for a long time, and they might never be used. Some of the existing statistics were given in [12] stating that out of 800,000 links added to the site in February, 66% of them were not even clicked or used once.

Also, even if the editor chooses particular hyperlinks, there is no guarantee that the users will find it useful, or click on that link to navigate between the pages. For example, there was an internet viral sticker ‘Trash Dove’ during February 2017. It was an

ugly purple dove picture created as a sticker on Facebook. A Wikipedia page for 'Trash Dove' was created on February 16, 2017. Up until February 19, 2017, this Wikipedia page had a link to 'Anthropology' but was later removed. The 'Trash Dove' might belong to a cultural anthropology category as being relevant. It was once stated by Thailand's newspaper Khao Sod as 'A Cultural Joke' but there is no valid explanation to say that users navigate to 'Anthropology' from 'Trash Dove.' For an internet meme, it can be irrelevant and hence may have been removed. So, even though editors might choose the hyperlinks for a particular time, it might not be used or clicked as much as the relevance suggests. Also, there is no specific notion of the creation of hyperlinks that can be considered valid for a long time. All the hyperlinks created have a single purpose, to be useful for internet users to navigate between pages. Also, considering the number of hit counts from one page to another does not add weight to the probability of creation of the link between those two pages. The best example to support this statement is Wikipedia's main page, which is a recursively changing article that has new content modified and added to it daily. This page's sections are updated every day. However, when we considered the Clickstream dataset [3] from February 2016, there were hundreds of hits from this page or to this page from a random article. It was mostly because users might have navigated to Wikipedia's main page first and then to their topic of interest. A higher number of hits between a pair of pages does not necessarily prove that there has to be a hyperlink created between these pages.

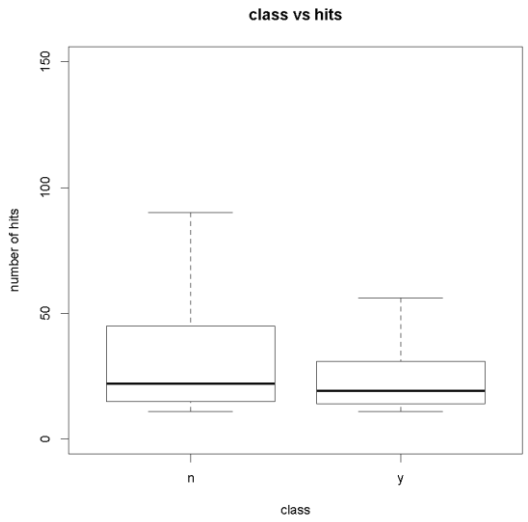


Figure 4. Box plot of hits concerning classes *y* and *n*.

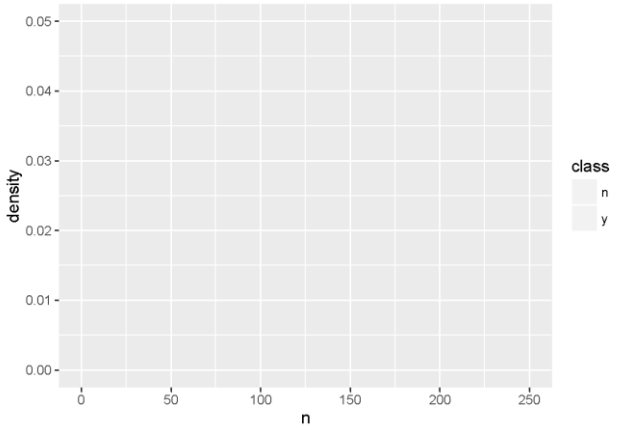


Figure 5. Histogram of hits density concerning classes *y* and *n*.

The experimental results above also prove the same where class *y* define the existence of interactivity and class *n* define non-existence of interactivity.

On social networking sites like Facebook, Twitter, or Instagram, users are often provided with suggestions to connect, which is most likely based on their network. For example, on Facebook, user □ will get friendship suggestions from user □ if user □: a) happens to be new to Facebook and has a mutual friend with user □; b) belong to the same sub-network as user □(workplace, same neighborhood, education at same school) or has a relatively higher number of mutual friends. Similar suggestions are also given on

other social networking sites like Twitter and Instagram. For all these suggestions, the only assumption is that they may know each other or may become future acquaintances. However, there is no guarantee that these users will have persistent interactions or no interaction at all. In such cases, these connections may not be useful.

In this thesis, when predicting links between a pair of nodes, priority is given to those links that will be useful in the future. On any social networking platform, it is better to recommend user □ to become friends with user □ if we know that these two nodes will interact in the future, rather than suggesting □ as a friend for □ even after we know that □ and □ will never interact. Thus, the longer the interaction is estimated to last between a pair of nodes; the higher is the priority for recommending a link between them. Predicting connections in this study is different from the classical Link Prediction in the following ways:

I. Link Prediction aspects to predict links mostly by analyzing semantic similarities among the nodes. Whereas, the current study focuses on predicting connections based on Persistent Indirect Interactions.

II. Link Prediction focuses on growing the network by suggesting missing edges. However, it can sometimes lead to over-crowding by unused edges. This study focuses on predicting only such connections that are most likely to be useful in the future.

The motto of edges/links/connections on the Internet is to help the user with better navigation and interaction. However, if the suggested connections do not suffice the requirement, there is no point in clouding the network with more less-used edges. Identifying the potential connections can be tricky in the way that hit counts on

Wikipedia do not explain. Hence, the problem statement and the approach is to identify such Persistent Indirect Interactions and predict connections.

Two supervised learning approaches are proposed for the problem of predicting the Duration of Interactions. Given a set of network-based predictors, the basic approach consists of learning a binary classifier to predict whether or not an observed Indirect Interaction will last in the future. The second and more fine-grained approach consists of estimating how long the interaction will last by modeling the problem via Survival Analysis or as a Regression task. Once the duration is estimated, this information is leveraged for the Link Prediction problem.

An extensive experimental evaluation was performed with two longitudinal datasets, namely Facebook network and wall interactions, and Wikipedia Clickstream. This approach was tested to predict the Duration of Indirect Interaction and its application to the Link Prediction task. Based on all the experiments, the results show that the more fine-grained approach (Survival Analysis on Facebook and Regression model on Wikipedia) has maximum improvement for predicting the Duration of Indirect Interactions and achieved an AUROC of 0.85 for Facebook and 0.77 on Wikipedia for Link Prediction. Moreover, this approach beats a Link Prediction model that does not consider the duration of interaction, is based only on network properties, and performs with an AUROC of 0.80 and 0.68 on Facebook and Wikipedia, respectively.

The remainder of this study is organized as follows. Chapter two details the related work. Chapter three explains the datasets and how ground truth was estimated on those datasets. Chapter four discusses the methods used. Chapter five provides details on the proposed approach. Chapter six discusses the various predictors used. Chapter seven

is about all the experiments we performed, and Chapter eight concludes with recommendations for future work.

CHAPTER 2: RELATED WORK

2.1 Link Prediction

Link Prediction is a problem of predicting connections between two nodes in a network. This problem can be applied to different types of networks [21,22,23,24,25,53]. For comparatively small networks, it is possible to determine the links and add them to the network manually. However, due to the complexity and size of social networks, it is important to automate the process to reduce human intervention. Some of the notable types of approaches to tackle the Link Prediction problem are discussed next.

Liben-Nowell and Kleinberg [6] point out that social networks are highly dynamic objects, new edges are added to the network along with the removal of some old (or) unused edges, from time to time. Their proposed work for the Link Prediction problem uses a graph structure on five co-authorship networks available from the physics e-Print arXiv, www.arxiv.org. Some of the features introduced are the graph's distance-based Common Neighbors, Jaccard's coefficient, and Adamic/Adar features. One of the main problems of having these features on their dataset is that the pages of similar categories might have more neighbors in common, and hence have more leniency over predicting a hyperlink between those pages. While pages with a different set of neighbors belong to different categories, but are somehow related, might not have the same probabilities as previous pages. Similar work has been conducted by Hasan et al. [16] where a dataset of authors and their papers [28] was chosen. They attempted to predict which set of authors are most likely to publish a technical paper together in the near

future. Though the size of the dataset was small and it was during that time when the Link Prediction Problem was not yet addressed directly, they were able to explore the possible network and graph-based features. Their insights to those features are still used as baselines for most of the recent works. They address topological features like Shortest Distance, Clustering Index, some other Proximity features based on keyword match count, and Aggregated features like sum of neighbors, and the count of commonly published papers. Also, they give insight into the nature of different classifiers' performances where SVM tend to have a good understanding and prediction over other classifiers.

Considering a network as a graph with nodes and edges, Grover, A., & Leskovec, J. [15] created an algorithm, 'Node2Vec', on multi-label classification and Link Prediction problem that can be used on various real-world networks like Facebook or Protein-protein Interactions. It is a semi-supervised algorithm for scalable feature learning in networks. It is an optimized graph-based objective function to preserve neighborhood using random walks. Using the information from their algorithm, it is possible to understand and predict most probable labels of nodes in a network that are useful for Link Prediction. Depending on varying parameters in using Node2Vec (like the number of walks per node, context size, or the fraction of missing edges), it is possible to estimate node and edge features for any network in any domain. This algorithm can even be applied to an incomplete network with missing edges.

2.1.a Link Prediction on Wikipedia

On Wikipedia, plenty of work has been done on the Link Prediction problem which is close to the idea of predicting connections between Indirect Interactions.

However, Link Prediction does not entirely focus on Indirect Interactions; it involves other factors that can be helpful to predict future interactivity. There are many state-of-the-art works published based on the Link Prediction task. Some of the interesting related works on Wikipedia are discussed below.

Adafre et al. [5] proposed an approach to find missing links in the network by considering Wikipedia corpus and its underlying abstract words. They clustered all the Wikipedia pages to rank them by using LTRank and Lucene algorithms on the existing links to find similar pages. They stated that similar pages should have similar hyperlinks. They extracted anchor text from their related pages and predicted missing out-links in those similar pages. They then evaluated those missing links manually. Similar work was done by Noraset et al. [9] by considering text from Wikipedia pages. They proposed '3W' to use semantic information of those pages and identify words/concepts to determine links to their referent pages.

Another study was conducted by West et al. [10] where they used human navigation logs available from The Wiki Game [30] (a collection of five different challenges like least clicks, speed race, five clicks [or fewer] to Jesus, no United States and six degrees of Wikipedia) and Wikispeedia [31,32] to identify missing links on Wikipedia. The Wiki Game challenges refer to various ways of reaching a page *t* by starting at a page *s* and using only hyperlinks on page *s* to navigate. Wikispeedia is a similar game to reach a random page *t* from a random page *s* with minimum path length. The user needs to click only on page links/hyperlinks on a reduced and static snapshot of Wikipedia. West et al. rated the source candidates based on relatedness by using the Milne Witten measure [8] and path frequency using singular value-decomposition to

obtain link candidates for the path. Based on those ranks, they were able to predict the top K pages for Link Prediction. Their evaluation was based on human raters available from Amazon Mechanical Turk (a platform where researchers post forms containing questionnaire and participants get paid for each form response they give). Another such work by West et al. [11] is based on dimensionality reduction where they created an adjacency matrix based on out-links from a page □ to page □. They used principal component analysis on that matrix to determine which of those two pages should be linked. Their system was also evaluated using human raters' responses on Amazon Mechanical Turk.

Paranjape et al. [12] constructed trees from the server logs of Wikipedia. These server logs consisted of information about each HTTP request from a user. The logs were grouped by user id, and most recent requests to a page were selected. On this available dataset, they used search proportion, path propagation, random walks and a combination of search and path propagation methods to identify potential link candidates. The number of page hits was the main component for three objective functions to list top K pairs of pages to be considered for a link between them. They tested their unsupervised results over editors' choice of newly added links in the following month. Their results showed that most of the pairs predicted matched the editors' choice of hyperlinks. This work is similar to the approach in this current thesis but is different in the following aspects:

I. The dataset for West et al. is the server logs obtained from Wikipedia and by constructing heuristic trees to identify the potential link candidates. The dataset for this current work is provided by Wikipedia Clickstream consisting of counts

of pairs from request logs. However, the pairs with counts less than ten are not included.

II. Their approach was to identify the top K pairs like (*s*, *t*) to place a link based on click-through rate which is the measure of times that users click on *t* given that they are in *s*. This current study's approach focuses on determining Persistent Indirect Interactions and suggest links based on users' usage. Also, it does not solely rely on hit count but also on various other features as it was experimentally proven that hit counts do not effectively address a solution to the current problem.

III. They used Search proportion, Path proportion, and Random walks to identify potential pairs. This current study's approach focuses only on the Search-based proportion, i.e. *other* pairs (see section 3.1) to determine Indirect Interactions.

IV. Their work was validated with editors' choice of links in the following months. This current study's approach is validated against the users' choice of Persistent Interactions irrespective of a link.

2.2 Strength of Relationship

Link Strength Prediction is a problem close to Link Prediction and is defined as given an existing link between two nodes, predict the weight or strength of that link. While some works focus on predicting the number of interactions between two linked nodes [19, 38], others attempt to predict the type of the relationship (i.e., weak or strong tie [37,39,49,51,52,55], or degree of likes/dislikes [20, 54, 56, 67]). For any connection, it is important to have good relationship strength as it determines how often the nodes in that connection will interact or how important their connection is in the network. Whereas Link Prediction determines which nodes should have a connection between

them because of the similarities among them.

There have been some interesting works published on estimating the strength of a relationship [19,35,36,37,38] in social networks like Twitter, Facebook, and Orkut. While some of the works focused on interaction, others focused on the connected network to identify string ties. Kumar et al.'s [20] study emphasizes on predicting edge weights to demonstrate the strength of their relationship. Zignani et al.'s [19] study was conducted to predict the strength of new links on the Facebook dataset. They re-used a dataset from [29] and tried to predict the strength of newly connected Facebook users. Their approach was to identify the strength of connection at the time of creation without the knowledge of prior interactions. They used temporal features to understand the interactivity.

Wilson et al. [36] addressed the issue of whether all the connections/links are valid indicators of real interactions among the users in a social network by performing experiments on 10 million crawled Facebook user profiles. They observed that the interactivity on Facebook skewed towards a smaller portion of users' friendship networks raising doubt as to whether or not all links imply equal friendship relationships. They also suggested that applications in social networks should consider interaction activity rather than mere connections.

Kahanda et al.'s [37] experimental findings indicate that it is necessary to consider transactional events such as file sharing, wall posts, photograph tags, and messages as they are very useful for predicting link strength among the users in the social network. They also stated that while considering friendship, wall, picture and group attributes for the Facebook dataset, wall interactions had an utmost impact on their model's performance. Kamath et al.'s [38] study aimed at predicting future interactions in

the Twitter network based on historical interactivity. Their approach is to estimate the relationship strength between users [49, 50, 54, 55] based on direct interactions. Their framework included various graph-based, user-based and interaction-based features to fit their model.

Upon considering all the above-stated works, it is evident that for any social network, it is necessary to consider the interactivity to understand and perform any type of prediction tasks accurately and thus validating this current study's approach.

2.3 Survival Analysis and Regression

Survival Analysis is a statistical measure to determine the probability that an event will occur. This current study estimates the duration of interaction between a pair of indirectly interacting nodes by using Survival Analysis to predict individual survival probabilities for an event (they will stop interacting), i.e., the probability that they will not stop interacting in the given period. Survival Analysis [40,57,64] (see section 4.2) is not only used in the medical domain to predict the probabilities for the occurrence of an event (i.e., chances or survival, estimated death probabilities), but also in generalizing an event and estimating its probability to occur. For example, Dave, V. S et al. [42] used Survival Analysis for the Reciprocal Link Prediction problem (RLTP). They used a cocktail Algorithm [40], and two other statistical survival methods Accelerated Failure Time (AFT) and Buckley-James (BJ) models along with Regression models like RidgeReg, LassoReg, FFNN, and SVR. They used various Epinion (a consumer review website), MC-Email and Enron datasets (emailing websites) to determine how long it would take to get a response to their requests. By using various survival models and regression models, they attempted to estimate the duration of a reciprocal response.

Rakesh et al. [62] and Li et al. [63] used Survival Analysis on a Crowdfunding projects list. Crowdfunding is a platform for people to seek donations for completion of a project. It is an open platform where people can donate to a project of their interest. They studied whether the goal for the crowdfunding project was met within the stipulated time or not. While Rakesh et al. [62] examined the duration of successful projects by using censored regression models, Li et al. [63] also included the failed projects by using various logistic distributions.

Student retention rate is one of the major problems for a university. After completing a semester, the rate of students who return to the same university to begin the next semester is called student retention rate. Student retention rate is important for a university to be ranked higher than other universities in a nation and also to secure government released funds. Survival Analysis was used on such data by Murtaugh et al. [59] and Ameri et al. [64] to estimate the time of event occurrence, i.e., whether a student will drop out or not and if so, when will they drop out.

The Internet provides us with many features. One such feature is advertisements. Using advertisements on a website attracts users to click on those links. User-clicking probability is the percentage of users who click on the ad with respect to the number of times the ad was displayed on the webpage. This is also called the click-through rate (CTR). Studies have been conducted to estimate the time it takes for a user to click [60] on the advertisement depending on the content of that website and displayed ads [61].

CHAPTER 3: DATASETS

This study used two datasets to test Wikipedia Clickstream and Facebook network with wall interactions. The former is an example of a Node-Independent interactions' network while the latter is an example of a Node-Dependent interactions' network. Both datasets are discussed next.

3.1 Wikipedia Clickstream

Wikipedia Clickstream is Wikimedia's research project in progress. It is a dataset consisting of pairs consisting of (referrer page, resource page) obtained from the extracted request logs of Wikipedia. There are eight months of datasets released to date, starting from January 2015. Each dataset consists of four fields (Source: [1]).

1. prev: the result of mapping the referrer URL (or) Page title if it is on Wikipedia.
2. curr: the title of the webpage the client requested (or) Page title if it is on Wikipedia.
3. type: describes (prev, curr)
    a. link: if the referrer and request are both Wikipedia pages and the referrer links to the request;
    b. external: if the referrer host is not en.wikipedia.org;
    c. other: if the referrer and request are both Wikipedia pages but the referrer does not link to the request. This can happen when clients search or spoof their referer.

4. n: the number of occurrences (greater than 10) of the (referrer, resource) pair. Considered as the number of hits from prev to curr.

Thus far, the following datasets have been released for the English version of Wikipedia:

a) January 2015: This dataset includes columns of page ids for prev and curr. Redirects were not resolved.

b) February 2015: This dataset includes columns of page ids for prev and curr. Redirects were not resolved.

c) February 2016: More granular set of fixed values for hit counts.

d) March 2016: More granular set of fixed values for hit counts.

e) April 2016: There are three language versions of this dataset—Arabic, English, and Farsi. More granular set of fixed values is given in this dataset.

f) August 2016, September 2016, January 2017: These are latest versions released.

For this study, February 2016, March 2016 and April 2016 are used as they are the longest consecutive months available in Clickstream.

We focused on the February 2016 dataset, and we build hyperlinks network by considering *links* from its *type* column. For nodes having Indirect Interactions, we considered the pairs having *type* as *others.* The *Other type* refers to a pair of pages from Wikipedia that do not have a direct link between them. This consists of a pair of pages that users tried to navigate through the search bar on the Wikipedia page. We considered these pairs as potential candidate pairs. To determine Persistent Indirect Interactions, it is essential to understand how long will they interact.

3.1.a Estimating ground truth about duration of interactions

Table 1 gives an estimate of the total number of pairs for each type in each month. The Main_page is a recursively changing web page on Wikipedia. Though there are pairs with a considerably higher number of hit counts, they are considered as noise in the dataset. Hence, all the datasets are filtered to remove any occurrence of the Main_page among the pairs.

Table 1: Wikipedia Clickstream Dataset Statistics of Row Count

	February	March	April	August
Total	27M	25M	21M	24M
Other	2.38M	2M	0.6M	0.67M
External	10M	10M	8.7M	9.2M
Link	14.62M	13M	12M	14M
Other (Except Main_page)	2.03M	1.7M	0.3M	0.37M

Of all the potential candidate pairs, we estimated how many pairs stopped interacting in March 2016 and then how many pairs continued interacting in April 2016. Thus the statistics of these estimates, irrespective of a link in later months, are detailed in Table 2. For the classification problem of understanding how many pairs had Persistent Indirect Interactions, we classified these pairs into *Positives* and *Negatives*. Of all the potential candidate pairs, we narrowed down 190,124 pairs that had interactions continuing until April 2016 while the rest (1,638,796) did not exhibit Persistent Interactions. Also, the duration of these interactions is used in the Survival Analysis approach to estimate how long they would interact.

Table 2: Statistics of Computed Datasets

February 2016 (Potential candidate pairs)	2.03M
Pairs that stopped interacting in February	536,380
Pairs that stopped interacting in March	1,102,416
Pairs that continued to interact in April	190,124
Positives	190,124
Negatives	1,638,796

3.2 Facebook

On Facebook, even though a pair of users may not be friends, they can still be part of a common activity. There are different types of interactions on Facebook through user's wall posts, messages, comments, shares, and likes. Facebook wall interactions are when a user posts something on a friend's timeline or vice versa that includes tagging. Interactions with the user's friends list are Direct Interactions and interactions with public Facebook users or common friends with another user are called Indirect Interactions. The typical Indirect Interactions on Facebook include:

a) A mutual friend tagging two or more unconnected Facebook users in a single post.

b) Commenting on a common friend's post.

c) Joining a common Facebook group and participating together by commenting on a post.

This idea can be studied using a dataset collected by Vishwanath et al. [29] which is available for public research. They crawled a New Orleans Facebook network and

obtained data from September 2006 to January 2009. All the nodes and their information are anonymized. This dataset consists of information about:

a) Friendship (user1, user2, friendship creation timestamp)

b) Wall interactions (user1, user2, posts' timestamp); where user2 is posting on user1's wall at a given timestamp.

Based on the availability of information in the dataset, only the interactions through wall posts are included in this current study.

3.2.a Estimating ground truth about duration of interactions

Since there is no evident information about the direction of friendship in the dataset, an undirected graph network is assumed. We considered half-yearly timestamps to construct six datasets on friendship and a similar six datasets on wall interactions (i.e., 2006b, 2007a, 2007b, 2008a, 2008b, 2009a where 'a' refers to the first six months in the year and 'b' refers to next six months in the year). To identify Indirect Interactions, we grouped the wall interactions in each dataset by timestamp and user. From the results of grouping, we formulated all possible pairs that were not connected yet, as the probable candidate pairs. We took the 2006b dataset as the starting time, and the candidate pairs in the dataset were considered as potential candidate pairs for this study. Table 3 shows the statistics on the row count in each dataset.

Table 3: Facebook Dataset Statistics of Row Counts

Timestamps/Datasets	Friendship	Number of users	Wall interactions
2006b	37,641	9,108	217,714
2007a	80,812	14,568	564,622
2007b	123,220	22,732	789,354
2008a	201,859	32,584	877,832
2008b	456,553	53,578	1,874,332
2009a	5,480	4,963	161,026

With the similar approach followed for Wikipedia, we considered all the potential candidate pairs and determined which of those pairs stopped interacting or continued interacting in later months. The statistics of these results are given in Table 4. We considered the end time as the 2008b dataset and used the 2009a dataset for evaluation purposes. Hence, positives in this dataset for the classification approach are 88,155, and the negatives are 4,155.

Table 4: Facebook Dataset Final Statistics for Wall Interactions

Potential candidate pairs in 2006b	175,577
Pairs that stopped interacting in 2006b	88,155
Pairs that stopped interacting in 2007a	52,858
Pairs that stopped interacting in 2007b	21,399
Pairs that stopped interacting in 2008a	8,738
Pairs that stopped interacting in 2008b	4,155
Pairs that continued to interact in 2009a	271

CHAPTER 4: METHODS

Two supervised learning approaches are proposed to identify candidate pairs for Link Prediction. Given a set of network-based predictors, the basic approach consists of learning a binary classifier to predict whether or not an observed Indirect Interaction will last in the future. The second and more fine-grained approach consists of estimating how long the interaction will last by modeling the problem via Survival Analysis or as a Regression task. An outline of these methods is presented below.

4.1 Classification

Classification in Machine Learning is the categorization of data into different classes. There are different approaches and algorithms on how to classify based on the type of datasets (for example, documents can be classified based on content similarity). In machine learning, there are two different types of classifications—binary classification and multi-class classifications. Binary classification is the problem of having only two classes generally named as $'0'$ or $'1'$. The class $'0'$ can also be defined as the classification of data into negatives (i.e., data does not belong to the desired class) and hence the class $'1'$ can be defined as a classification of data into positives (i.e., data belongs to the desired class). Multi-class classification consists of more than two classes for the data to be classified. There are two learning approaches for classification: supervised and unsupervised. The supervised learning model is the task of learning on a labeled training data and predicting class on a labeled test data. Training data is most of the dataset and test data is a smaller part of the dataset. There are various evaluation metrics to check the

predictability. The unsupervised learning model uses only unlabeled data to find underlying structures in the dataset. There are no metrics to evaluate the unsupervised learning models. Some of the supervised algorithms that were used for this thesis are:

4.1.a K-nearest Neighbors (KNN)

K-nearest neighbors (KNN) is a classification algorithm that uses distance or similarity for prediction. It places the tuples in the space to which they are closest to. Classification is done based on the majority vote of each nearest neighbors. For each class, there will be a query point that acts as a point of reference to calculate closeness. For the first iteration, a random data point is chosen as the query point and then iteratively calculates query point until no other changes are possible. Figure 6 shows the distribution of points in a dataset with three classes and fifteen neighbors. The 'weights' parameter defines the value assigned on each data point. By default, it is 'uniform' and assigns equal weights to all data points. With weights= 'distance,' the classifier assigns weights inversely to each point regarding its distance to the query point.

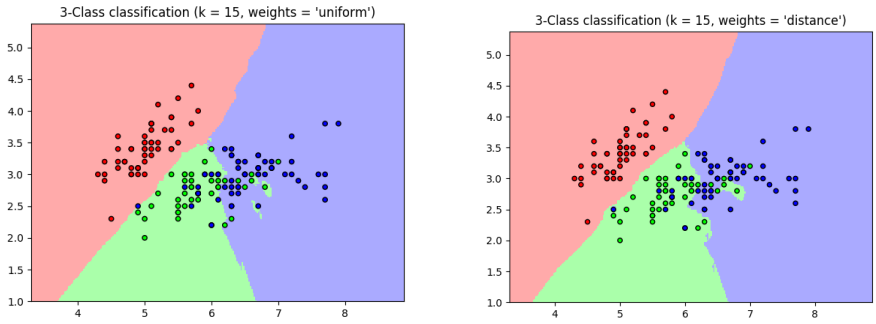


Figure 6. K-nearest Neighbor example. Source: http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html.

4.1.b Random Forests

Random Forests is a type of an Ensemble model. The Ensemble model is a combination of more than one model. For example, in Figure 7, Classifier1 creates a decision boundary 1 to separate three shapes: circle, triangle, and square. It is not accurate but works fine. Classifier 2 creates another decision boundary 2 and Classifier 3 creates decision boundary 3. An ensemble model formed by combining all the three classifiers gave an accurate decision boundary to separate shapes effectively.

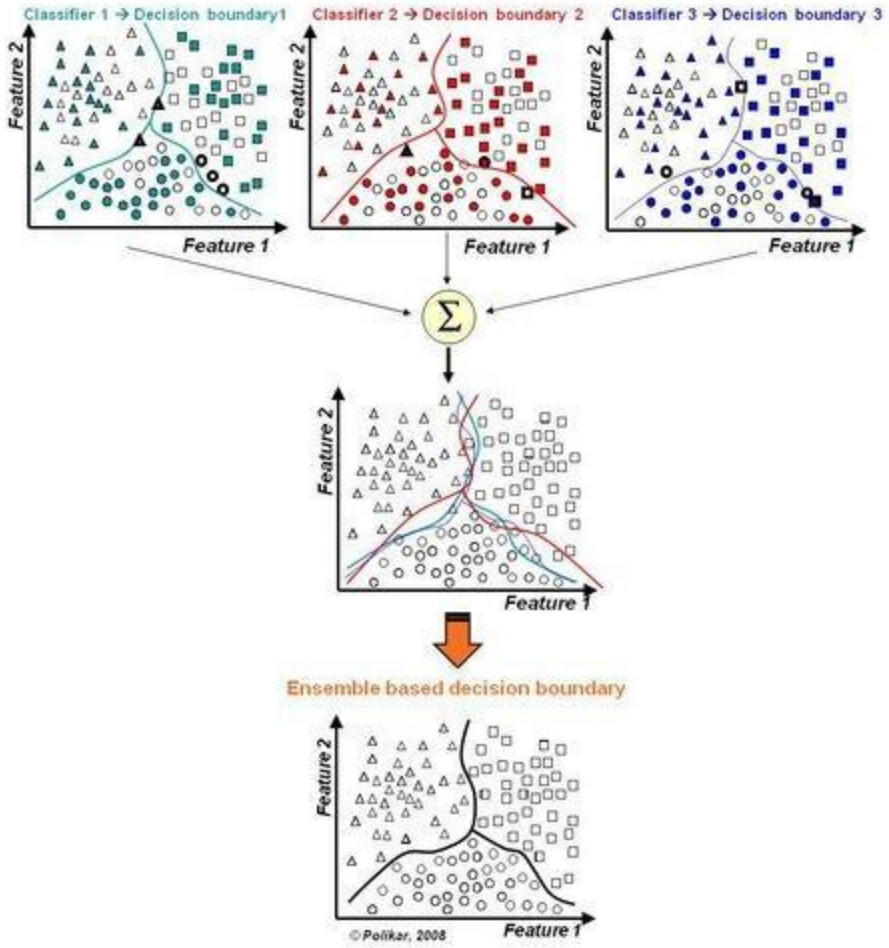


Figure 7. Pictorial Representation of how Ensemble works. Source: http://magizbox.com/training/machinelearning/site/ensemble/.

Random Forests is one such model that uses an ensemble of randomly generated trees on various subsets of the dataset. It uses averaging to improve accuracy and also controls the problem of over-fitting.

4.1.c Logistic Regression

Logistic Regression gives the probabilistic view of Regression [65]. Assuming a binary class, and as the vector of features for the classifier, then logistic regression finds the probability that the class belongs to class $'1'$ then the probability is given by

$$ = \frac{1}{^{-} + 1}$$

where is a vector of constants.

It is also a supervised learning model with different algorithms like the liblinear, newton-cg, or saga. This current study used the liblinear algorithm.

4.1.d Support Vector Machines

Support Vector Machines (SVM) is a supervised learning algorithm usually used for classification, regression and also detection of outliers task [66]. There are different kernels available using SVM. For a given training data, SVM determines a hyperplane that separates the examples to categorize into classes by using a sample of training points called support vectors in the decision function. Figure 8 shows the distribution of classes for the Iris flower dataset based on flower's sepal width and length.

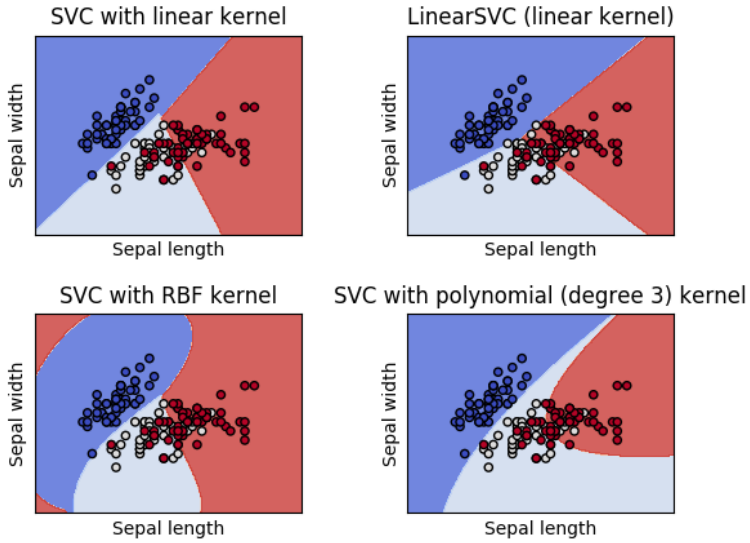


Figure 8. Pictorial representation of different kernels classification in SVM. Source: http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html.

4.2 Survival Analysis

Estimating the duration of an event to occur is one of the critical problems in analyzing data. It is possible that during any study on a group of entities, there might be instances for which the event did not occur within the study's duration or may have incomplete, missing, or unavailable data about the event occurrence. Such instances are called censored instances that can be effectively approached using Survival Analysis [40, 41, 55, 58, 59]. Survival Analysis uses hazard functions which determines the rate of occurrence of an event at a time *t* with the condition that the event did not happen before time *t*. The results of this function are applied to different statistical methods of Survival Analysis to estimate the final probabilities for each of the entities. The following descriptions explain the different types of methods [41]:

a) Non-parametric: Specific methods of this type are Kaplan-Meir, Nelson-Aalen, and Life-Table. These methods are mainly used when there no theoretical distribution of the event occurrences are known.

b) Semi-parametric: Specific methods of this type are the Cox model, Regularized Cox, CoxBoost and Time-Dependent Cox. These methods are mainly used where the knowledge about the distribution of survival times are not necessary.

c) Parametric: Specific methods of this type are Tobit, Buckley-James, Penalized Regression and Accelerated failure Time. These methods are mainly used when the patterns in survival times distribution are known.

d) Machine Learning methods: There are Survival trees, Bayesian methods, and Neural networks.

Figure 9 illustrates various Survival Analysis methods.

Various parametric ACT models and a Cox model were used in this current study. In the ACT models, the Weibull, LogNormal and Exponential methods were used.

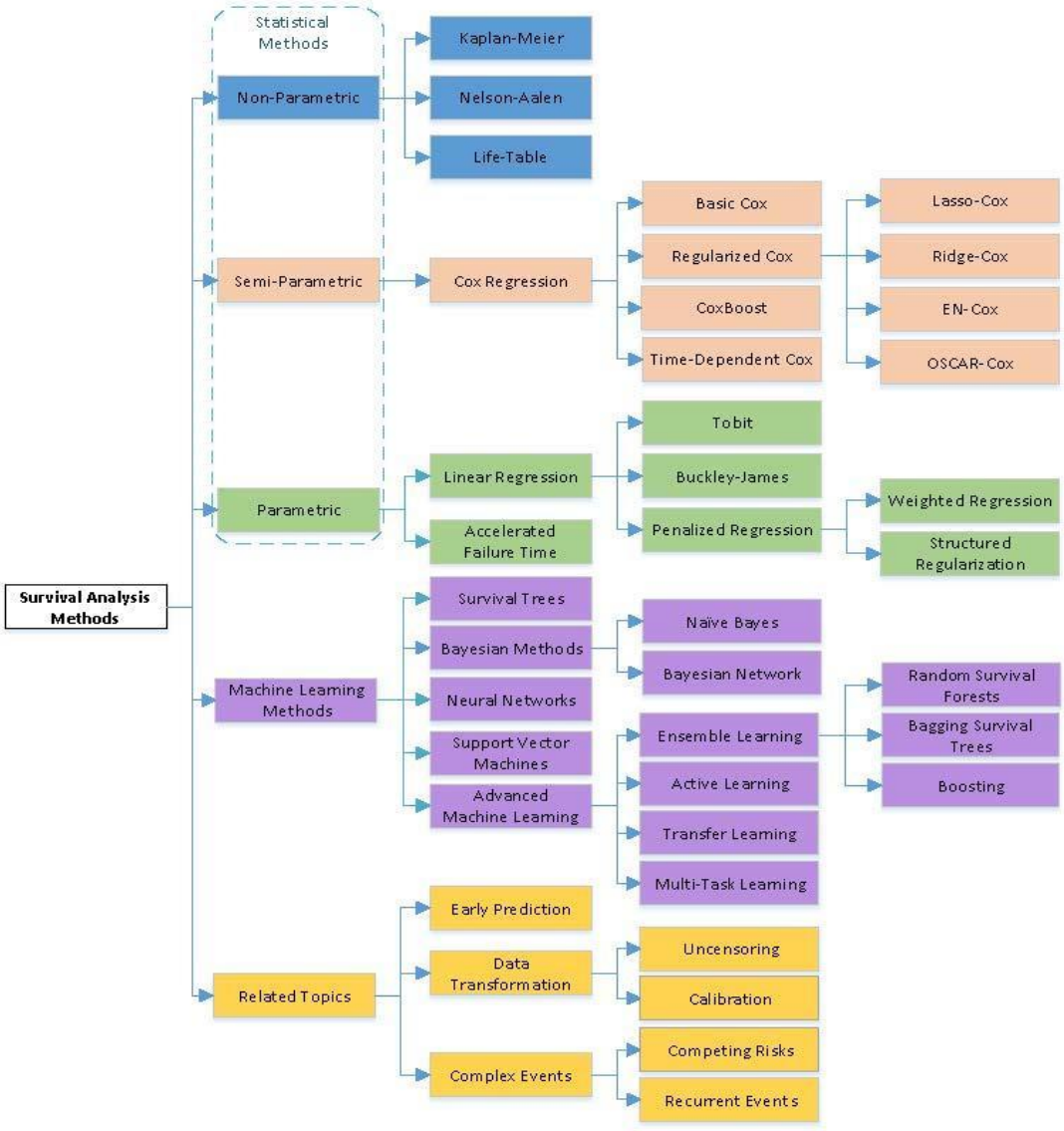


Figure 9. Taxonomy of methods developed for Survival Analysis. Image source: P. Wang et al. [41]

4.3 Regression

In statistical analysis, regression is a process of estimating the relationships among variables, such as a regression model for variable 𝑌 with respect for feature vector variable 𝑋 and a constant vector 𝛽. i.e.,

$$Y \approx f(X, \beta)$$

There are many regression models in statistics. This current study used Lasso, Ridge and Support Vector Regression with Radial Basis Function kernel. Ridge Regression is a technique for analysis multi-collinear data. Multi-collinearity occurs when the variance is so large that it is far away from the true value. Lasso Regression functions to improve the prediction accuracy of the statistical model.

CHAPTER 5: APPROACH

In this chapter, we propose approaches to the problem of predicting how long an Indirect Interaction between two nodes will last. The Indirect Interaction on Facebook is about a pair of unconnected users participating in a common interaction like comments, group messages, shares, and wall posts. Similarly, on Wikipedia, Indirect Interaction is between two pages where users navigate through the search box. This study examines the cases that involved multiple interactions between u and v during an observational time interval $[t_u, t_v]$. The Indirect Interaction between nodes u and v during that time interval indicates that u and v may have something in common, and it might be useful to link them.

Problem Definition: Given two nodes u and v in a network such that during the time interval $[t_u, t_v)$

(1) there is no link between them, and

(2) we observe an "Indirect Interaction" between u and v

predict how long the nodes u and v will keep interacting after t_u.

This problem has applications in link recommendation. When recommending a link between two nodes u and v, we prefer to recommend links that will be useful in the future. In fact, in any social networking platform, it is better to recommend user u to become friends with user v if we know that these two nodes will interact in the future,

rather than suggesting □ as a friend for □ even after we know that □ and □ will never interact.

Thus, the longer the interaction is estimated to last between a pair of nodes; the higher the priority is for recommending a link between them.

In our framework, we consider the time period divided into the following intervals:

a) $[□_□, □_□)$ is the time interval used to observe Indirect Interactions between nodes;

b) $[□_□, □_□]$ is the time interval used to observe when the Indirect Interaction(s) will stop;

c) $(□_□, □_□]$ is the time interval used to observe which links have been formed.

We propose two supervised learning approaches to the problem of predicting the duration of Indirect Interaction. Given a candidate pair of indirectly interacting nodes $□ = (□, □)$ and a vector of network-based predictors $□_□$, the basic approach consists of predicting whether or not the Indirect Interaction $□ = (□, □)$ will last, while the more fine-grained approach consists of estimating how long the Indirect Interaction will last. These approaches are detailed in the following sections.

5.1 Basic Approach

In order to predict whether the Indirect Interaction between a pair of non-linked nodes $□ = (□, □)$ will last or not, we model the problem as a binary classification task where the input features are given by $□_□$ and the positive class is given by the set of instances that do not stop interacting in the time interval $[□_□, □_□]$, i.e., their Indirect Interaction continues after $□_□$. All the instances that stop their Indirect Interaction at any

time $t_s \leq t \leq t_e$ fall into the negative class.

5.2 Fine-grained Approach

To predict the Indirect Interaction duration in the time interval $[t_s, t_e]$, two different methods were used—Survival Analysis and Regression. Next, we describe how we model the problem according to these two methods.

5.2.a Modeling the Problem via Survival Analysis

Survival Analysis is a statistical method to estimate the expected duration of time until an event of interest occurs [41].

We apply Survival Analysis to the interval $[t_s, t_e]$ to compute the survival time of an Indirect Interaction. The event of interest is when the Indirect Interaction between two nodes stops. The time when the event of interest happens for the instance $x = (u, v)$ is denoted by t_x. During the study of our Survival Analysis problem, it is possible that the event of interest was not observed for some instances. This occurs because we are observing the problem in a limited time window $[t_s, t_e]$ or we missed the traces of some instances. If this happens for an instance x we say that x is censored and denote by c_x the censored time. Given an instance x, we denote by v_x the feature vector. Let δ_x be a Boolean variable indicating whether or not the instance x is not censored, i.e., if $\delta_x = 1$ then the instance x is not censored. We denote by y_x the observed time for the instance x that is equal to t_x if x is uncensored and c_x otherwise.

$$y_x = \{ \; t_x \quad \text{if } \delta_x = 1 \; t_x \quad \text{if } \delta_x = 0$$

Given a new instance x described by the feature vector v_x, Survival Analysis estimates a survival function S_x that gives the probability that the event for the instance x will occur after time t.

$S_i(t) = Pr(T_i \geq t)$

The survival function can be used to compute the expected time $\widehat{T_i}$ of event occurrence for an instance i as explained in the next section.

5.2.b Modeling the Problem via Regression

Another way to estimate the duration of an Indirect Interaction is to model the problem as a Regression task, i.e., estimating the parameters θ of a function f such that $T_i \approx f(i, \theta)$, for each instance i in the training set.

The main difference between Regression and Survival Analysis is that regression is not able to incorporate the information coming from censored instances within the predictive model. In the case of regression, censored instances are typically ignored or their observed time t is modeled as constant T_i much bigger than T_i.

5.3 Link Prediction

In this section, we address the problem of predicting whether or not a pair of non-linked nodes showing an Indirect Interaction will become a link in the future. We adopt the Link Prediction framework with a single feature proposed by Liben-Nowell and Kleinberg [6]. This framework consists of the following steps:

a) assign a score to each candidate pair of non-linked nodes,

b) order the candidate pairs in descending order and take the top-n pairs with the highest score,

c) evaluate how many of these top-n pairs are links in the test set.

Our assumption in this study is that pairs of nodes for which we can predict that the Indirect Interaction will last longer should be prioritized concerning candidates that

are predicted to last a shorter time because in the former case linking the two nodes can be more beneficial for both of them. Therefore, we propose to assign $\mathit{score}(u, v)$ to each pair of indirectly interacting nodes (u, v) that are not linked during the time interval $[t_1, t_2)$ that is proportional to the estimated duration of their Indirect Interaction during the time interval $[t_1, t_2]$. As we proposed various methods in the previous section to predict the Indirect Interaction duration, the value of $\mathit{score}(u, v)$ depends on the method used.

5.3.a Classification

When using classification, $\mathit{score}(u, v)$ is equal to the probability given by the classifier that the instance $x = (u, v)$ is in the positive class, i.e., the Indirect Interaction will last.

5.3.b Survival Analysis

When using Survival Analysis, the score is given by the expected time $\widehat{t_e}$ the interaction is predicted to stop, i.e. $\mathit{score}(u, v) = \widehat{t_e}$. The predicted expected time $\widehat{t_e}$ of event occurrence is computed as follows. From the Survival Analysis model, we will have the following probabilities:

$S_e(t_e) = Pr(T_e \geq t_e)$, $S_e(t_e + 1) = Pr(T_e \geq t_e + 1)$, …, $S_e(t_2) = Pr(T_e \geq t_2)$. The probability $Pr(t_1 + k \leq T_e \leq t_1 + k + 1)$, that the Indirect Interaction will stop in the interval $[t_1 + k, t_1 + k + 1)$ is given by

$$Pr\,(t_1 + k \leq T_e \leq t_1 + k + 1) = Pr\,(T_e \geq t_1 + k) - Pr(T_e \geq t_1 + k + 1)$$

for $0 \leq k < n$, where n is the number of units we divided the interval $[t_1, t_2]$ into. The expected time $\widehat{t_e}$ when the Indirect Interaction will stop is given by

$$\widehat{d_i} = \left(\sum_{j=1}^{n} (k+1) \times \mathbb{1}\left(t_j + \delta \le t_i \le t_j + \delta + 1\right)\right) + \left(k \times \mathbb{1}\left(t_i \ge t_n\right)\right)$$

Because we have a limited time window $[t_1, t_n]$ to observe when the Indirect Interaction is stopping, when $\widehat{d_i} = t_n$ we have a lower bound on when the interaction is stopped, i.e., the Interaction can stop at any time $t \ge t_n$, but we do not know exactly when.

5.3.c Regression

When using Regression, $\mathrm{Model}(i, k) = \widehat{d_i}$ where $\widehat{d_i} \approx d(i, k)$ is the estimated duration of the Indirect Interaction i predicted by the Regression model.

CHAPTER 6: LIST OF PREDICTORS

In this chapter, we report the set of predictors used in our thesis. The predictors are computed by considering the network [34] and are divided into three types of features namely Node-based features, Neighborhood-based features, and Network-based features. Additionally, some extra features on page content are used based on the availability of resources (Wikipedia API - sandbox) to gather such information.

6.1 Notations in Formulae:

Let $\square = (\square, \square: \square)$ be an undirected weighted graph where

a) $\square$ is a set of vertices and

b) $\square \subseteq \square \times \square$ is the set of Edge(s)

c) $\square: \square \rightarrow \square^{+}$

Let *u* be a node in $\square$, we denote by $\square(\square)$ is the set of neighbors of node $\square$ where

$$\square(\square) = \{\square \in \square | (\square, \square) \in \square \vee (\square, \square) \in \square\}$$

For directed graphs, we define $\square_{\square\square}(\square)$ to be the set of neighbors pointed towards $\square$, i.e.,

$$\square_{\square\square}(\square) = \{\square \in \square | (\square, \square) \in \square\}$$

And $\square_{\square\square\square}(\square)$ to be the set of neighbors pointed away from $\square$, i.e.,

$$\square_{\square\square\square}(\square) = \{\square \in \square | (\square, \square) \in \square\}$$

6.2 Node-based features

6.2.a Degree

For an undirected graph, the number of edges that connects a node with its neighbors is called Degree [44] of that node, denoted by

$$\text{Degree}(v) = |N(v)|$$

For directed graphs, we consider In-degree and Out-degree denoted by Indegree and Outdegree respectively, where

$$\text{Indegree}(v) = |N_{in}(v)|$$

and $$\text{Outdegree}(v) = |N_{out}(v)|$$

6.2.b Reciprocity

For a directed graph with u, v as nodes, if u has an edge to v, reciprocity is to identify if there was an edge from v to u. The weights on such edges are used as one of the predictors, i.e.,

$$\text{Reciprocity}(u, v) = r(v, u)$$

6.3 Neighborhood-based features

6.3.a Common-Neighbors (CN)

Common Neighbors between nodes u and v is the number of nodes that have a common edge with both nodes u and v. As stated in [6,16,20,21,23], common neighbor is a state-of-the-art measure that can be applied to any network to understand the popularity of the pair of nodes. Common neighbors is given by the following formula.

$CN(u, v) = |N(u) \cap N(v)|$, if the graph is undirected.

$CN(u, v) = |(N_{in}(u) \cup N_{out}(u)) \cap (N_{in}(v) \cup N_{out}(v))|$, if the graph is directed.

6.3.b Jaccard similarity

It is a measure mainly used to compute similarity and diversity of the two nodes. If □and □ are two nodes in a network, Jaccard similarity [6,16] is measured as the intersection of neighbors between *u* and *v* over union of their neighbors. Hence, it is given by the following formula for undirected graphs

$$□□□(□,□) = \frac{|□(□) \cap □(□)|}{|□(□) \cup □(□)|}$$

and for directed graphs,

$$□□□(□,□) = \frac{|(□_{□□}(□) \cup □_{□□□}(□)) \cap (□_{□□}(□) \cup □_{□□□}(□))|}{|(□_{□□}(□) \cup □_{□□□}(□)) \cup (□_{□□}(□) \cup □_{□□□}(□))|}$$

6.3.c Adamic-Adar similarity

It is a similarity measure that weights common neighbors with few connections more heavily. It ensures to prioritize the least connected common neighbor. Mathematically, Adamic-Adar [4] is the sum of the inverse log of the count of neighbors of all the common neighbors between *u* and *v*.

$$□□(□,□) = \sum_{□ \in □(□) \cap □(□)} \frac{1}{□□□|□(□)|}$$

6.3.d Preferential Attachment score

The preferential attachment score is an aggregated neighborhood-size based feature. For this study, degree(s) are aggregated as preferential attachment score [16] and is given by

$□□(□,□) = |□(□)| \times |□(□)|$ for undirected graphs.

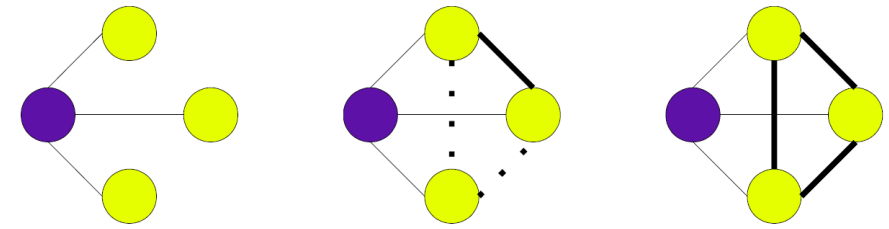
$□□(□,□) = |□_{□□□}(□)| \times |□_{□□}(□)|$ for directed graphs.

6.3.e Local Clustering Coefficient

Local Clustering coefficient [16] is the property of a node within the network. If v is a node, the number of triangles formed using its neighbors with respect to number of its neighbors is the local clustering coefficient. It is the probability that the neighbors of the node are connected.

$$C(v) = \frac{2 \times |\{e_{1,}e_{2} \in E(v)\}|}{|E(v)| \times (|E(v)| - 1)}$$

Figure 10 shows an example for computing local clustering coefficient in a sample network.



(a) No pairs formed among neighbors: ***C = 0*** (b) One pair formed among neighbors: ***C = 1 / 3*** (c) Three pairs formed among neighbors: ***C = 3 / 3***

Figure 10. Graphical representation of local clustering coefficient. Image source: Santillán et al. [35].

6.4 Network-based features

6.4.a PageRank

PageRank [17,33,43] is an algorithm developed by Google to assign a rank on the websites for better search results. It works by counting the number and quality of links to that page. Its purpose is to measure the relative importance of a web page. For example, the higher the PageRank of pages pointing towards page v, higher is the importance (PageRank) of page v. This algorithm can be applied in a network to identify popular nodes. PageRank is represented by the formula:

$$\square\square(\square) = \frac{1-\square}{|\square|} + \square\sum_{\square\in\square_{\square\square}(\square)} \frac{\square\square(\square)}{|\square_{\square\square\square}(\square)|}$$

where d is the dumping factor usually set to 0.85.

Figure 11 shows an example of PageRank for all nodes in an example network.

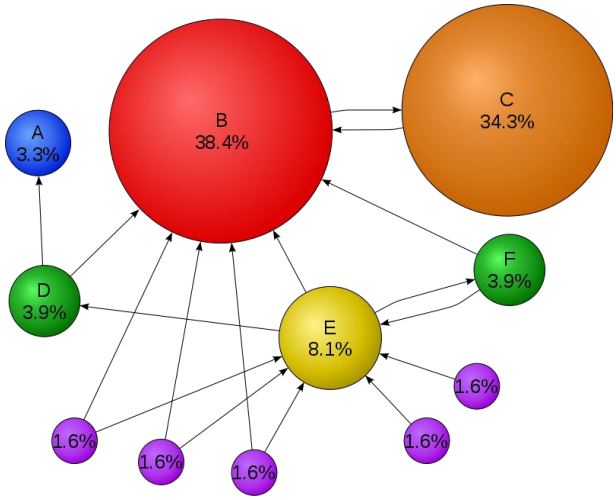


Figure 11. An example of how PageRank is calculated. Image source: https://en.wikipedia.org/wiki/PageRank

6.4.b Node2Vec

Network embedding is a technique for mapping each node of a graph in a geometric high dimensional space. Once the embedding is obtained for each entity, its geometric representation can be used as features in input to machine learning algorithms. Recently, several network embedding techniques have been defined, and all the proposed techniques can be categorized into three broad categories, namely (1) factorization based, (2) random walk based, and (3) deep learning based (see Goyal et al. [45] for a survey). The majority of these embedding techniques work for unlabeled graphs, while Lin et al. [50] propose an embedding model for knowledge graphs. This current study focuses on one embedding technique: Node2Vec (Grover and Leskovec [15]). We chose Node2Vec because it outperforms other embedding techniques such as LINE (Tang and Liu [47]),

Deepwalk (Perozzi et al. [46]), and Spectral Clustering (Tang et al. [48]) in the task of node Link Prediction. Node2Vec is an embedding technique based on random walks. It computes the embedding in two steps. First, the context of a node (or neighborhood at distance) is approximated with biased random walks of length that provides a trade-off between breadth-first and depth-first graph searches. Second, the values of the embedding features for the node are computed by maximizing the likelihood of generating the context by the given node. Node2Vec uses only the structure of the network and does not consider any node or edge label.

As the node features are computed for each in pair (,), edge features can be learned with a choice of any below binary operator on those node features. For and nodes, $_{}()$ and $_{}()$ are their respective Node2Vec features.

a) Adamard: $\frac{(_{}()+_{}())}{2}$

b) Hadamard: $(_{}() \times _{}())$

c) Weighted-L1: $|_{}() - _{}()|$

d) Weighted-L2: : $|_{}() - _{}()|^2$

The Hadamard was used in this study for learning edge features as it was shown to perform best by Grover and Leskovec [15]. Regarding the significance of weights on the edges, we used hits as weights on the network [20]. We used 40 dimensions with six walk-length, and variable set to 0.3 as parameters for the Node2Vec algorithm.

6.5 Additional Wikipedia Page features

As page content on Wikipedia can be retrieved either through parsing of web pages or from the dumps available through the Wikimedia Foundation, the following set

of additional features can be included in the dataset. However, for Facebook, as the data is anonymized, it is difficult to retrieve individual node properties like content from their wall posts, comments or any other information that can help to identify additional node/edge properties.

6.5.a Categories' similarity

Each Wikipedia page falls under a set of categories. As an example, the page 'Niagara Falls' falls under the set of categories Waterfalls of Ontario, Block waterfalls, Waterfalls of New York, and so forth. Thus, pages with more categorical similarities may have some relatedness between them [18]. Let u and v be Wikipedia pages; we define $cat(u)$ as the set of categories of page u and $pages(c)$ as the set of pages that belong to category c. Then we defined categories' similarity for a pair of pages u and v in terms of:

a) the Jaccard similarity between the pages' categories as

$$sim_{jac}(u,v) = \frac{|cat(u) \cap cat(v)|}{|cat(u) \cup cat(v)|}$$

b) the Adamic-Adar similarity as

$$sim_{aa}(u,v) = \sum_{w \in cat(u) \cap cat(v)} \frac{1}{\log |pages(w)|}$$

c) the Preferential Attachment score on pages' categories as

$$sim_{pa}(u,v) = |cat(u)| \times |cat(v)|$$

CHAPTER 7: EXPERIMENTS

As explained in the framework for our approach, we distributed each of the datasets into three time periods. The time periods were set as follows:

a) for Facebook, we set t_0 = September 2006, t_1 = January 2007, t_2 = t_3 = December 2008, and t_4 = January 2009;

b) for Wikipedia, we set t_0 = February 2016, t_1 = March 2016, t_2 = April 2016, t_3 = July 2016 and t_4 = August 2016.

For each dataset, we selected all pairs of nodes *I* that showed Indirect Interaction(s) in the time interval $[t_0, t_2)$. Then, for each pair $i = (u, v) \in I$, we checked whether (u, v) continued to interact in the interval persistently. If not, we set $\delta_i = 0$, if yes, then δ_i was set to the time $t \in [t_1, t_2]$ when the persistent interaction stopped, and if the persistent interaction never stopped in the interval $[t_1, t_2]$, we considered the instance i to be censored. The number of instances of nodes with Indirect Interactions and the number of censored instances is reported for each dataset in Table 5. For Wikipedia, we filtered *other* types of tuples having *prev* or *curr* page title as Wikipedia's main page from February 2016. This is because the Main page is a constantly changing Wikipedia article and the searches from or to this page represent noise in the dataset. We computed the set of predictors by considering the status of the network during the time interval $[t_0, t_1)$. For Link Prediction ground truth, we considered new links formed in the time interval $(t_1, t_2]$.

In all the datasets, we considered the class imbalance problem (i.e., we have more negative instances (instances that stopped interacting anytime $t \in [t_p, t_f]$ than positive instances (instances that did not stop interacting in the observational period); these are censored instances). We used a majority under sampling strategy for a similar problem [42]. For each dataset, we created a pool of sub-datasets. First, we created ten random samples of the majority class whose size is set to be the same as one of the minority class. Then, we added each of these samples to all the instances in the minority class and performed a five-fold cross-validation on each of those ten balanced datasets. We finally averaged the results obtained from all the five-fold cross-validated datasets. We used the same subsets of datasets across all the experiments.

Table 5: Indirect Interactions

Dataset	Instances of Indirect Interactions	Censored Instances
Facebook	175,577	4,155
Wikipedia	2.03M	190,124

7.1 Predicting Duration of Indirect Interactions

The first problem studied in this thesis was to estimate the duration of interactions in each of the datasets. This problem was addressed in two ways. First, the binary classification approach was used to determine whether or not they will continue to interact at time t_f. Second, the more fine-grained approach was used to estimate the duration of interactions using Survival Analysis and Regression.

7.1.a Will the Indirect Interactions last or not?

In this experiment, we consider our predictors as input to a binary classifier to

predict whether the Indirect Interactions would last or not. In this case, we considered censored instances as positive instances (i.e., to say that the interaction will last) and instances where the event occurred (i.e., they stopped interacting) as negative instances.

For each dataset, using all the listed predictors, five classification models are used to determine the best fit model in predicting whether or not they will continue to interact at time □□. Table 6 below shows the results obtained using the following classifiers.

1. K-nearest neighbors with number of neighbors set as 5.
2. Random Forests with 100 trees and criterion set as "entropy."
3. Linear-Support Vector Machine with a maximum of 100 iterations.
4. Radial Basis Function (RBF) kernel of SVM.
5. Linear model's Logistic Regression.

Table 6: Results from Classifiers - Facebook

Classifier	Accuracy	AUROC	MAP	Precision (class1)	Precision (class0)
KNN	0.603	0.633	0.648	0.610	0.597
Linear SVM	0.680	0.743	0.751	0.734	0.646
Logistic Regression	0.677	0.737	0.746	0.715	0.651
Random Forests	0.832	0.901	0.894	0.808	0.861
SVM_RBF	0.500	0.505	0.598	0.541	0.468

Considering AUROC as the best metric to evaluate the performance, the classifier with the best performance is Random Forests with an AUROC of 90%. By using the

same parameters on the classifiers for experiments on the Wikipedia dataset, Table 7 shows the results obtained.

Table 7: Results from Classifiers - Wikipedia

Classifier	Accuracy	AUROC	MAP	Precision (class1)	Precision (class0)
KNN	0.668	0.720	0.732	0.671	0.664
Linear SVM	0.631	0.689	0.693	0.674	0.605
Logistic Regression	0.634	0.694	0.703	0.678	0.607
Random Forests	0.712	0.783	0.785	0.730	0.697
SVM_RBF	0.499	0.507	0.549	0.416	0.495

The classifier with the best performance is Random Forests with an AUROC of 78.3%. Tables 6 and 7 show results by using similar features in both datasets. However, by including additional categorical features on the Wikipedia Dataset, the following results are observed.

Table 8: Results from Classifiers including Categorical Features - Wikipedia

Classifier	Accuracy	AUROC	MAP	Precision (class1)	Precision (class0)
KNN	0.697	0.761	0.762	0.698	0.696
Linear SVM	0.659	0.731	0.726	0.742	0.619
Logistic Regression	0.670	0.730	0.732	0.692	0.663
Random Forests	0.747	0.834	0.825	0.743	0.752
SVM_RBF	0.499	0.505	0.481	0.433	0.498

The classifier with the best performance is Random Forests with an AUROC of 83.4% and is higher than the previously achieved AUROC of 78.3%. From the results above, it is evident that inclusion of available categorical features improved the system's performance.

7.1.b Feature Importance

Of all the predictors used for the approach, we measured the importance of each predictor to understand their contribution towards the performance of the approach. The following figures compare the 'feature importance' for each of the predictors.

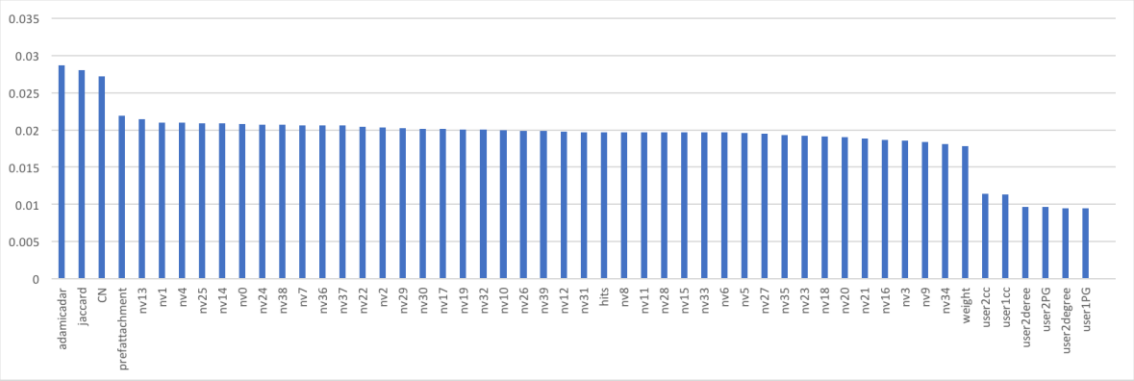


Figure 12. Feature importance for the Facebook dataset.

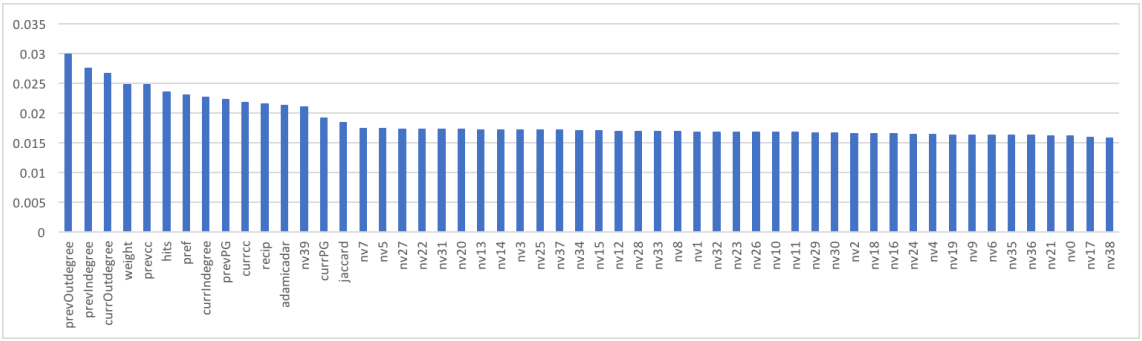


Figure 13. Feature importance for the Wikipedia dataset.

In each dataset's feature importance, the predictor prefix 'nv' denotes the Node2Vec edge feature, and suffix 'cc' denotes Local Clustering Coefficient. For Facebook, while the pair of nodes u, v is denoted by *user1* and *user2*, for Wikipedia,

they are denoted by *prev* and *curr*. Also, ‘CN’ denotes Common Neighbors, ‘PG’ denotes PageRank, ‘weight’ is the reciprocal edge weight and ‘recip’ is Reciprocity of the edge. As we can see in the figures, the highest important features vary for different datasets. There is no concrete notion as to which of these predictors are commonly important for any dataset.

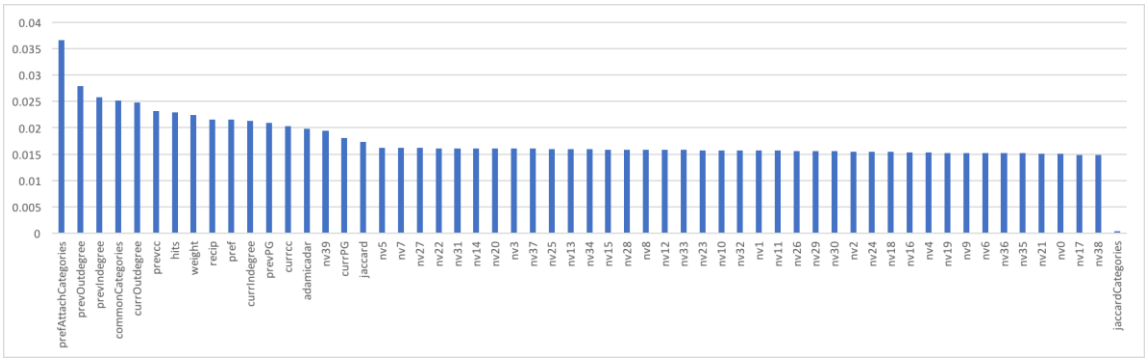


Figure 14. Feature importance for Wikipedia dataset (with Categorical features).

Figure 14 shows the order of important features for the Wikipedia dataset including additional categorical features. It is evident that these additional features contribute comparatively higher towards better performance.

7.1.c Comparison of Classification with Baselines

Some of the important predictors that are capable of predicting independently are Hits, Jaccard similarity score, Adamic-Adar similarity score, Preferential Attachment score and Node2Vec features. As we had evaluated 40 Node2Vec features for each node, we calculated Cosine similarity of those node features to construct it in a single column. These predictors are scores evaluated on pairs (□, □) together. To effectively determine the performance of our proposed system, it was necessary to compare and understand if it could perform better than the baselines. We used the following features to compare:

1. Jaccard similarity score

2. Adamic-Adar similarity score
3. Preferential Attachment score
4. Cosine similarity of Node2Vec node features.
5. Hit counts

Only the above-stated features were chosen as they determine the feature of (□, □) together in a network. Table 9 and 10 show baselines' results for both the datasets.

Table 9: Results for Baselines - Facebook

Baselines	AUROC	MAP
Hits	0.633	0.693
Jaccard Similarity	0.707	0.701
Adamic Adar Similarity	0.292	0.386
Preferential Attachment	0.612	0.624
Node2Vec	0.576	0.579

Table 10: Results for Baselines - Wikipedia

Baselines	AUROC	MAP
Hits	0.455	0.459
Jaccard Similarity	0.556	0.556
Adamic Adar Similarity	0.573	0.611
Preferential Attachment	0.572	0.591
Node2Vec	0.583	0.570

7.1.d How long will the Indirect Interaction last?

Using Survival Analysis (see Chapter Four 2nd section) and Regression models (see Chapter Four 3rd section), it is possible to estimate the probability of "will they stop

interacting at time t_i." We then used the predicted probability to calculate the Survival Function values. For each pair of nodes in the datasets, we calculated the expected value of survival probabilities, i.e., $S(x, t)$. In Survival Analysis, we used the Accelerated Failure Time (AFT) model with five distributions, and three Regression models:

1. For AFT:
    a. Weibull
    b. LogNormal
    c. Exponential
    d. Cox
2. Regression:
    a. Ridge Regression
    b. Lasso Regression
    c. Support Vector Regression (SVR) using 'rbf' kernel

As these datasets have censored information; classical AUROC is not suitable. To compare the performances of these two sets of algorithms, we considered two metrics that are commonly used to evaluate Survival Analysis models, namely c-index and Mean Absolute Error (MAE) [41]. The c-index is given by the formula:

$$c - index = \frac{1}{n_{pairs}} \sum_{i:\delta_i=1} \sum_{t_i>t_j} 1(\hat{t}_i > \hat{t}_j)$$

where n_{pairs} is the number of all pairs (t_i, t_j) such that $\delta_i = 1$ (non-censored instances) and it holds that the time t for the latter is greater than the former (i.e., $t_i > t_j$), and $\hat{t}$ is the estimated duration predicted by the model. The c-index measures the concordance probability between the actual observation times and the predicted values. It

is worth noting that the baseline value for the c-index of a random classifier is not 0.5, but it is 0.31 for Facebook and 0.22 for Wikipedia according to the distribution of observed times for events of interest in the datasets.

The mean absolute error is defined as the average absolute difference between the predicted duration of the event and the actual one. It is calculated by using the formula:

$$\square\square\square = \frac{1}{\square}\sum_{\square=1}^{\square} \quad (\square_{\square}|\square_{\square} - \widehat{\square}_{\square})$$

where *N* is the number of non-censored instances, i.e., for which the event occurred. The results reported in this thesis for MAE are normalized.

Also, we considered the Pearson Correlation Coefficient (PCC) between the actual time and the predicted time of event occurrence. Higher c-index and PCC, and lower MAE are desirable. Table 11 and 12 report the values for the above three metrics for Survival Analysis on both the datasets.

Table 11: Results of Survival Analysis - Facebook

Models	c-index	MAE	Pearson CC
Cox	0.498	0.250	0.459
Weibull	0.499	0.350	0.460
Exponential	0.499	0.350	0.460
LogNormal	0.499	0.350	0.460

Table 12: Results of Survival Analysis - Wikipedia

Models	c-index	MAE	Pearson CC
Cox	0.502	0.118	0.320
Weibull	0.285	0.363	0.208
Exponential	0.501	0.112	0.315
LogNormal	0.501	0.142	0.198

Table 13 and 14 show the results for c-index, MAE, and PCC metrics by using Regression Models on Facebook and Wikipedia datasets.

Table 13: Results for Regression - Facebook

Models	c-index	MAE	Pearson CC
Ridge Regression	0.497	0.384	0.409
Lasso Regression	0.497	0.385	0.409
SVR rbf	0.330	0.222	0.057

Table 14: Results for Regression - Wikipedia

Models	c-index	MAE	Pearson CC
Ridge Regression	0.501	0.094	0.910
Lasso Regression	0.500	0.095	0.910
SVR rbf	0.418	0.375	0.035

By looking at these results, it is clear that for Facebook, Survival Analysis with the Cox Model performs better than Regression in predicting the duration of the Interaction. In fact, both the Cox model, and Ridge and Lasso Regression achieve the same value of c-index, but Cox has a lower MAE (0.25 versus 0.38) and a higher PCC

(0.46 versus 0.41). Even though SVR has the best MAE (0.22), its values for c-index and PCC are bad. In the case of Wikipedia, c-index was not able to differentiate between Survival Analysis and Regression, but according to MAE and PCC, Regression (either Ridge or Lasso) was the best with MAE of 0.09 and PCC of 0.91. In comparison, Cox achieves 0.11 for MAE and 0.32 for PCC.

7.1.e Comparison of Survival Analysis and Regression with Baselines

Table 15 and 16 show the c-index, MAE, and PCC regarding the baselines. In this case, we are using the values of the baselines to approximate the Duration of Interaction.

Table 15: Results for Baselines - Facebook

Baselines	c-index	MAE	Pearson CC
Hits	0.213	0.162	0.28
Jaccard Similarity	0.500	0.339	0.323
Adamic Adar Similarity	0.500	0.292	0.355
Preferential Attachment	0.496	0.345	0.076
Node2Vec	0.496	0.340	0.146
Our Approach	0.502	0.118	0.320

Table 16: Results for Baselines - Wikipedia

Baselines	c-index	MAE	Pearson CC
Hits	0.507	0.418	0.133
Jaccard Similarity	0.531	0.410	0.083
Adamic Adar Similarity	0.544	0.415	0.146
Preferential Attachment	0.534	0.418	0.066
Node2Vec	0.544	0.412	0.055

Our Approach	0.501(-1)	0.094(+1)	0.910

Again, the c-index results proved not to be a good measure to compare our approaches with the baselines because their values were all comparable. Overall, by considering the values from all the metrics, our approach is better than the baselines for both the datasets. For Facebook, our best MAE and PCC obtained with Survival Analysis are better than all the baselines except for Hits that beat us regarding MAE. However, hits achieve very low values for c-index and PCC in comparison to our approach. For Wikipedia, our results obtained with Regression are always better than the baselines according to MAE and PCC and comparable with respect to the c-index.

7.2 Link Prediction

On Wikipedia, the problem was predicting whether or not there should be a hyperlink placed from page □ to page □. We considered the hyperlinks present in type "link" of the August 2016 Clickstream dataset as ground truth. For Facebook, the problem was predicting whether or not two nodes □ and □ should become friends. As the ground truth for Facebook, we used the links formed in January 2009. In this experiment, we first trained the model (Binary Classification, Survival Analysis, or Regression) to predict when (or "if" for Binary Classification) the Indirect Interaction would stop and then measured the AUROC between these predicted times and the class values (1 if the link has been created, 0 otherwise). For the case of classification, we considered the predicted probability as having a link in the future.

The following tables show the AUROC values for Binary Classification, Survival Analysis, and Regression. As we can see in Table 17 and 18, by using the information on whether the interaction will last or not, achieved a good AUROC of 0.8 with linear SVM

for the Facebook dataset. Unfortunately, this was not sufficient for the Wikipedia dataset where this approach was not working the same (AUROC of 0.5 with SVM by using RBF kernel). Instead, as Tables 19, 20 and 21, 22 show, if we were using a more fine-grained approach (Survival Analysis or Regression), i.e., predicting how long the interaction will last, we could improve the previous results and achieve an AUROC of 0.854 on the Facebook dataset with Survival Analysis (Exponential algorithm) and 0.769 on the Wikipedia dataset with Regression (either Lasso or Ridge Regression algorithm).

7.2.a Classification approach

To maintain consistency in the analyses, similar classifiers as that of the first problem were used on both datasets. However, the classes were changed. For these experiments, any candidate pair from who existed as a link in network snapshot at time belonged to the positives class. All the other pairs that did not have a link at time belonged to the negatives class. We used the same subset of datasets for all the experiments. We trained the model on predicting the duration of Interaction and tested for Link Prediction. Table 16 and 17 below shows the results obtained for different classifiers on Facebook and Wikipedia datasets.

Table 17: Results from Classifiers - Facebook

Classifier	Accuracy	AUROC
KNN	0.535	0.634
Linear SVM	0.621	0.800
Logistic Regression	0.592	0.793
Random Forests	0.469	0.770
SVM rbf	0.939	0.497

Considering AUROC as the best metric to evaluate the performance, as the dataset is skewed, the classifier with the best performance is “Linear SVM” for the Facebook dataset.

Table 18: Results from Classifiers - Wikipedia

Classifier	Accuracy	AUROC
KNN	0.457	0.451
Linear SVM	0.461	0.476
Logistic Regression	0.494	0.486
Random Forests	0.440	0.438
SVM rbf	0.441	0.500

The classifier with the best performance is “SVM rbf” for the Wikipedia dataset. The low values were probably because of validation made with August 2016 dataset which is four months away from the time period $□_{□}$.

7.2.b Survival Analysis and Regression

Similarly, for the problem of Survival Analysis, we determined the positives and negatives class. We used the results obtained from Survival Analysis of the first problem and calculated the probability of expected value, i.e., $\frac{□(□,□)}{|□-1|}$ where $□(□, □)$ is the expected value of the survival probabilities for each pair. While Survival Analysis predicts probabilities for each period, Ridge Regression determines a single estimated duration of time. For Regression models, we took the probability of those predicted times, i.e., $\frac{□□□□(□,□)}{|□-1|}$ where $□□□□(□, □)$ is the probability from Regression Model. We used AUROC and Mean-Average Precision metrics to evaluate if those values could determine

which class these pairs belong to. Table 19, 20 and 21, 22 shows the results for Survival Analysis and Regression for each of the datasets.

Table 19: Results from Survival Analysis - Facebook

Model	AUROC
Cox	0.812
Weibull	0.822
Exponential	0.854
LogNormal	0.836

Table 20: Results from Survival Analysis - Wikipedia

Model	AUROC
Cox	0.519
Weibull	0.517
Exponential	0.518
LogNormal	0.528

Table 21: Results from Regression models - Facebook

Model	AUROC
Ridge Regression	0.779
Lasso Regression	0.790
SVR rbf	0.538

Table 22: Results from Regression models - Wikipedia

Regression model	AUROC
Ridge Regression	0.768
Lasso Regression	0.769
SVR rbf	0.499

To determine whether or not Regression was the best algorithm for the Link Prediction problem on Wikipedia's dataset, we experimented on a classical approach by training the class of links on both the dataset's best performing approach. Based on the following results, we determined that the Regression Model performed better. For Facebook, though AUROC showed satisfactory results, the values for the MAP are way too low. The reason for these numbers is that the datasets are very largely skewed towards the negative class.

7.2.c Comparison of both approaches

Table 23: Comparison of Approaches - Facebook

Model	AUROC
Classification (Linear SVM)	0.80
Survival Analysis (Exponential)	0.854
Regression(Lasso)	0.790

Table 24: Comparison of Approaches - Wikipedia

Model	AUROC
Classification (SVM rbf)	0.500
Survival Analysis (LogNormal)	0.528
Regression(Lasso)	0.769

Table 23 and Table 24 shows the best of three approaches and Survival Analysis is proven to perform better than other approaches for the Link Prediction problem on Facebook and Wikipedia datasets. Hence, it is proved that when using a more fine-grained approach (Survival Analysis or Regression), i.e., predicting how long the interaction will last, we can achieve an AUROC of 0.854 on Facebook datasets with Survival Analysis (Exponential algorithm), and 0.769 on Wikipedia datasets with Regression (either Lasso or Ridge Regression).

Table 25: Traditional Link Prediction Approach- Wikipedia

Model	AUROC
KNN	0.679
Logistic Regression	0.675
Linear SVM	0.634
Random Forests	0.437
SVM rbf	0.503

Table 26: Traditional Link Prediction Approach- Facebook

Model	AUROC
KNN	0.634
Logistic Regression	0.800
Linear SVM	0.497
Random Forests	0.793
SVM rbf	0.770

Another experiment was performed for comparison purposes. We used all the

predictors computed with a snapshot of the network during the time interval $[t_0, t_0)$ as input to the binary classifier to directly predict whether or not there would be a link during the time interval $(t_0, t_0]$. Results are reported in Table 25 and 26. As we can see, if we skip the approach of predicting the Duration of Interaction (i.e., using a model based only on network properties), it performs with an AUROC of 0.80 and 0.68 on Facebook and Wikipedia datasets respectively for the Link Prediction problem, which is lower than what we could achieve with our best approach that considers the predicted Duration of Interaction.

7.2.d Comparison with baselines

For the problem of predicting the duration of interaction, the baselines gave the following results.

Table 27: Baselines - Facebook

Baselines	AUROC
Hits	0.565
Jaccard Similarity	0.645
Adamic Adar Similarity	0.206
Preferential Attachment score	0.578
Node2Vec	0.527
Our Approach	0.854

From tables 27 and 28 for Facebook and Wikipedia respectively, it is observed that our proposed system outperforms baselines for Link Prediction problem as well.

Table 28: Baselines - Wikipedia

Baseline	AUROC
Hits	0.445
Jaccard Similarity	0.555
Adamic Adar Similarity	0.573
Preferential Attachment score	0.572
Node2Vec	0.583
Our Approach	0.769

7.3 Comparison with Paranjape et al. [12]

This study used Persistent Indirect Interactions as a base for predicting links on Wikipedia. Related work close to our approach was conducted by Paranjape et al. [12]. They designed an unsupervised algorithm that will prioritize and predict top K links that should be added to each page. We created a Supervised version of their efficient algorithm $\square_3$ and passed our Wikipedia dataset as input with K value equal to the number of Positives in our dataset. We validated the resulted estimates against our Positives and Negatives. Their other algorithms $\square_1$ and $\square_2$ gave similar results. Table 29 summarizes our results:

Table 29: Comparison with Paranjape et al. - Wikipedia

Algorithm	AUROC
$\square_3$ (Predicting Duration of Interaction)	0.5
Thesis (Predicting Duration of Interaction)	0.783
$\square_3$ (Link Prediction)	0.5
Thesis (Link Prediction)	0.769

From the above table, it is clear that our system performed better than an existing approach for both Predicting Duration of Interaction and Link Prediction on Wikipedia's dataset.

7.4 Summary

Based on all the experiments, the current thesis' results show that Survival Analysis works best on Facebook datasets, whereas Regression Model performs the best on Wikipedia datasets. The experiments showed maximum improvement for predicting the Duration of Interaction and achieved an AUROC of 85.4% on Facebook and 77% on Wikipedia datasets for the Link Prediction problem. Also, by including other Available Categorical features on Wikipedia datasets, the models performed better than that without inclusion of those features. We also observed that if we do not consider the predicted interactivity, a model based only on network properties performed with 80% and 68% AUROC on Facebook and Wikipedia datasets, respectively, on the Link Prediction problem which is lower than what was achieved by considering predicted interactivity.

CHAPTER 8: CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this thesis, we proposed a novel approach to predict links between a pair of nodes that show Indirect Interactions. We addressed the problem in two steps. First, we focused on the problem of predicting how long two nodes would interact in a network by identifying potential pairs of nodes (□, □) that are not connected, yet show some Indirect Interactions. Second, once the Duration of Interaction was estimated, we leveraged this information for the Link Prediction problem.

We proposed two supervised learning approaches to predict Duration of Interaction. Given a set of network-based predictors, the basic approach consisted of learning a Binary Classifier to predict whether or not an observed Indirect Interaction would continue in the future. The second and more fine-grained approach consisted of estimating how long the interaction would last by modeling the problem via Survival Analysis or as a Regression task.

Experiments were conducted on the longitudinal Facebook network and wall interactions, and Wikipedia Clickstream datasets to test our approach to the Link Prediction problem. The experimental results for the Survival Analysis on Facebook datasets and Regression model on Wikipedia datasets showed maximum improvement for predicting the Duration of Interaction (MAE 0.25 and PCC 0.46 on Facebook and MAE 0.09 and PCC 0.91 on Wikipedia) and achieved an AUROC of 0.85 on Facebook and 0.77 on Wikipedia for Link Prediction. We also observed that if we do not consider the

predicted Duration of Interaction, a model based only on network properties performed with 0.80 and 0.68 AUROC on Facebook and Wikipedia datasets respectively on the Link Prediction problem, which is lower than what was achieved by considering predicted Duration of Interaction.

8.2 Future Work

As there is always a scope for improvement, more predictor variables can be included in the approach. Page similarities determine which pages are closely related [7, 26, 27], though it need not be the only main feature to predict connections; along with other features, it can help to achieve good results. One of the measures to calculate page content similarity is Latent Dirchlet allocation (LDA) which can be used by learning on corpora of Wikipedia network. It uses words in the document to identify topics and how much the contents of the document relate to the topics.

Various datasets can be used to apply our approach. For example, the Amazon co-purchased network can also be applied to stock markets to identify a pair of stocks that follow similar trends and this information can be helpful to invest smartly. In cyber security, this approach can be applied to identify differently behaving user accounts that tend to violate the security standards. Also, as the proposed system is independent of the language of the content, it can be applied to various language versions of Wikipedia [13] as well.

REFERENCES

[1] Wikipedia. The Free Encyclopedia, 2005. URL: http: //www.wikipedia.org

[2] https://en.wikipedia.org/wiki/Main_Page

[3] E. Wulczyn and D. Taraborelli. Wikipedia Clickstream. Website, 2015. http://dx.doi.org/10.6084/m9.figshare.1305770 (accessed October 10, 2016).

[4] Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. Social networks, 25(3), 211-230.

[5] Adafre, S. F., & de Rijke, M. (2005, August). Discovering missing links on Wikipedia. In Proceedings of the 3rd international workshop on Link discovery (pp. 90-97). ACM.

[6] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. journal of the Association for Information Science and Technology, 58(7), 1019-1031.

[7] Milne, D., & Witten, I. H. (2008, October). Learning to link with wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management (pp. 509-518). ACM.

[8] Witten, I., & Milne, D. (2008, July). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA (pp. 25-30).

[9] Noraset, T., Bhagavatula, C., & Downey, D. (2014, October). Adding High-Precision Links to Wikipedia. In EMNLP (pp. 651-656).

[10] West, R., Paranjape, A., & Leskovec, J. (2015, May). Mining missing hyperlinks from human navigation traces: A case study of Wikipedia. In Proceedings of the 24th international conference on World Wide Web (pp. 1242-1252). ACM.

[11] West, R., Precup, D., & Pineau, J. (2009, November). Completing wikipedia's hyperlink structure through dimensionality reduction. In Proceedings of the 18th ACM conference on Information and knowledge management (pp. 1097-1106). ACM.

[12] Paranjape, A., West, R., Zia, L., & Leskovec, J. (2016, February). Improving website hyperlink structure using server logs. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (pp. 615-624). ACM.

[13] https://meta.wikimedia.org/wiki/List_of_Wikipedias

[14] https://stats.wikimedia.org/EN/TablesWikipediaZZ.htm

[15] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 855-864). ACM.

[16] Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006, April). Link Prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security.

[17] Chung, F., & Zhao, W. (2010). PageRank and random walks on graphs. In Fete of combinatorics and computer science (pp. 43-62). Springer Berlin Heidelberg.

[18] Suyehira, K., & Spezzano, F. (2016, October). DePP: A System for Detecting Pages to Protect on Wikipedia. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 2081-2084). ACM.

[19] Zignani, M., Gaito, S., & Rossi, G. P. (2016, April). Predicting the Link Strength of “Newborn” Links. In Proceedings of the 25th International Conference Companion on World Wide Web (pp. 147-148). International World Wide Web Conferences Steering Committee.

[20] Kumar, S., Spezzano, F., Subrahmanian, V. S., & Faloutsos, C. (2016, December). Edge Weight Prediction in Weighted Signed Networks. In Data Mining (ICDM), 2016 IEEE 16th International Conference on (pp. 221-230). IEEE.

[21] Lü, L., & Zhou, T. (2011). Link Prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications, 390(6), 1150-1170.

[22] Sarukkai, R. R. (2000). Link Prediction and path analysis using Markov chains. Computer Networks, 33(1), 377-386.

[23] Papadimitriou, A., Symeonidis, P., & Manolopoulos, Y. (2012). Fast and accurate Link Prediction in social networking systems. Journal of Systems and Software, 85(9), 2119-2132.

[24] Liu, J., & Deng, G. (2009). Link Prediction in a user–object network based on time-weighted resource allocation. Physica A: Statistical Mechanics and its Applications, 388(17), 3643-3650.

[25] Kunegis, J., & Lommatzsch, A. (2009, June). Learning spectral graph transformations for Link Prediction. In Proceedings of the 26th Annual International Conference on Machine Learning (pp. 561-568). ACM.

[26] Tombros, A., & Ali, Z. (2005, March). Factors affecting web page similarity. In European Conference on Information Retrieval (pp. 487-501). Springer Berlin Heidelberg.

[27] Smucker, M. D., & Allan, J. (2007, July). Using similarity links as shortcuts to relevant web pages. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 863-864). ACM.

[28] http://dblp.uni-trier.de/ Computer Science Bibiliography

[29] Viswanath, B., Mislove, A., Cha, M., & Gummadi, K. P. (2009, August). On the evolution of user interaction on Facebook. In Proceedings of the 2nd ACM workshop on Online social networks (pp. 37-42). ACM.

[30] A. Clemesha. The Wiki Game. Website, 2009. http://www.thewikigame.com

[31] R. West. Wikispeedia. Website, 2009. http://cs.mcgill.ca/~rwest/wikispeedia/

[32] R. West, J. Pineau, and D. Precup. Wikispeedia: An online game for inferring semantic distances between concepts. In IJCAI, 2009.

[33] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab.

[34] Al Hasan, M., & Zaki, M. J. (2011). A survey of Link Prediction in social networks. In Social network data analytics (pp. 243-275). Springer US.

[35] Santillán, L. C., & Pérez, A. C. (2016). Discovering epistemological axes for academic programs in computer science through network analysis. ReCIBE, 1(1).

[36] Wilson, C., Boe, B., Sala, A., Puttaswamy, K. P., & Zhao, B. Y. (2009, April). User interactions in social networks and their implications. In Proceedings of the 4th ACM European conference on Computer systems (pp. 205-218). Acm.

[37] Kahanda, I., & Neville, J. (2009). Using Transactional Information to Predict Link Strength in Online Social Networks. ICWSM, 9, 74-81.

[38] Kamath, K., Sharma, A., Wang, D., & Yin, Z. (2014). RealGraph: User interaction prediction at twitter. In User Engagement Optimization Workshop@ KDD.

[39] Gilbert, E., & Karahalios, K. (2009, April). Predicting tie strength with social media. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 211-220). ACM.

[40] Yang, Y., & Zou, H. (2012). A cocktail algorithm for solving the elastic net penalized Cox's Regression in high dimensions. Statistics and its Interface, 6(2), 167-173.

[41] Wang, P., Li, Y., & Reddy, C. K. (2017). Machine learning for Survival Analysis: A survey. arXiv preprint arXiv:1708.04649.

[42] Dave, V. S., Al Hasan, M., & Reddy, C. K. (2017). How Fast Will You Get a Response? Predicting Interval Time for Reciprocal Link Creation. In ICWSM (pp. 508-511).

[43] Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18), 3825-3833.

[44] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social networks, 1(3), 215-239.

[45] Goyal, P., & Ferrara, E. (2017). Graph Embedding Techniques, Applications, and Performance: A Survey. arXiv preprint arXiv:1705.02801.

[46] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710). ACM.

[47] Tang, L., & Liu, H. (2011). Leveraging social media networks for classification. Data Mining and Knowledge Discovery, 23(3), 447-478.

[48] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015, May). Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web (pp. 1067-1077). International World Wide Web Conferences Steering Committee.

[49] Dhakal, N., Spezzano, F., & Xu, D. (2017). Predicting Friendship Strength for Privacy Preserving: A Case Study on Facebook. In 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM ’17).

[50] Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, January). Learning Entity and Relation Embeddings for Knowledge Graph Completion. In AAAI (pp. 2181-2187).

[51] Xiang, R., Neville, J., & Rogati, M. (2010, April). Modeling relationship strength in online social networks. In Proceedings of the 19th international conference on World wide web (pp. 981-990). ACM.

[52] Onnela, J. P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., ... & Barabási, A. L. (2007). Structure and tie strengths in mobile communication networks. Proceedings of the national academy of sciences, 104(18), 7332-7336.

[53] Kashima, H., & Abe, N. (2006, December). A parameterized probabilistic model of network evolution for supervised Link Prediction. In Data Mining, 2006. ICDM'06. Sixth International Conference on (pp. 340-349). IEEE.

[54] West, R., Paskov, H. S., Leskovec, J., & Potts, C. (2014). Exploiting social network structure for person-to-person sentiment analysis. arXiv preprint arXiv:1409.2450.

[55] Jones, J. J., Settle, J. E., Bond, R. M., Fariss, C. J., Marlow, C., & Fowler, J. H. (2013). Inferring tie strength from online directed behavior. PloS one, 8(1), e52168.

[56] Arnaboldi, V., Guazzini, A., & Passarella, A. (2013). Egocentric online social networks: Analysis of key features and prediction of tie strength on Facebook. Computer Communications, 36(10), 1130-1144.

[57] Singer, J. D., & Willett, J. B. (1991). Modeling the days of our lives: using Survival Analysis when designing and analyzing longitudinal studies of duration and the timing of events. psychological Bulletin, 110(2), 268.

[58] Harrell Jr, F. E. (2015). Regression modeling strategies: with applications to linear models, logistic and ordinal Regression, and Survival Analysis. Springer.

[59] Murtaugh, P. A., Burns, L. D., & Schuster, J. (1999). Predicting the retention of university students. Research in higher education, 40(3), 355-371.

[60] Richardson, M., Dominowska, E., & Ragno, R. (2007, May). Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th international conference on World Wide Web (pp. 521-530). ACM.

[61] Cacheda, F., Barbieri, N., & Blanco, R. (2017, February). Click Through Rate Prediction for Local Search Results. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 171-180). ACM.

[62] Rakesh, V., Lee, W. C., & Reddy, C. K. (2016, February). Probabilistic group recommendation model for crowdfunding domains. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (pp. 257-266). ACM.

[63] Li, Y., Rakesh, V., & Reddy, C. K. (2016, February). Project success prediction in crowdfunding environments. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (pp. 247-256). ACM.

[64] Ameri, S., Fard, M. J., Chinnam, R. B., & Reddy, C. K. (2016, October). Survival Analysis based framework for early prediction of student dropouts. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 903-912). ACM.

[65] Zafarani, R., Abbasi, M. A., & Liu, H. (2014). Social media mining: an introduction. Cambridge University Press.

[66] http://scikit-learn.org/stable/modules/svm.html

[67] Qian, Y., & Adali, S. (2014). Foundations of trust and distrust in networks: Extended structural balance theory. *ACM Transactions on the Web (TWEB)*, *8*(3), 13.