6-2017

Time-Series Link Prediction Using Support Vector Machines

Proceso L. Fernandez Jr
Ateneo de Manila University, pfernandez@ateneo.edu

Jan Miles Co

Custom Citation

Co, J.M., Fernandez, P. (2017/06). Time-series link prediction using support vector machines. Philippine Journal of Science, 146(2), 105-116.

Philippine Journal of Science
146 (2): 105-116, June 2017
ISSN 0031 - 7683

Time-Series Link Prediction Using Support Vector Machines

Jan Miles Co* and **Proceso Fernandez**

Department of Information Systems and Computer Science
Ateneo de Manila University, Quezon City, Philippines

The prominence of social networks motivates developments in network analysis, such as link prediction, which deals with predicting the existence or emergence of links on a given network. The Vector Auto Regression (VAR) technique has been shown to be one of the best for time-series based link prediction. One VAR technique implementation uses an unweighted adjacency matrix and five additional matrices based on the similarity metrics of Common Neighbor, Adamic-Adar, Jaccard's Coefficient, Preferential Attachment and Research Allocation Index. In our previous work, we proposed the use of the Support Vector Machines (SVM) for such prediction task, and, using the same set of matrices, we gained better results. A dataset from DBLP was used to test the performance of the VAR and SVM link prediction models for two lags. In this study, we extended the VAR and SVM models by using three, four, and five lags, and these showed that both VAR and SVM improved with more data from the lags. The VAR and SVM models achieved their highest ROC-AUC values of 84.96% and 86.32% respectively using five lags compared to lower AUC values of 84.26% and 84.98% using two lags. Moreover, we identified that improving the predictive abilities of both models is constrained by the difficulty in the prediction of new links, which we define as links that do not exist in any of the corresponding lags. Hence, we created separate VAR and SVM models for the prediction of new links. The highest ROC-AUC was still achieved by using SVM with five lags, although at a lower value of 73.85%. The significant drop in the performance of VAR and SVM predictors for the prediction of new links indicate the need for more research in this problem space. Moreover, results showed that SVM can be used as an alternative method for time-series based link prediction.

Key words: classification, link prediction, new links, support vector machine, vector auto regression

INTRODUCTION

As of August 2015, there are 3.175 billion active Internet users, with 2.206 billion active social media users. Over the year 2014, social media users have risen by 176 million in just a single year (Regan 2015). The rapid increase in social media users implies that either existing networks are growing or new networks are being created. The development in social networks serves as the main motivation for our study in network analysis, specifically link prediction.

Link prediction is an area in network analysis that deals with determining the existence or emergence of links given a network. Link prediction can be classified into two types: static link prediction and dynamic link prediction. In static link prediction, the detection of hidden links is based on a known partial snapshot, and the objective is to predict currently hidden but existing links in the known partial snapshot of the network (Tang et al. 2015). In dynamic

*Corresponding author: janmilesco@yahoo.com.ph link prediction, the objective is to predict future links at time *t*' (*t*' > *t*) that might emerge from the snapshot of the network at time *t* (Soares & Prudencio 2012).

Some previous works (Lee & Adorna 2012; Gupta & Singh 2014) have explored the static approach for the link prediction task.

In dynamic link prediction, the network is represented by a series of *n* snapshots $G(t^0),..,G(t^{n-1})$ that represent a network across time, and then used to predict the state of the network at t^n. As opposed to static link prediction, dynamic link prediction is able to use information regarding the occurrence and frequency of links across a network. Some previous works (Huang & Lin 2009; Soares & Prudencio 2012; Ozacan & Oguducu 2015) have explored the dynamic approach. According to surveys on link prediction techniques (Dhote Y et al. 2013; Wang & Liao 2014), most works focus on the static approach, while only few studies have explored the dynamic approach. Hence, we focus our efforts on the dynamic approach for the link prediction task.

Recent literature indicates that the Vector Auto Regression (VAR) technique is the most effective technique in time-series based link prediction (Ozacan & Oguducu 2015). In our earlier work (Co & Fernandez 2016), we incorporated some ideas from this technique and explored ways of improving its prediction performance. Since the VAR model assumes a linear dependence of the temporal links on multiple time-series, we proposed the use of Support Vector Machine (SVM) in order to more robustly handle a non-linear type of dependency while retaining the assumption that the dependency is on multiple time-series. SVM has been widely used (Hasan et al. 2006; Mengshoel et al. 2013; Nguyen-Thi et al. 2015) for the link prediction task. Using the AUC-ROC as the performance measure, we were able to improve the performance of VAR with two lags by 0.52% with SVM. A two-tailed paired t-test suggests that there is a significant difference, at 90% confidence level, with the performance of VAR and SVM. Specific to our co-authorship network from DBLP (Co 2016), the improvement of VAR by using SVM motivated us to perform more experiments to improve SVM further. This paper describes our experimentation on SVM for the link prediction task.

MATERIALS

VAR Technique for Link Prediction

The VAR econometric model is one of the most successful models for analyzing multivariate time-series (Ozacan & Oguducu 2015). In a recent work (Ozacan & Oguducu 2015), the VAR model was applied in time-series based link prediction where a network was represented by unweighted and weighted adjacency matrices. For each of these adjacency matrices, five matrices were created based on different similarity metrics, which are the Number of Common Neighbor (CN), Adamic-Adar Coefficient (AA), Jaccard's Coefficient (JC), Preferential Attachment (PA) and Resource Allocation Index (RA). The variables used in the multivariate time-series incorporate the values of the adjacency matrix and the five similarity metrics. In the VAR Model of Order p: Let $Y_t(t = 1,..., T)$ be a multivariate time-series with T observations, the p^{th} order vector autoregression, written as VAR(p), is a process that evolves as:

$$\hat{Y}_t = C + \Pi^1 Y_{t-1} + \Pi^2 Y_{t-2} +_{\dots} + \Pi^p Y_{t-p} + \epsilon_t; t = 1, \dots, T \qquad (1)$$

where $\hat{Y}_t = (Y_{1t}, Y_{2t}, \dots, Y_{nt})^T$ is a vector of n dimensions consisting of the estimated values of the variables in hand at time t; C is a vector of n dimension of intercepts; $\prod^j$ $j = 1, \dots, p$ are $n \times n$ coefficient matrices; and ϵ^t is a vector of n dimensions of errors following a multivariate white noise process which has zero mean, constant variance, and finite covariance and is uncorrelated with its past values, and n denotes the number of variables. Akaike Information Criterion (AIC) was used to compute the VAR model parameters and the lag (values of the specified variables occurring prior to the current observation) length for VAR(p) (Ozacan & Oguducu 2015).

Using a co-authorship network from DBLP, in both unweighted and weighted network representation, the performance of VAR with two lags was compared to static link prediction (by using one network snapshot for link prediction) and several dynamic link prediction techniques such as Moving Average (MA), Random Walk (RW), and Autoregressive Integrated Moving Average (ARIMA) for the prediction of both repeated (non-connected and connected link pairs) and new links (non-connected pairs in the last three previous snapshots). For the prediction of both repeated and new links, VAR showed the best performance among the many link prediction techniques. While in the prediction of new links only, VAR surpassed the performance of ARIMA in both unweighted and weighted network representations (Ozacan & Oguducu 2015).

Using a co-authorship network from DBLP, in both unweighted and weighted network representation, the performance of VAR with two lags was compared to static link prediction (by using one network snapshot for link prediction) and several dynamic link prediction techniques such as Moving Average (MA), Random Walk (RW), and Autoregressive Integrated Moving Average (ARIMA) for the prediction of both repeated (non-connected and connected link pairs) and new links (non-connected pairs in

the last three previous snapshots). For the prediction of both repeated and new links, VAR showed the best performance among the many link prediction techniques. While in the prediction of new links only, VAR surpassed the performance of ARIMA in both unweighted and weighted network representations (Ozacan & Oguducu 2015).

SVM in Highly Imbalanced Datasets

Support Vector Machine (SVM) is a well-known learning model that is mainly used for classification and regression analysis. For the classification type of problems, which is the problem type for our research, SVM computes for a maximum-margin line (or hyperplane) separator that can classify the instances of each class correctly. For datasets that are not linearly separable, SVM provides a mechanism for efficient projection of the instances to higher-dimensional space where the instances are presumed to be more easily separable. In any case, the computation for the maximum-margin hyperplane involves the optimization of a convex cost function using well-established numerical methods. In this study, we used the SVM implementation in R's e1071 library to build the classifier model for our dataset.

SVM has been shown to be very successful in many applications such as image retrieval, handwriting recognition and text classification. However, the performance of SVM drops significantly when faced with a highly imbalanced dataset. A highly imbalanced dataset is characterized by having instances from one class far significantly outnumbering the instances from another class. This makes it difficult to classify instances correctly due to a small number of the sample size for one class (Akbani et al. 2004). This type of dataset is observed in our co-authorship network (Co 2016), since there are significantly many potential co-authorship links but only less than 1% of these actually exist.

In order to improve the prediction performance of SVM, we applied several techniques from previous works that attempt to address the problem of highly imbalanced datasets.

Techniques for Handling Highly Imbalanced Datasets

In a previous work (Akbani et al. 2004), Synthetic Minority Over-sampling Technique (SMOTE) with Different Error Costs (DEC) was used to handle highly imbalanced datasets for SVM. Applying DEC to different classes in SVM forces the boundary away from the majority class, because in imbalanced datasets, the learned boundary of SVM tends to be too close to that class. Moreover, SMOTE makes positive instances more densely distributed, making the boundary better defined. Ten datasets were used to test the performance of SVM with the original dataset, SVM with random undersampling, SVM with SMOTE, SVM with DEC, and SVM with a combination of SMOTE and DEC. In seven out of ten datasets, the best performance was achieved with a combination of SMOTE and DEC (Akbani et al. 2004).

K-means clustering is one of the simplest unsupervised learning algorithms and has been used to solve well-known clustering problems. In a previous work (Rahman & Davis 2013), K-means clustering was used as an undersampling technique for a highly imbalanced dataset. The training set was divided into two. The first set contains the minority instances while the second set contains the majority instances. The majority instances were divided into K clusters, where K > 1. Each cluster was combined with the minority instances to form a candidate training set. The quality of the candidate training set was evaluated by using the Fuzzy Unordered Rule Induction Algorithm (FURIA) (Lotte et al. 2007). The best training set was used for classification with C4.5 decision tree (Barros et al. 2012). Using cardiovascular datasets from Hull and Dundee clinical sites, the proposed K-means undersampling method (Rahman & Davis 2013) outperformed the use of the original dataset and the use of another K-means undersampling technique (Yen & Lee 2009).

New Links Prediction and Its Challenges

Recent works on the link prediction task (Dunlavy et al. 2011; Ozacan & Oguducu 2015) have recognized the difficulty with the prediction of new links. New links can be defined as "the links that have not been previously seen at any time in the training set*"* (Dunlavy et al. 2011). In a previous work (Dunlavy et al. 2011), a dataset was created from inproceedings between 1991 to 2007 in DBLP and was used to test the performance of several link prediction methods such as CANDECOMP/PARAFAC (CP) tensor decomposition, Truncated Singular Value Decomposition (TSVD) - Collapsed Tensor (CT), Katz scores based on a Truncated spectral decomposition (TKatz) - CT, TKatz - Collapsed Weighted Tensor (CWT), Katz-CT, Katz-CWT. For each experiment, ten years were used as a training set and the eleventh year was used as a test set. The performances of the algorithms were tested for the prediction of all links and for the prediction of new links. The performances were measured using the Area Under the Receive Operating Characteristic Curve (AUC-ROC, or simply AUC). The highest AUC was achieved by Katz-CWT with an AUC of 95.7% for the prediction of all links, and an AUC of 91.2% for the prediction of new links (Dunlavy et al. 2011).

In a previous work (Ozacan & Oguducu 2015), using a dataset from DBLP, several link prediction methods were tested, such as the Auto Regressive Integrated Moving

Average (ARIMA) and the Vector Auto Regression (VAR). The performances of the algorithms were measured for the prediction of all links and for the prediction of new links, using the AUC as metric. The best performance was achieved by VAR with an AUC of 93% for the prediction of all links, and an AUC of 83% for the prediction of new links (Ozacan & Oguducu 2015).

Two different researches show that there is a difference in the prediction accuracy of all links and new links. Hence, we test the robustness of our VAR and SVM models in both the prediction of all links and the prediction of new links.

METHODS

Pre-Processing of Dataset for All Links

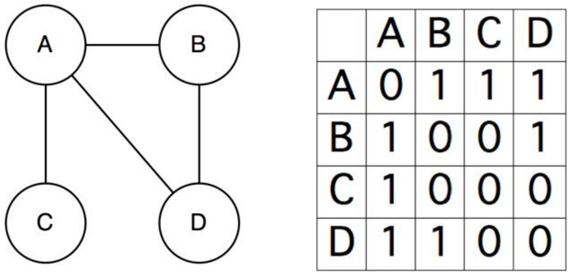
DBLP computer science bibliography is an online reference for open bibliographic information on computer science journals and proceedings. The co-authorship network that exists in DBLP was used for the link prediction task. We attempted to recreate the dataset used in a previous work (Ozacan & Oguducu 2015). However, we were unable to replicate the previous dataset because of two factors. First, DBLP does not allow database access by a specified date. Previous items might have been deleted and new items added to the database since the earlier work. Second, the previous work did not indicate which types of items were used. For simplification, we selected items labelled as articles. The previous dataset reported a total of 4,439 authors whereas our dataset contains only 1,743 authors.

First, we accessed DBLP and downloaded the dataset on August 11, 2015 (Co 2016). Next, we selected all items labelled as "article" and removed all items not labelled as "article," which includes: "inproceedings", "proceedings", "book", "incollection", "phdthesis", "masterthesis" and "www". Then we selected articles only from 2003 to 2013. Next, we removed all single-authored articles. Then, we removed all nonactive authors (authors who have 50 articles or less). Then we removed all articles that only have one active author. In our undirected network, the nodes represent authors, and the links represent co-authorship between two authors. The final dataset (Co 2016) contains 1,743 authors and 21,920 articles. The number of positive co-authorship links (a link exists between two authors) in this dataset is less than 1% of the total number of possible links.

The dataset was partitioned based on the year of publication to create the time-series models. The resulting 11 subsets correspond to years 2003 (t=0) up to 2013 (t=10) as snapshots of the dynamic co-authorship network graph. A snapshot is represented by an n x n unweighted adjacency matrix, where n = 1,743 (the number of authors). In the adjacency matrix, a value of 1 is placed if the corresponding two authors have co-authored an article that was published on the given year. Otherwise the value is 0. Suppose the network contains only four authors: A, B, C, D. An example of such a network and the corresponding unweighted adjacency matrix are shown in Figures 1 and 2.

Based on the unweighted adjacency matrix (A) for each snapshot, five matrices were created, based on five similarity metrics that are commonly used for link prediction, which were also used by a previous work (Ozacan & Oguducu 2015):

Number of Common Neighbor (CN)



	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	0
D	1	1	0	0

Figure 1. Network representation using an unweighted adjacency matrix example.

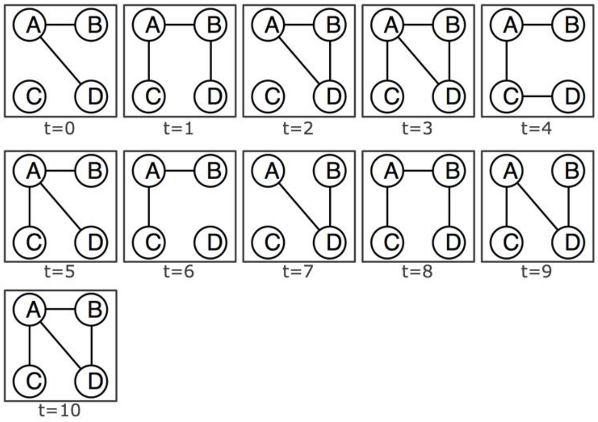


Figure 2. Network representation for each snapshot from t = 0 to t = 10.

$$\mathrm{CN}\ (x,y) = |\Gamma\ (x) \cap \Gamma\ (y)| \qquad (2)$$

Adamic-Adar Coefficient (AA)

$$AA\ (x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log\left(|\Gamma\ (z)|\right)} \qquad (3)$$

Jaccard's Coefficient (JC)

$$JC\,(x,y) = \frac{\Gamma\,(x) \cap \Gamma\,(y)}{\Gamma\,(x) \cup \Gamma\,(y)} \tag{4}$$

Preferential Attachment (PA)

$$PA\,(x,y) = |\Gamma(x)|^{*}|\Gamma(y)| \tag{5}$$

Resource Allocation Index (RA)

$$RA\,(x,y) = \sum_{Z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma\,(z)|} \tag{6}$$

where Γ(x) is the set of nodes adjacent to node (author) x.

VAR Predictor and SVM with Two Lags

For completeness, we include the method for VAR with two lags and SVM with two lags that we performed in our earlier work (Co & Fernandez 2016). We used this formula to train the VAR model with two lags, which we derived from a previous work (Ozacan & Oguducu 2015). Given a Metric Set $MS = \{A, CN, AA, JC, PA, RA\}$:

$$\hat{Y}_t^A = C_{t-1} + \sum_{i=1}^{2}\sum_{m\in MS} \alpha_{t-i}^{m} Y_{t-i}^{m} \alpha_{t-1}^{A} Y_{t-1}^{A} \tag{7}$$

where $\hat{Y}_t^A$ represents the predicted value of the link at time t and Y_t^m represents the actual values of the similarity metric m at time t. We computed the time-based VAR model parameters C_j and scalar coefficients a_j^i by linear regression, using the lm() function in R. This is a function used to fit linear models that returns a vector of coefficients including an intercept, and we used it to find the best fitting parameter set for each snapshot. We used the VAR model parameters to predict $\hat{Y}_t^A$ and constructed the predicted adjacency matrix at time t. The VAR model described here assumes that the co-authorship link at time t is linearly dependent on its corresponding factors.

First, we trained a VAR model using information from t, $t-1$, and $t-2$. We then used the VAR coefficients C_j and scalar coefficients a_j^i and information from t and $t-1$ to predict the link in the next time-period t +1.

$$Y_t^A = C_{t-1} + \sum_{i=1}^{2}\sum_{m\in MS} \alpha_{t-i}^{m} Y_{t-i}^{m} \tag{8}$$

$$\hat{Y}_{t+1}^A = C_{t-1} + \sum_{i=1}^{2}\sum_{m\in MS} \alpha_{t-i}^{m} Y_{t+1-i}^{m} \tag{9}$$

To train the SVM Model, we transformed the dataset where each instance that represents the presence or absence of a link is mapped to a multidimensional feature space that follows the linear dependency assumed in the VAR model. For VAR with two lags, the actual co-authorship link at time t is mapped to a 13-dimensional feature space with values from two previous time-periods $t-1$ and $t-2$, and the last column representing the presence of absence of the co-authorship link at time t.

For the training set, we reduced the dataset to have equal number of positive and negative instances (a link does not exists between two authors). For each year, we got the number of positive instances. We then randomly selected negative instances until we have an equal number of positive and negative instances. We used R for random sampling. Our code can be accessed in (Co 2016). We used an SVM linear kernel function to project the instances to a higher dimension where it is presumed to be linearly separable. The trained SVM model is used to predict the class for each instance, and these predictions are collected to construct the predicted adjacency matrix, $\hat{Y}_t^A$, of the network at time t (see Figure 3).

Attempts to Improve the SVM-Based Predictor

In the case of our dataset, the number of co-authorship links is less than 1% of the total number of possible links. We performed several techniques to enhance the performance of SVM on a highly imbalanced dataset. We applied some methods suggested by previous works (Akbani et al. 2004; Rahman & Davis 2013) to handle the imbalanced dataset.

Train an SVM Model Using k Samples

	Variables												Class
Link ID	Y_{t-1}^{A}	Y_{t-1}^{CN}	Y_{t-1}^{AA}	Y_{t-1}^{JC}	Y_{t-1}^{PA}	Y_{t-1}^{RA}	Y_{t-2}^{A}	Y_{t-2}^{CN}	Y_{t-2}^{AA}	Y_{t-2}^{JC}	Y_{t-2}^{PA}	Y_{t-2}^{RA}	Y_{t}^{A}
0													
1													
…													
k – 1													

Apply the Trained Model in n^2 Pairs

	Variables												Class
Link ID	Y_{t}^{A}	Y_{t}^{CN}	Y_{t}^{AA}	Y_{t}^{JC}	Y_{t}^{PA}	Y_{t}^{RA}	Y_{t-1}^{A}	Y_{t-1}^{CN}	Y_{t-1}^{AA}	Y_{t-1}^{JC}	Y_{t-1}^{PA}	Y_{t-1}^{RA}	Y_{t+1}^{A}
0													
1													
…													
n^2-1													

Figure 3. Training and using an SVM Model.

First, we set the Different Error Costs (DEC) to SVM. We applied this technique by setting the cost ratio to the inverse of the imbalance ratio. We created an imbalanced ratio of 1:2 by having twice as many negative instances as there are positive instances, and then we set the error cost to 1 for positive links (pairs of nodes that are linked) and 2 for negative links (pairs of nodes that are not linked). Second, we applied SMOTE by oversampling the positive instances by 100% and undersampling our negative instances by randomly selecting negative instances until we have an equal number of positive and negative instances for the training set. Third, we combined the first two techniques, which are DEC and SMOTE. We used SMOTE to create an imbalance ratio of 1:2 by having twice as many negative instances as there are positive instances, and then we applied Different Error Cost to SVM and set the error cost to 1 for positive links and 2 for negative links. Lastly, we used K-Means clustering to undersample the majority instances. We separated the positive instances from the negative instances, and then clustered the negative instances into two clusters. From the larger cluster, we randomly selected negative instances until we obtained an equal number of positive and negative instances. We created the final undersampled dataset by combining the selected negative instances to the positive instances and used this dataset for SVM. As illustrated in Figure 4, we performed the four experiments and then computed the AUC for each.

Benchmarking

To measure the performance of the prediction models, we used backtesting. For two lags, we built model for time *t* using the metric values for two previous time-periods, i.e., *t*-1 and *t*-2, and then used this model to predict the values at time *t*+1 using values from time *t* and *t*-1. We compared the prediction against the known values at time *t*+1. We were able to measure the performance of the VAR and SVM model with two lags in eight years from Year 3 to Year 10.

In the Receiver Operating Characteristic Curve (ROC), the *x*-axis corresponds to the False Positive Rate (FPR) and the *y*-axis corresponds to the True Positive Rate (TPR), which are computed as follows:

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{10}$$

$$TPR = \frac{True\ Positive}{True\ Positive + True\ Negative} \tag{11}$$

In our context, positives correspond to pairs of nodes that are linked while negatives correspond to pairs of nodes that are not linked. Note that a perfect model has an AUC of 100%, while a random model has an expected AUC of 50%.

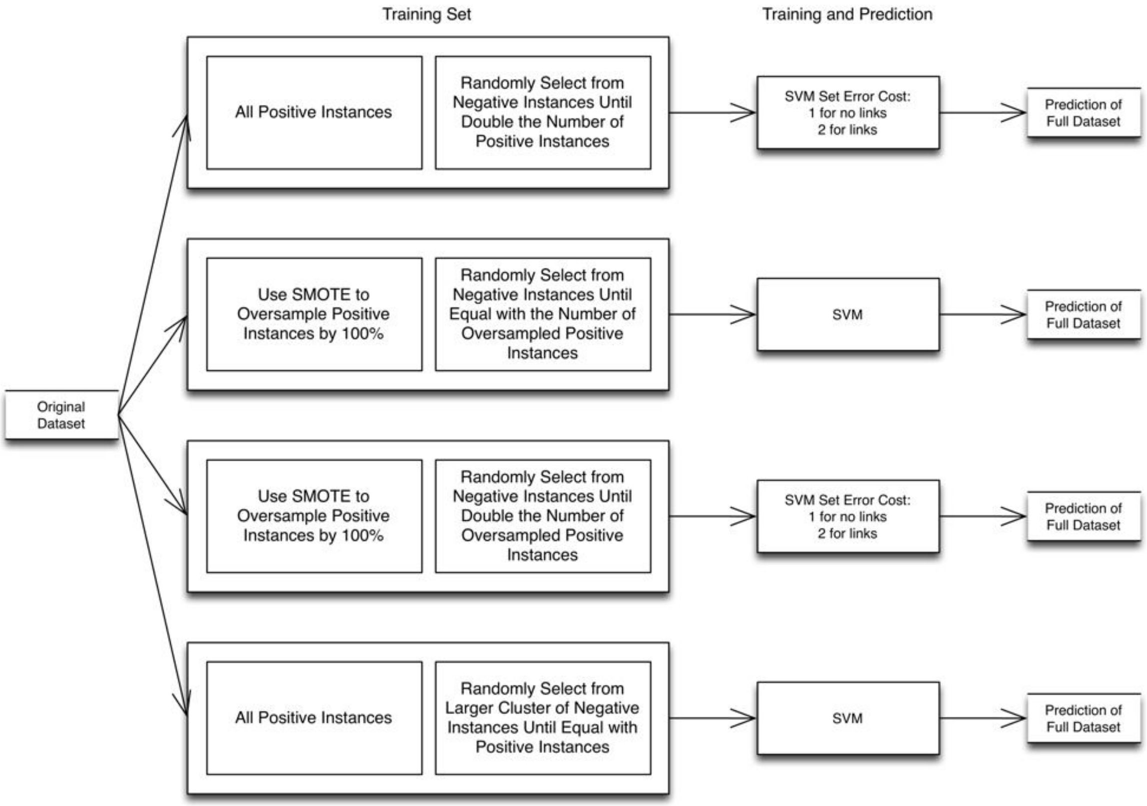

Training Set
Training and Prediction
Original Dataset
All Positive Instances
Randomly Select from Negative Instances Until Double the Number of Positive Instances
SVM Set Error Cost: 1 for no links 2 for links
Prediction of Full Dataset
Use SMOTE to Oversample Positive Instances by 100%
Randomly Select from Negative Instances Until Equal with the Number of Oversampled Positive Instances
SVM
Prediction of Full Dataset
Use SMOTE to Oversample Positive Instances by 100%
Randomly Select from Negative Instances Until Double the Number of Oversampled Positive Instances
SVM Set Error Cost: 1 for no links 2 for links
Prediction of Full Dataset
All Positive Instances
Randomly Select from Larger Cluster of Negative Instances Until Equal with Positive Instances
SVM
Prediction of Full Dataset

Figure 4. Improvements to SVM for highly imbalanced datasets.

Extending Two Lags to Three, Four and Five Lags
To incorporate additional lag values to the VAR model with two lags, we extended the formula for VAR with two lags to compute for VAR with three, four, and five lags. We did this by incorporating additional lag values.

For VAR with three lags, we added six factors for *t*-3, which is computed by this formula:

$$\hat{Y}_t^A = C_{t-1} + \sum_{i=1}^{3} \sum_{m \in MS} \alpha_{t-i}^m Y_{t-i}^m \alpha_{t-1}^A Y_{t-1}^A \qquad (12)$$

For VAR with four lags, we added 12 factors for *t*-3 and *t*-4, which is computed by this formula:

$$\hat{Y}_t^A = C_{t-1} + \sum_{i=1}^{4} \sum_{m \in MS} \alpha_{t-i}^m Y_{t-i}^m \alpha_{t-1}^A Y_{t-1}^A \qquad (13)$$

Finally, for VAR with five lags, we added 18 factors for *t*-3, *t*-4, and *t*-5, which is computed by this formula:

$$\hat{Y}_t^A = C_{t-1} + \sum_{i=1}^{3} \sum_{m \in MS} \alpha_{t-i}^m Y_{t-i}^m \alpha_{t-1}^A Y_{t-1}^A \qquad (14)$$

We did not use AIC to compute for the best lag length. Normally, AIC is used to avoid overfitting, by penalizing when there are too many parameters. In our case, the accuracies in our backtesting experiments increase as we increase the lag length, indicating that the models have not overfitted even at five lags.

To incorporate additional lag values to the SVM model with two lags, we added more features to accommodate the corresponding additional lag values similar to what was done in the lag extensions for the VAR technique.

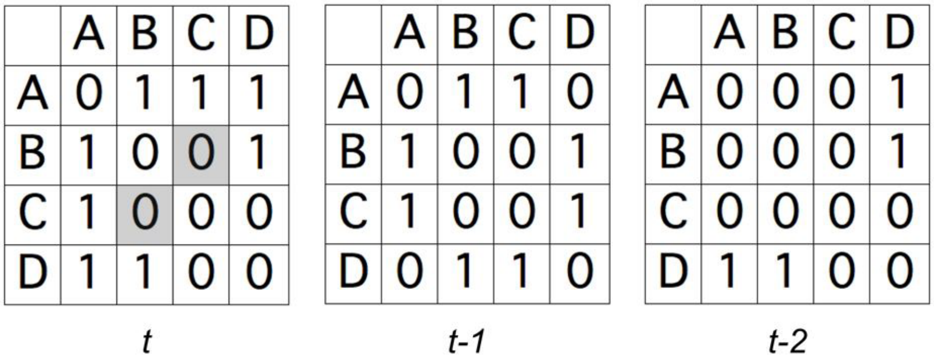
We performed experiments for three, four, and five lags mainly to see whether the prediction improves as we increase the lags. To do this, we compared predictions from Year 6 to Year 10 for three, four, and five. As we extended two lags to five lags, there was a great impact on run-time (time it takes to generate a prediction). Hence, we deemed that three, four, and five lags are sufficient to show whether adding lags does affect prediction performance positively or negatively, and we leave lag of six and higher for future studies.

New Link Prediction
From our basic experiments, we observed that there is difficulty with the prediction of new links. As stated earlier, new links can be defined as "the links that have not been previously seen at any time in the training set" (Dunlavy et al. 2011). For our research, we identified new links relative to a given network snapshot as links that did not exist in all of its previous lags in consideration. Hence, we separated all the candidate new links and trained a predictor for new link prediction. We created a new dataset for new link prediction where we selected all candidate new links only. For example, in two lags, a link at *t* is selected if the values at *t*-1 and *t*-2 in the unweighted adjacency matrix are both zero regardless of the value of the link at *t*. In Figure 5, the node pair B and C is not connected in *t*-1 and *t*-2 with a corresponding value of zero. Hence, the link between B and C is considered a potential new link regardless of the value of link B and C at time *t*.

We selected the candidate new links for each of the lags beginning from two lags to five lags. We trained new VAR and SVM models for new link prediction for two lags to five lags. Finally, we used AUC to measure the performance of the various models.

RESULTS AND DISCUSSION

Improvement of SVM over VAR
For completeness, we have included our previous results (Co & Fernandez 2016) for VAR, SVM, and enhancements for SVM with two lags. We were able to get the AUC-ROC in eight years, from Year 3 to Year 10. We compared the



	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	0
D	1	1	0	0

t

	A	B	C	D
A	0	1	1	0
B	1	0	0	1
C	1	0	0	1
D	0	1	1	0

t-1

	A	B	C	D
A	0	0	0	1
B	0	0	0	1
C	0	0	0	0
D	1	1	0	0

t-2

Figure 5. Potential new links example at *t*.

performance of VAR, SVM, and the technique to handle highly imbalanced datasets (see Table 1).

In five out of eight years, SVM was able to achieve a better performance than VAR. In terms of average AUC, SVM was able to outperform VAR with values of 81.87% and 81.35% respectively. A two-tailed paired *t*-test was conducted in order to determine if there is significant difference at 90% confidence level. The resulting *p*-value of 0.074 indicates that there is a statistically significant difference. For our dataset (Co 2016), in five out of eight years, SVM was able to achieve a better performance than VAR.

We attribute the improvement of SVM over VAR to the characteristic of the dataset. We performed Principal Component Analysis (PCA) on the 13 features for Lag of 2 at *t*=2. A visualization of the first two principal components reveals that dataset is nonlinear. See Figure 6. VAR assumes a linear dependency while SVM is more flexible in handling the nonlinear dataset. Hence, we attain better link prediction results with SVM.

In Table 1, we observed that the improvements to SVM contributed to a decrease in link prediction performance. The ratio of number of positive instances to negative instances might have been a contributing factor for the performance loss. Previously (Akbani et al. 2004), the techniques SVM-DEC, SVM-SMOTE, SVM-DEC &

Table 1. AUC of VAR and SVM with Two Lags.

	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10	Average
VAR	75.49	76.76	78.78	82.51	**81.72**	**84.41**	83.49	**87.65**	81.35
SVM	**76.20**	**78.08**	**79.48**	**82.56**	81.65	84.39	**85.17**	87.44	**81.87**
SVM-DEC	69.64	76.94	79.06	78.76	81.28	82.48	83.79	87.33	79.91
SVM-SMOTE	55.53	60.81	65.34	69.54	74.93	80.45	83.88	87.36	72.23
SVM-DEC & SMOTE	55.94	52.42	65.05	59.70	74.17	80.73	82.31	87.25	69.70
SVM-K-Means	55.72	60.77	65.13	70.06	74.76	80.52	83.86	86.91	72.22

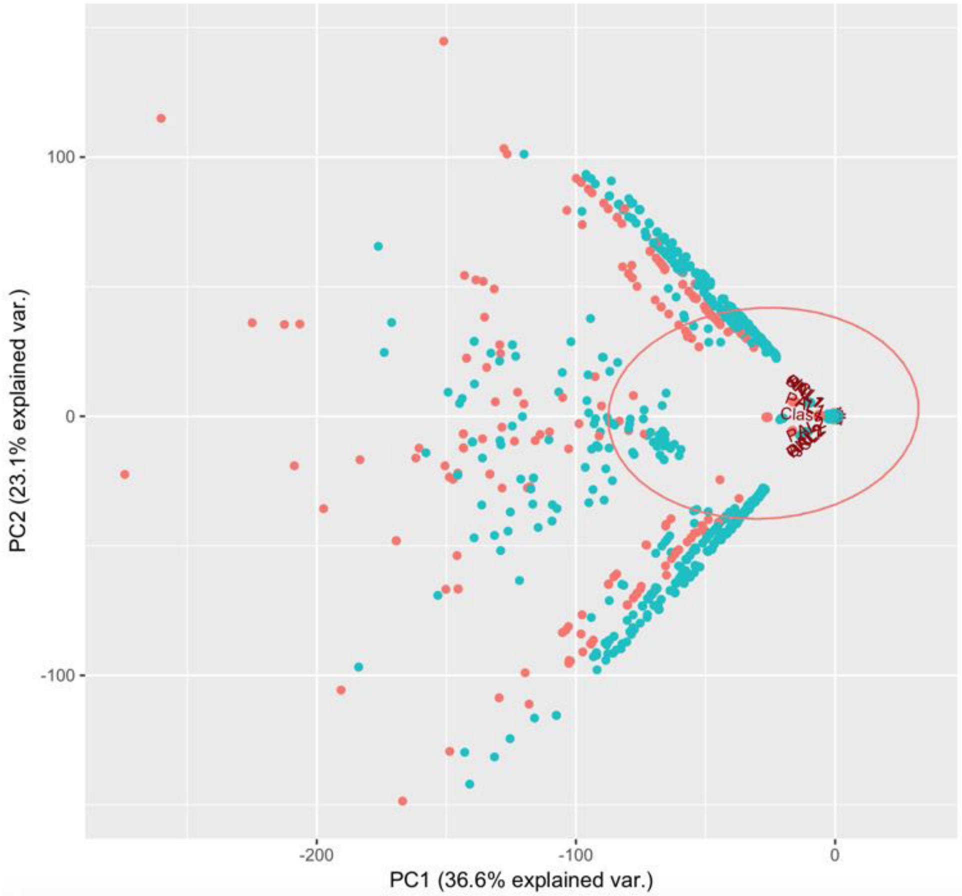


Figure 6. Illustration of First Two Principal Components of Dataset at *t*=2.

SMOTE were applied in datasets that have a positive sample to negative sample ratio of 1:4 to 1:130. For our dataset, the ratio across the time-slices is from 1:539 to 1:1526. The larger imbalance in our dataset might have contributed to the poor performance of SVM-DEC, SVM-SMOTE, and SVM-DEC & SMOTE. Furthermore, these techniques were previously used (Akbani et al. 2004) without undersampling whereas we were forced to perform sampling due to insufficient computing resources that can handle such a large dataset. As for SVM-K-Means, the sampling technique was previously used (Rahman & Davis 2013) to remove noise from a dataset. We may infer that our dataset has less noise and sampling by K-Means clustering removes proper instances (instances that are not noise) that contribute to an information loss.

Enhancing VAR and SVM with Additional Lag Values and Enhancements to SVM

For three lags, we were able to get the AUC-ROC in seven years from Year 4 to Year 10. Analogously, we were able to get AUC-ROC results in six and five years for four and five lags, respectively. To compare the impact of additional lag values, we compared the performance of VAR and SVM in five years from Year 6 to Year 10. A two-tailed paired *t*-test was conducted in order to determine if there is significant difference in the means (two lags compared to three, four, and five lags) at 90% confidence level (see Tables 2 and 3). In Table 3, in the last column, we indicate the *p*-value when VAR (with two, three, four, five lags) is compared to SVM (with two, three, four, five lags).

In three out of five years, the VAR model with five lags achieved the highest AUCs. In terms of Average AUC, the highest was also achieved with the VAR model with five lags at 84.96%. The *p*-value decreases as we increase the lag length. In three, four and five lags, at 90% confidence level, there is a statistical difference between the prediction performances.

In four out of five years, the SVM model with five lags achieved the highest AUCs. In terms of Average AUC, the highest was also achieved with the SVM model with five lags at 86.32%. The *p*-values are much smaller in SVM than VAR, which reflect that there is more statistical difference in the prediction performance of SVM with lag length greater than 2.

Table 2. VAR Predictor with two Lags to five Lags.

	Year 6	Year 7	Year 8	Year 9	Year 10	Average	*p*-value (from two lags)
Lag 2	82.51	81.72	**84.41**	83.49	87.65	84.26	-
Lag 3	83.34	82.32	84.04	84.13	88.99	84.71	0.092
Lag 4	83.69	82.57	84.13	84.86	**89**	84.9	0.044
Lag 5	**83.99**	**82.82**	84.17	**85.12**	88.69	**84.96**	0.038

Table 3. SVM Predictor with two Lags to five Lags.

	Year 6	Year 7	Year 8	Year 9	Year 10	Average	*p*-value (from two lags)
Lag 2	82.56	81.65	84.39	85.17	87.44	84.98	-
Lag 3	84.71	82.7	**85.5**	86.76	88.92	85.88	0.002
Lag 4	85.26	83.11	85.35	87.32	89.18	86.18	0.004
Lag 5	**85.61**	**83.35**	85.35	**87.67**	**89.61**	**86.32**	0.004

Figure 7 shows that as more information from the lags is added, the performance of the VAR and SVM models also improves. The SVM predictor was able to outperform the VAR predictor from two lags to five lags by increasingly bigger margins. In five lags, SVM was able to outperform VAR by 1.36% with an Average AUC 86.32%. A *p*-value of 0.017 using a two-tailed paired *t*-test with at least 90% confidence interval indicates that there is a significant difference with the performance of the VAR and SVM predictor. The techniques to handle the imbalanced dataset were unable to improve the performance of the SVM predictor from two to five lags as shown in Figure 7.

It is intuitive that as we add more information form the lags, the VAR and SVM models improve. The smaller lags (two, three, four lags) are subsets of five lags but with the other corresponding feature coefficients set to 0. Thus, incorporating more information from the lag values enriches the VAR and SVM models.

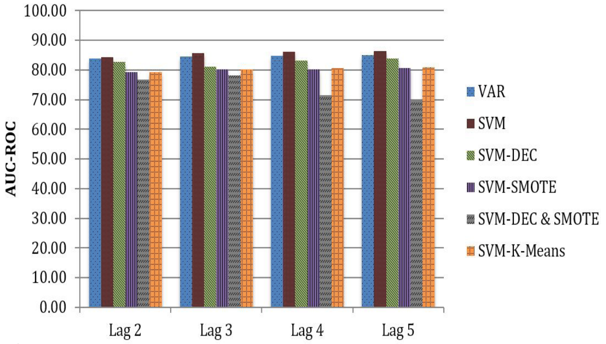


Figure 7. Improvements to SVM.

Prediction of New Links

For both VAR and SVM, the highest Average AUC was achieved with five lags at 72.97% and 73.85% respectively. In five lags, SVM was able to outperform VAR by 0.88% with an Average AUC 73.85%. A *p*-value of 0.082 using a two-tailed paired *t*-test at 90% confidence level indicates that there is a significant difference between the performances of the VAR and SVM new link predictors. The techniques applied to handle the imbalanced dataset were generally unable to improve the performance of the SVM predictor for new links from two to five lags as shown in Figure 8.

For the prediction of new links, the dataset in two lags has a positive to negative sample ratio of 1:946 to 1:2597. While for five lags, the ratio is from 1:1069 to 1:1715. The larger imbalance in the dataset for new links (compared with the imbalance in the dataset for all links) might have contributed to the difficulty in link prediction. Furthermore, this indicates that the feature set, which is composed of the adjacency matrix and the similarity metrics, do not perform well on new links compared to the prediction of all links.

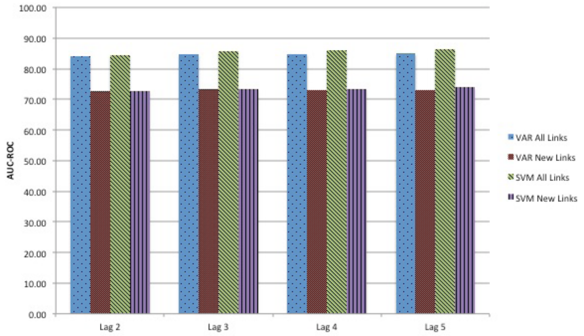


Figure 8. VAR and SVM on New Links.

Other Contributing Factors that Affect Link Prediction

Our dataset was preprocessed according to the previous work (Ozacan & Oguducu 2015) on link prediction with VAR mainly for replication. The preprocessing method used might or might not be a good representation of the larger network. Important nodes and links might have been removed from the original network, which affected the similarity metrics that were used. For future work, we propose the use of graph (network) sampling techniques from previous works (Dempsey et al. 2011; Kurant et al. 2010, Mohaisen et al. 2012; Zou & Holder 2010; Gjoka et al. 2010) where the main goal is to extract a smaller network from a larger network that is otherwise too large for analysis (Nagpal & Garg 2014), and use this representative network for link prediction.

Moreover, we used an undirected unweighted co-authorship network from DBLP. The weighted co-authorship network counterpart may be used for the same link prediction problem. Other types of dataset such as Facebook wall-posts and Twitter posts can be used to explore directed networks. With the use of other datasets, the performance of VAR and SVM can be analyzed further.

Furthermore, we acknowledge three weaknesses for the use of SVM for link prediction. First, the training time for SVM increases with the size of the training set. Hence, for our large co-authorship dataset (approximately 1.5 million instances per time-slice), we were forced to perform random sampling for the training set. Second, SVM is known to perform poorly on highly imbalanced datasets, which is the case for our co-authorship dataset. Our attempts to handle the highly imbalanced dataset failed to improve SVM. Third, SVM (and VAR) requires training a model (parameterization), which entails additional processing before link prediction (compared to that of unsupervised link prediction methods). Other classification techniques, such as Artificial Neural Network, can be used to address the weaknesses of SVM for link prediction.

CONCLUSION AND FUTURE WORK

In this study, we perform several experiments to further explore our previously proposed SVM-based link prediction technique that uses the VAR model multivariate time-series (with two lags) as a feature set for classification. We were able to further improve the VAR and SVM models by extending the VAR and SVM models of two lags to three, four, and five lags. The VAR and SVM models achieved their highest AUC values of 84.96% and 86.32% respectively using five lags. Results indicate that the performance of both VAR and SVM are improved with more data from the lags. Furthermore, techniques to handle imbalanced datasets for SVM, which is the case for our co-authorship network, failed to improve link prediction.

We also identified that the performance of both models is constrained by the difficulty in the prediction of new links, which we define as links that do not exist in any of their corresponding lags. Hence, we created new VAR and SVM models for the prediction of new links. The highest AUC-ROC was achieved by using SVM with five lags at a lower value of 73.85%. The significant drop in the performance of VAR and SVM models for new links implies that more research is needed to create more robust prediction models for new links.

Overall, we were able to show that SVM can be used as an alternative method for time-series link prediction. We used the result of our experiments to identify possible areas to explore on link prediction such as applying VAR and SVM in other datasets, developing similarity metrics specific for new links, using graph sampling techniques for improved network representation, and using other classification techniques such as Artificial Neural Network (ANN) for link prediction.

ACKNOWLEDGMENT

Funding from DOST-ERDT is greatly acknowledged.

REFERENCES

AKBANI R, KWEK S, JAPKOWICZ N. 2004. Applying support vector machine to imbalanced datasets. In: Boulicaut JF, Esposito F, Giannotti F, Pedreschi D, ed. Machine learning: ECML 2004. 15th European Conference on Machine Learning; 2004 September 20-24; Pisa, Italy. p. 39-50

BARROS RC, BASGALUPP MP, CARVALHO ACPLF, FREITAS AA. 2012. A survey of evolutionary algorithms for decision-tree induction. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(3): 291-312.

CO JM, FERNANDEZ P. 2016. Improving the vector auto regression technique for time-series link prediction by using support vector machine. MATEC Web of Conferences 56(01008): 1-5.

CO JM. 2016. Filtered DBLP for time-series based link prediction. Retrieved from https://www.researchgate.net/publication/300928276_Filtered_DBLP_for_Time-Series_Based_Link_Prediction.

DEMPSEY K, DURAISAMY K, ALI H, BHOWMICK A. 2011. A parallel graph sampling algorithm for analyzing gene correlation networks. Procedia Computer Science 4(2011): 136-145.

DHOTE Y, MISHRA N, SHARMA S. 2013. Survey and analysis of temporal link prediction in online social networks. In: 2013 International Conference on Advances in Computing, Communications and Informatics; 2013 August 22-25; Mysore, India. p. 1178-1183.

DUNLAVY D, KOLDA T, ACAR E. 2011. Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data 5(2): 10:1-7.

GJOKA M, KURANT M, BUTTS CT, MARKOPOULOU A. 2010. Walking in Facebook: A case study of unbiased sampling of OSNs. In: IEEE International Conference on Computer Communications; 2010 March 15-19; San Diego, CA, United States. p. 1-9.

GUPTA N, SINGH A. 2014. A novel strategy for link prediction in social Networks. In: 2014 CoNEXT on Student Workshop; 2014 December 2; Sydney, Australia. p. 12-14.

HASAN MA, CHAOJI V, SALEM S, ZAKI M. 2006. Link prediction using supervised learning. In: Link Analysis, Counterterrorism and Security 2016; 2006 April 22; Bethesda, Maryland.

HUANG Z, LIN DKJ. 2009. The time-series link prediction problem with applications in communication surveillance. INFORMS Journal on Computing 21(2): 286-303.

KURANT M, MARKOPOULOU A, THIRAN P. 2010. On the bias of BFS. In: 22nd International Teletraffic Congress; 2010 September 7-9; Netherlands. p. 1-8.

LEE JB, ADORNA H. 2012. Link prediction in a modified heterogeneous bibliographic network. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; 2012 August 26-29; Instanbul. p. 442-449.

LOTTE F, LECUYER A, ARNALDI B. 2007. FuRIA: A novel feature extraction algorithm for brain-computer interfaces using inverse models and fuzzy regions of interest. In: 2007 3rd international IEEE/EMBS Conference on Neural Engineering; 2007 May 2-5; Kohala Coast, Hawaii. p. 175-178.

MENGSHOEL OJ, DESAI R, CHEN A, TRAN B. 2013. Will we connect again? Machine learning for link prediction in mobile social networks. In: 11th Workshop on Mining and Learning with Graphs; 2013 August 11; Illinois, USA.

MOHAISEN A, LUO P, LI Y, KIM Y, ZHANG Z. 2012. Measuring bias in the mixing time of social graphs due to graph sampling. In: IEEE Military Communications Conference; 2012 October 29 – November 1; Orlando, Florida, United States. p. 1-6.

NAGPAL R, GARG R. 2014. Algorithms for reducing the size of network. International Journal of Innovative Research in Computer and Communication Engineering 2(11): 6512-6518.

NGUYEN-THI AT, NGUYEN PQ, NGO TD, NGUYEN-HOANG TA. 2015. Transfer AdaBoost SVM for link prediction in newly signed social networks using explicit and PNR features. Procedia Computer Science 60: 332-341.

OZACAN A, OGUDUCU SG. 2015. Multivariate temporal link prediction in evolving social networks. In: 2015 International Conference on Information Systems; 2015 June 28- July 1; Las Vegas, Nevada, United States. p.185-190.

RAHMAN MM, DAVIS DN. 2013. Cluster based undersampling for imbalanced cardiovascular data. In: World Congress on Engineering; 2013 July 3-5; London, UK. p. 1480-1485.

REGAN K. 10 Amazing social media growth stats from 2015. Retrieved from http://www.socialmediatoday.com/social-networks/kadie-regan/2015-08-10/10-amazing-social-media-growth-stats-2015.

SOARES PR, PRUDENCIO RB. 2012. Time series based link prediction. In: The 2012 International Joint Conference on Neural Networks; 2012 June 10-15; Brisbane, Australia. p. 1-7.

TANG J, CHANG S, AGGARWAL C, LIU H. 2015. Negative link prediction in social media. In: WSDM '15 Proceedings of the Eight ACM International Conference on Web Search and Data Mining; 2015 January 31 - February 6; Shanghai, China. p. 87-96.

WANG T, LIAO G. 2014. A review of link prediction in social networks. In: Proceedings of 2014 International Conference on Management of E-Commerce and E-Government; 2014 October 31 - November 2; Shanghai, China. p.147-150.

YEN SJ, LEE YS. 2009. Cluster-based under-sampling approaches for imbalanced data distributions. Journal Expert Systems with Applications: An International Journal 36(3): 5718-5727.

ZHOU R, HOLDER LB. 2010. Frequent subgraph mining on a single large graph using sampling techniques. In: Proceedings of the Eight Workshop on Mining and Learning with Graphs; 2010 July 25-28; Washington, DC, United States. p. 171-178.